

z/OS
3.2

DFSMSHsm Storage Administration



Note

Before using this information and the product it supports, read the information in [“Notices” on page 1415.](#)

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

Last updated: 2025-09-30

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

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

Contents

Figures.....	xxxvii
Tables.....	xlvi
About this document.....	li
Who should read this document.....	li
Major divisions of this document.....	li
Required product knowledge.....	liv
z/OS information.....	liv
How to read syntax diagrams.....	liv
Symbols.....	liv
Syntax items.....	lv
Syntax examples.....	lv
How to provide feedback to IBM.....	lvii
Summary of changes.....	lix
Summary of changes for z/OS 3.2.....	lix
Summary of changes for z/OS 3.1.....	lx
Part 1. DFSMSHsm Storage Administration Guide.....	1
Chapter 1. Welcome to DFSMSHsm.....	7
Purpose of space management.....	7
Purpose of availability management.....	7
What you are expected to do.....	8
Tasks for managing storage with DFSMSHsm.....	8
Overview of DFSMSHsm functions.....	9
DFSMSHsm functions for SMS-managed volumes.....	10
DFSMSHsm functions for non-SMS-managed volumes.....	14
Devices supported by DFSMSHsm.....	16
How to use the example system.....	17
Multiple z/OS images in the example system.....	18
DASD storage in the example system.....	18
Tape devices in the example system.....	19
Other assumptions in the example system.....	20
Chapter 2. Space management of SMS-managed storage.....	23
Space management functions.....	23
Automatic primary space management.....	23
Automatic secondary space management.....	24
Event-driven migration.....	24
Automatic recall.....	32
Automatic dynamic volume expansion space management.....	32
Command space management.....	32
Tasks for automatic space management.....	32
Specifying the storage to be managed and the space requirements.....	32
Specifying how groups of data sets are managed.....	35
Specifying when automatic primary space management can process.....	46

Specifying when automatic secondary space management can process.....	46
Specifying migration of data sets to overflow volumes.....	47
Specifying how to migrate data sets to migration level 2.....	47
Specifying how to do extent reduction.....	49
Specifying how to scratch expired data sets.....	49
Specifying the maximum automatic volume space management tasks and subtasks.....	49
Specifying the maximum automatic secondary space management tasks.....	50
Specifying the maximum interval migration or on-demand migration tasks.....	51
Specifying an on-demand migration number of concurrently queued volumes notification limit.....	52
Specifying recall tasks.....	52
Specifying small data set packing.....	52
Specifying the first qualifier for names of migrated data sets.....	53
Specifying if class transitions should be performed during event-driven migration.....	53
Configuring migration cleanup.....	54
Identifying migration level 2 volumes to DFSMSHsm.....	55
Specifying duplexing for migration level 2 tapes.....	55
Specifying the allowed target tiers.....	55
Integrity age for space management.....	56
Automatic space management operation examples.....	56
Automatic space management.....	56
Integrating cloud storage into a DFSMSHsm automatic space management environment.....	58
Starting automatic secondary space management.....	59
Starting automatic primary space management.....	64
Automatic primary space management.....	65
Event-driven migration example scenario.....	81
Class transitions for Db2 objects example scenario.....	82
Automatic recall.....	83
Additional space management during migration and recall when DFSMSHsm is the data mover.....	87
Command space management tasks.....	88
Deleting migrated data sets.....	88
Migrating an individual data set.....	89
Transitioning an individual data set.....	90
Migrating and transitioning an SMS-managed volume.....	91
Migrating and transitioning a storage group.....	92
Recalling SMS-managed data sets.....	92
Recalling direct access data sets.....	93
Moving data sets.....	93
Reconnecting data sets using fast subsequent migration.....	94
Chapter 3. Space management of non-SMS-managed storage.....	99
Differences in what the example system shows.....	99
Tasks for space management of non-SMS-managed storage.....	99
Specifying the age for deleting list and utility data sets.....	100
Specifying the minimum age for migrating data sets.....	101
Specifying when data sets become eligible for level 1 to level 2 migration.....	101
Specifying recall characteristics for the type of volume.....	101
Defining primary volumes to DFSMSHsm.....	102
Attributes of primary volumes.....	103
Defining pools of volumes.....	105
Pools in systems running under JES2.....	105
Pools in systems running under JES3.....	106
Automatic space management processing for non-SMS-managed storage.....	109
Automatic secondary space management.....	110
Automatic primary space management.....	111
Automatic interval migration.....	119
Recall.....	119

Command space management.....	123
Command volume space management.....	124
Command volume space management (primary).....	124
Controlling migration of data sets and volumes.....	124
Recalling a data set to a specific volume.....	124
Recalling an SMS-managed data set to a non-SMS-managed volume.....	125
Recalling an SMS-managed BDAM data set to a non-SMS-managed volume.....	125
Recalling an SMS-managed sequential extended-format data set to a non-SMS-managed volume.....	125
Chapter 4. Other space management considerations.....	127
Space management of extended address volumes (EAVs).....	127
Track management for SMS-managed volumes.....	128
Track management for non-SMS-managed volumes.....	128
Controlling allocations to extended address volumes (EAVs).....	129
Recommendations for using extended address volumes (EAVs).....	129
Common recall queue.....	129
Connecting to a common recall queue list structure.....	131
Populating the common recall queue.....	131
Selecting requests from the common recall queue.....	132
Processing recall requests on the common recall queue.....	132
Disconnecting from the common recall queue.....	132
Processing of common recall queue requests during shutdown.....	133
Deleting the common recall queue structure.....	133
Processing failures in a common recall queue environment.....	133
Processing priority of recalls and deletions.....	135
Specifying the order in which data sets from a single tape are processed.....	136
Migration volume contention priorities.....	136
Tape migration level 2 volume contention.....	136
Specifying the desired number of ML2 partial tapes.....	141
SDSP data set contention.....	142
Multitasking considerations for SDSP data sets.....	147
Preventing the processing of data sets during space management.....	149
Deferred generation data sets.....	149
Security considerations for automatic space management.....	149
Space management of data sets with discrete data set profiles.....	150
Password protection of data sets on DASD.....	151
Access method services considerations.....	152
GUARANTEED SPACE attribute considerations.....	152
Results of recalling GUARANTEED SPACE VSAM data sets.....	152
Possible problems.....	153
DFSMSHsm retrieval of management classes and storage groups.....	154
Deletion of dynamic I/O device considerations.....	155
Space management considerations for sequential extended-format data sets.....	155
Volume space management processing.....	155
Automatic primary space management processing.....	155
Recall processing.....	155
Object storage in the cloud.....	156
System set up to use the cloud.....	156
DASD volumes.....	156
SETSYS.....	156
Cloud password.....	156
Using the Cloud.....	157
Workload distribution with TCT.....	157
Migration.....	157
Recall.....	158
Delete.....	158
Listing information.....	158

Auditing information.....	158
Chapter 5. Space management procedures.....	161
Interrupting and restarting primary space management, secondary space management, or event-driven migration.....	161
When primary space management, secondary space management, or event-driven migration needs to be restarted.....	161
Automatic restart of primary space management or secondary space management.....	162
Automatic restart of event-driven migration.....	162
Manual restart of primary space management or secondary space management.....	163
Defining a new primary space management or secondary space management processing window.....	164
Allowing another host to take over secondary space management functions.....	164
Chapter 6. Availability management of SMS-managed storage.....	165
What the example system shows.....	165
Automatic incremental backup.....	165
Automatic full-volume dump.....	165
Command availability management.....	165
Tasks for automatic availability management.....	166
Dump tasks.....	166
Specifying which volumes to dump and the dump classes to use.....	166
Specifying when automatic dump processing starts.....	168
Specifying the DFSMSdss DASD I/O buffering technique to use for dump.....	168
Specifying concurrent copy for volume dumps.....	169
Specifying the maximum number of dump tasks.....	169
Specifying the day on which a dump occurs.....	170
Specifying the characteristics of dump classes.....	171
Defining dump volumes to DFSMSShsm.....	177
Backup tasks.....	177
Specifying which storage group has automatic backup performed.....	178
Specifying how to manage groups of data sets.....	178
Specifying a storage group priority for autobackup processing.....	179
Specifying whether to use concurrent copy.....	179
Specifying frequency of backup.....	180
Guaranteeing frequency of backup.....	180
Specifying the maximum number of backup versions to keep.....	181
Specifying how long to keep backup versions.....	181
Specifying who can back up data sets by command.....	185
Specifying whether to perform automatic backup.....	186
Example management classes.....	186
Specifying the start time for automatic backup.....	188
Specifying whether to perform volume backup and the kind of device to use.....	189
Specifying the first qualifier for names of backup versions.....	189
Specifying the maximum number of volume backup tasks in each host.....	190
Specifying the data set backup by command environment.....	190
Specifying whether to back up RACF discrete data set profiles.....	190
Specifying whether DFSMSShsm performs spill processing.....	190
Specifying whether to back up only changed data sets.....	190
Specifying how DFSMSShsm should respond to a data set in use.....	191
Specifying the days on which automatic backup occurs.....	192
Defining backup volumes to DFSMSShsm.....	192
Specifying duplexing for backup tapes.....	193
Automatic availability management operation.....	193
Dump processing.....	193
Deleting expired dump copies.....	195
Deleting empty dump containers.....	195
Defining what to do with tapes marked unavailable during DUMP processing.....	195

Dumping DFSMSHsm-managed volumes.....	196
Deleting and uncataloging excess dump VTOC copy data sets.....	198
Automatic backup processing.....	199
Phase 1: Backing up the control data sets.....	199
Phase 2: Moving backup versions.....	200
Phase 3: Backing up migrated data sets.....	200
Phase 4: Backing up SMS-managed volumes from storage groups with the automatic backup attribute.....	201
Selecting daily backup volumes.....	207
Backup volume spill process.....	215
Selecting spill backup volumes.....	216
Initial spill backup volume selection (DASD and tape).....	217
Subsequent spill backup volume selection (DASD and tape).....	218
Cleanup processing.....	218
Backup volume cleanup process.....	219
Releasing a backup volume.....	220
Automatic backup example.....	220
Command availability management.....	221
Dumping a volume.....	222
Dumping the volumes in one or more storage groups.....	222
Backing up an individual data set or UNIX file.....	222
Using the data set or UNIX file backup by command capabilities.....	224
Backing up all data sets on one or more volumes.....	230
Backing up all data sets in one or more storage groups.....	230
Backing up all UNIX files in a directory.....	230
UNIX file wildcard processing.....	231
Backing up all eligible changed data sets on one or more volumes.....	232
Backing up all eligible changed data sets in one or more storage groups.....	232
Backing up all eligible changed UNIX files in a directory.....	232
Deleting expired backup versions from the backup volumes.....	233
Deleting backup versions of specific data sets or UNIX files.....	239
Command recovery and restore.....	240
Data set recovery and restore.....	241
UNIX file recovery.....	250
Restoring a data set from a DFSMSHsm dump copy.....	251
Restoring VSAM data sets.....	253
Managing volume backups with fast replication.....	254
Combining fast replication options in DFSMSHsm.....	256
Combining copy services operations in DFSMS.....	257
Using the copy pool construct for fast replication setup.....	257
Using the copy pool backup storage group type for fast replication setup.....	259
Using the extend and overflow storage group types with fast replication.....	260
Preparing for the fast replication backup.....	260
Creating a fast replication backup.....	261
Recovering a fast replication backup.....	270
Withdrawing DFSMSHsm background copies.....	282
Holding and releasing the fast replication functions.....	283
Deleting unneeded DFSMSHsm fast replication backup versions.....	283
Obtaining DFSMSHsm fast replication backup information.....	284
Preventing DFSMSHsm functions from processing target volumes.....	288
Other copy pool considerations.....	288
Collecting fast replication messages in message data sets	291
Volume recovery and restore.....	294
Specifying the maximum number of data set recovery tasks in each host.....	294
Specifying the maximum number of volume recovery from dump tasks.....	294
Full-volume restore with update.....	295
Full-volume restore from a dump copy.....	298
Volume recovery from incremental backup versions.....	300

Processing priority of volume recoveries.....	302
Chapter 7. Availability management of non-SMS-managed storage.....	303
Tasks for availability management of non-SMS-managed storage.....	303
Specifying how frequently data sets are backed up.....	303
Specifying how many versions to keep for each backed up data set.....	303
Specifying which volumes are backed up automatically.....	304
Specifying which volumes are dumped automatically.....	305
Automatic availability management of non-SMS-managed volumes.....	306
Backing up cataloged and uncataloged data sets with the same name.....	307
Command availability management for non-SMS-managed storage.....	308
Dumping a volume.....	308
Changing the backup conditions for a data set.....	308
Backing up an individual data set.....	309
Inline backup of an individual data set.....	309
Backing up all primary volumes.....	309
Backing up all changed data sets on volumes.....	309
Deleting expired backup versions of non-SMS-managed data sets.....	310
Deleting backup versions of specific data sets.....	311
Recovering specific data sets.....	311
Recovery of VSAM data sets.....	314
Data set restore from a DFSMSHsm dump copy.....	314
Data set recovery of retired data sets.....	315
Combining full-volume restore with volume recovery.....	316
Volume recovery from incremental backup versions.....	316
Chapter 8. Aggregate backup and recovery support (ABARS).....	319
How aggregate backup works.....	322
Specifying data sets to be backed up.....	322
Backing up applications.....	324
ABACKUP command.....	324
Example of data sets associated with an aggregate group.....	325
Filtering for generation data sets.....	326
Preparing for aggregate backup.....	326
SMS requirements during aggregate backup.....	328
Creating selection data sets.....	329
Creating an instruction data set.....	334
Defining an aggregate group.....	336
Management class considerations.....	338
DFSMSHsm coexistence considerations for aggregate groups and management classes.....	339
Protecting aggregate backup tape volumes.....	340
Checking authorization during aggregate backup.....	340
Authorizing comprehensive ABACKUP command authority.....	341
Authorizing restricted ABACKUP command authority.....	341
Using the aggregate backup installation exits.....	342
Ensuring that the necessary volumes are online.....	343
Defining SETSYS parameters for aggregate backup.....	343
Determining the unit type for the control and data files.....	345
Minimizing data set contention.....	345
Data sets that DFSMSHsm dynamically allocates during aggregate backup processing.....	345
Performing aggregate backup.....	346
ABACKUP command.....	347
Issuing the ABACKUP command with the VERIFY parameter.....	349
Issuing the ABACKUP command with the EXECUTE parameter.....	350
Aggregate backup output.....	351
Data files.....	351
Control file.....	351
Instruction and activity log file.....	352

Activity log.....	352
ABACKUP CPU time and accounting codes.....	353
DFSMSshm coexistence considerations for new ABACKUP output data sets.....	354
Expiring ABR records in the BCDS.....	354
ABACKUP considerations.....	355
ABARS user catalog and catalog alias ABACKUP support.....	356
Aggregate backup assist.....	357
How aggregate backup assist works.....	357
What aggregate backup assist produces.....	357
Special considerations for ABARS.....	357
Authorization and security requirements.....	358
Input requirements.....	358
SMF data.....	358
Aggregate job names.....	358
Storage requirements.....	359
DD statements used by ABA.....	359
Sample JCL.....	360
Sample output.....	360
Aggregate stacking procedure.....	360
Backing up data with ABARS aggregate backup function.....	361
Stacking multiple aggregates on one tape.....	362
Aggregate recovery from the stacked tape.....	364
Preventing aggregate roll off and tape expiration.....	365
Deleting specific versions of aggregates from a stacked tape.....	365
RACF protection of tapes used for stacking.....	366
RACF protection of tapes after EXPIREBV processing of versions.....	366
End of volume during the stacking process.....	367
Cleanup after stacking errors.....	367
Copying ABARS aggregates to another device type.....	368
How aggregate recovery works.....	372
Recovered aggregate data sets.....	373
Example aggregate recovery site.....	374
Preparing for aggregate recovery.....	374
Authorization checking during aggregate recovery.....	376
Authorizing comprehensive ARECOVER command authority.....	377
Authorizing restricted ARECOVER command authority.....	377
Considering storage management subsystem requirements.....	378
Determining aggregate recovery site constraints.....	379
Retrieving the instruction data set.....	379
Creating an ABR record in the BCDS.....	381
Issuing the DEFINE ARPOOL command.....	382
Defining SETSYS parameters for aggregate recovery.....	384
Defining the model entity.....	387
Using the aggregate recovery installation exits.....	387
Processing generation data sets.....	387
GDSs from level 0 DASD in the include list.....	388
Migrated GDSs in the include list.....	388
GDSs in the ALLOCATE list.....	389
GDS processing order.....	389
Processing migrated data sets in the INCLUDE list.....	389
Processing VSAM data sets in the ALLOCATE list.....	389
Catalog considerations during ARECOVER processing.....	390
Performing aggregate recovery.....	391
ARECOVER command.....	392
Issuing the ARECOVER command with the PREPARE parameter.....	398
Issuing the ARECOVER command with the VERIFY parameter.....	398
Issuing the ARECOVER command with the EXECUTE parameter.....	399
Issuing the ARECOVER command with the ONLYDATASET parameter.....	399

Recovering the ABARS instruction data set.....	400
Backing up and recovering the ABARS activity log.....	401
Conflict avoidance and conflict resolution processing options.....	401
ARECOVER data set name conflict resolution.....	402
Holding and releasing ABARS functions.....	408
Displaying SMS class definitions during ABACKUP and ARECOVER.....	409
ABARS user catalog and catalog alias ARECOVER support.....	410
Data sets that are dynamically allocated during aggregate recovery processing.....	410
Aggregate recovery output.....	411
Activity log.....	411
ARECOVER CPU time and accounting codes.....	413
Using ISMF panels to perform aggregate backup.....	413
Using ISMF panels to perform aggregate recovery.....	418
Using the NetView file transfer program to transmit aggregate Backup files.....	427
Using NetView file transfer program panels.....	427
Issuing the ARECOVER command when using NetView FTP.....	431
ABARS considerations for sequential data striping.....	431
Aggregate backup processing.....	431
Aggregate recovery processing.....	432
Aggregate roll-off processing.....	432
Defining GDGs at the recovery site.....	433
Ensuring backup copies of ARECOVERed SMS-managed data sets exist at the recovery site.....	433
Directing ABARS files to DASD.....	434
 Chapter 9. Making disaster backup copies of DFSMSHsm-owned tape volumes.....	435
Planning disaster recovery.....	435
Disaster backup facilities.....	436
Making copies of DFSMSHsm-owned tape volumes.....	436
Converting tape volumes marked partially full to marked full.....	440
Replacing an original with an alternate tape volume.....	440
Creating duplex tapes off site.....	441
Sending alternate tapes off site.....	441
Managing alternate volumes.....	441
Disaster alternate volumes.....	442
Considerations when using disaster alternate volumes.....	443
 Chapter 10. Other availability management considerations.....	445
Additional considerations for backup.....	445
Backup volume contention resolution.....	445
Implications of redefining the backup cycle.....	446
Inline backup.....	447
Preventing backup processing when using the HOLD BACKUP command.....	449
GUARANTEED SPACE attribute.....	451
Data set or UNIX file backup with concurrent copy.....	453
Data set backup using NEWNAME.....	454
UNIX file backup using NEWNAME.....	455
Additional considerations for dump.....	455
Dumping DFSMSHsm-owned volumes.....	455
Dumping control data sets.....	456
Dumping the journal data set.....	456
Preventing dump processing.....	456
Dumping all allocated space.....	457
Invoking full-volume dump compression.....	457
Bypassing enqueue processing between full-volume dump and DFP functions.....	458
Volume recovery and restore considerations.....	458
Restoring data sets or volumes directly with DFSMSdss.....	458
Requesting a list of dump volumes for a dump copy.....	458
Determining the dump copy data set name.....	458

Invoking the DFSMSdss RESTORE function.....	459
Preventing recovery and restore processing when using the HOLD RECOVER command.....	459
Using the date-last-referenced field for recovery.....	459
Deferred generation data sets.....	460
Security considerations for availability management.....	460
Authorization checking during recovery.....	460
Recovering or restoring RACF-indicated data sets.....	461
Maintaining password protection for recovered data sets.....	463
DFSMSHsm retrieval of management classes.....	464
Availability management considerations for extended SAM and extended and compressed	
VSAM data sets.....	464
Availability management considerations for sequential data striping.....	464
Backup processing.....	464
Recover and restore processing.....	465
Common dump queue.....	466
Types of hosts used in a common dump queue.....	468
Connecting to a common dump queue.....	469
Placing requests on the common dump queue.....	470
Selecting requests from the common dump queue.....	471
Processing requests off of the common dump queue.....	471
Disconnecting from the common dump queue.....	472
Using QUERY, CANCEL and ALTERPRI in the common dump queue.....	472
Processing of failures in the common dump queue.....	474
Common recover queue.....	474
Types of hosts used in a common recover queue.....	476
Connecting to a common recover queue.....	478
Placing requests on the common recover queue.....	479
Selecting requests from the common dump queue.....	480
Processing requests off of the common recover queue.....	480
Using QUERY, CANCEL and ALTERPRI in the common recover queue.....	481
Disconnecting from the common recover queue.....	482
Processing of failures in the common recover queue.....	483
Chapter 11. Availability management procedures.....	485
Backup and recovery procedures.....	485
Interrupting and restarting backup processing.....	485
Defining a new automatic backup start window.....	487
Backing up and recovering a catalog.....	487
Full-volume dump procedures.....	489
Interrupting and restarting automatic dump processing.....	489
Defining a new automatic dump start window.....	490
Chapter 12. Maintaining DFSMSHsm control data sets.....	493
Tasks for maintaining the control data sets.....	493
Specifying how DFSMSHsm backs up the control data sets.....	493
Allocating backup data sets.....	494
Choosing the data mover to be used.....	494
Using non-intrusive journal backup.....	496
Specifying how many backup versions to keep.....	497
Specifying the names of the backup data sets.....	497
Specifying the device type for backup of the control data sets.....	498
Invoking the CDS backup exit (ARCCBEXT).....	499
Backing up to multiple DASD data sets.....	499
Backing up to multiple tape data sets.....	501
Considerations in a multiple DFSMSHsm host environment.....	502
Considerations when using multicluster control data sets.....	502
Nulling the journal.....	503
Changing the names of the backup data sets.....	503

Changing backup device from tape to DASD.....	503
Changing backup device from DASD to tape.....	504
Using the QUERY command to get CDS information.....	505
Backing up the control data sets manually.....	505
Possible failures during CDS and journal backup.....	505
Reasons for recovering the control data sets.....	507
I/O errors.....	507
Recovering control data sets using the enhanced CDS recovery function.....	508
Requirements for using the enhanced CDS recovery function.....	508
Implementing enhanced CDS recovery using JCL.....	508
Recovering multicluster non-key range control data sets.....	513
Control data set recovery procedure using CDSVERSIONBACKUP with UPDATEC.....	513
Updating recovered control data sets using UPDATEC.....	516
VSAM considerations for the control data sets.....	516
Determining if control data sets are accurate.....	517
Repairing errors in the control data sets.....	517
Procedures for handling a journal EOV error.....	517
Recovering from a journal I/O error.....	518
Moving the journal.....	518
Steps to ensure that the control data sets are accurate.....	519
Control data set backup and recovery examples.....	519
Recovery examples using the enhanced CDS recovery function.....	519
Performance considerations in a multiple DFSMSHsm host environment.....	525
MASH configuration considerations.....	526
Chapter 13. Managing DFSMSHsm media.....	527
Moving all data sets from one level 0 volume to other level 0 volumes.....	527
Processing SMS-managed data sets.....	527
Processing non-SMS-managed data sets.....	528
Examples of moving data sets between level 0 volumes.....	528
Moving some data sets from migration level 1 volumes.....	530
Using the FREEVOL command for migration level 1 volumes.....	531
Using the MIGRATE command for migration level 1 volumes.....	532
Deleting non-SMS-managed data sets from migration volumes.....	532
Moving all the data sets from a migration level 1 volume.....	533
Processing for SMS-managed data sets.....	533
Processing for non-SMS-managed data sets.....	534
Moving data sets from a DASD migration level 2 volume to other migration level 2 volumes.....	534
Removing residual dump and backup VTOC copy data sets from migration level 1 volumes.....	535
Preventing further migration to a DASD migration volume.....	535
Moving data sets from DASD backup volumes.....	535
Removing volumes from DFSMSHsm management or control.....	536
Removing dump copies and dump VTOC copy data sets from DFSMSHsm.....	536
JES3 considerations.....	537
Consolidating valid data on tape volumes.....	537
Defining the parameters of the recycle process.....	537
Initiating recycle processing.....	539
PERCENTVALID calculation.....	541
Selecting tape volumes during recycling.....	541
Making cartridge-type, single-file-format tapes from reel-type or multiple-file-format tapes.....	542
Limiting data that is written to a cartridge-type device.....	542
Specifying capacity utilization.....	543
Considerations for using the 3592 Model J tape drive.....	544
Considerations for using the 3592 Model E05 tape drive.....	545
Considerations for using the 3592 Model E06 tape drive.....	546
Considerations for using the 3592 Model E07 tape drive.....	548
Limiting the number of ML2 and backup data sets that span tape volumes.....	549
How to handle inconsistencies between tape media contents and OCDS TTOC records.....	550

Recovering a damaged DFSMSHsm single-file migration level 2 volume.....	551
Making migration level 2 key range volumes available.....	551
Recovering a migrated data set.....	551
Recovering a migration volume.....	552
Procedure 1.....	552
Procedure 2.....	553
Recovering backup versions from a damaged migration level 1 volume.....	554
Returning dump tapes to use.....	555
Managing SDSP data sets.....	555
Reorganizing SDSPs.....	556
SDSP backup and recovery.....	556
Removing SDSPs.....	557
Protecting DFSMSHsm-owned tape volumes.....	558
Converting to a new tape technology.....	558
Requesting DFSMSHsm service from your applications.....	558
 Chapter 14. Obtaining information.....	 559
Obtaining information from the control data sets.....	559
Obtaining backup volume information.....	560
Obtaining data set information.....	560
Obtaining UNIX file information.....	561
Obtaining copy pool information.....	562
Obtaining dump class information.....	564
Obtaining dump volume information.....	564
Obtaining DFSMSHsm host serialization information.....	565
Obtaining lists of backup and dump parameters and backup volume serial numbers.....	566
Running new reports with the DFSMSrmm Report Generator.....	566
Obtaining information about migration and DFSMSHsm-managed volumes.....	567
Obtaining DFSMSHsm-managed volume information.....	568
Obtaining tape volume information.....	569
Obtaining user authorization information.....	570
Determining which tape dump volumes are available for output during dump processing.....	570
Obtaining information about the DFSMSHsm address space.....	570
Determining which tape volumes are available for output.....	573
Determining which tape backup volumes are available for output.....	573
Determining which tape migration level 2 volumes are available for output.....	574
Obtaining statistical information about DFSMSHsm operations.....	574
Running new reports with report generator.....	575
Using the problem determination aid to trace DFSMSHsm processing.....	586
Specifying to DFSMSHsm to use the PDA.....	586
Using the TRAP command to obtain information about conditions in the system when an error occurs.....	586
Using DEBUG to determine the results of DFSMSHsm processing without moving data sets.....	587
 Chapter 15. DFSMSHsm tape libraries.....	 589
Automated tape library.....	589
Manual tape library.....	589
 Chapter 16. Using DFSMSHsm with DFSMSrmm.....	 591
Retaining DFSMSHsm tape volumes by expiration date.....	591
Retaining DFSMSHsm tape volumes by DFSMSrmm vital record processing.....	591
Tape security when using DFSMSHsm and DFSMSrmm.....	592
Recommendations for using DFSMSHsm with DFSMSrmm.....	592
 Chapter 17. Operating procedures for DFSMSHsm.....	 593
Partitioning DFSMSHsm functions.....	593
Starting DFSMSHsm.....	594
Stopping DFSMSHsm.....	595

Shutdown processing on the common recall queue.....	596
Preventing DFSMSHsm functions from running.....	596
Allowing previously prevented DFSMSHsm functions to run.....	600
Canceling DFSMSHsm requests.....	604
Altering the priority of DFSMSHsm queued requests.....	606
Printing DFSMSHsm log information.....	607
Restarting DFSMSHsm after an abnormal end.....	614
Running automatic functions concurrently.....	615
Updating the DFSMSHsm control data sets.....	616
Chapter 18. RACF considerations.....	617
RACF basics.....	617
RACF data set profiles.....	617
RACF indication.....	618
Always-call support.....	618
RACF protect-all environment.....	619
RACF environment for DFSMSHsm.....	619
DFSMSHsm and RACF concepts.....	620
The RACF user ID for DFSMSHsm.....	622
RACF started task table.....	622
Always-call support for migrated data sets and backup copies.....	623
Migration and recall from DASD.....	624
Backup and recovery from DASD.....	624
Aggregate backup and recovery data set security.....	624
Backing up RACF-indicated data sets.....	625
Authorizing the use of concurrent copy for data set backup commands.....	625
Small data set packing data set security.....	625
Uncataloging migrated data sets.....	625
Controlling command access with RACF FACILITY class.....	625
Chapter 19. Health Checker for DFSMSHsm.....	627
Check registration.....	627
Chapter 20. Data recovery scenarios.....	629
Case 1: Damaged CDS, full journal.....	629
Case 2: Damaged journal, undamaged CDS.....	632
Case 3: Full journal, undamaged CDS.....	632
Case 4: Structurally damaged CDS, missing journal records.....	633
Case 5: Time sequence gaps in CDS records, journal records also missing.....	633
Case 6: Overwritten migration tape.....	634
Case 7: Overwritten backup tape.....	637
Case 8: Damaged ML1 volumes.....	641
Case 9: Reestablish access to previously deleted migrated data sets (no backup exists, ML2 only).....	646
Case 10: Correcting errors within the common recall queue.....	647
Case 11: Recovering a deleted ML1 data set without a backup.....	649
Case 12: Recovering from a large-scale cyber event.....	650
Chapter 21. Consolidated commands, storage group definitions, and management classes for the example system.....	653
ADDVOL command.....	653
DEFINE command.....	655
SETSYS command.....	656
Storage groups.....	658
Management classes.....	659
Chapter 22. Preventive methods for avoiding data loss.....	661
Steps for preventing a full journal (damaged CDS).....	661

Preventing a full journal (undamaged CDS).....	661
Preventing structural damage to CDS and missing journal records.....	662
Preventing time sequence gaps in CDS records (journal records also missing).....	662
Preventing overwritten migration tape or overwritten backup tape.....	663
Prepare for recovery of a head crash on an ML1 volume.....	663
Reestablishing access to previously deleted, migrated data sets (no backup exists, ML2 only)....	663
 Chapter 23. Linux-z/OS DFSMSHsm dump or restore how-to.....	665
How DFSMSHsm supports linux.....	665
Using DFSMSHsm with Linux requirements.....	665
Hardware environment.....	665
DASD background.....	665
Linux requirements.....	666
Linux disk utilities.....	666
Authorization requirements.....	667
Setting up your dump environment for Linux volumes.....	667
Performing only dumps.....	667
When do you want your dumps performed?.....	667
How long do you want to keep your dumps?.....	668
How many dumps are stacked on each dump tape?.....	668
How many tape drives do you have?.....	668
Recovering your data.....	668
Recovering a volume with DFSMSHsm.....	668
Recovering a volume with DFSMSdss.....	669
Recovering a partition with DFSMSHsm.....	669
Recovering a partition with DFSMSdss.....	670
Example dump environment.....	670
How to define your dump cycle.....	671
How to define your dump classes.....	671
How to ADDVOL your volumes.....	672
How to start automatic dumping.....	672
Recovering from a dump.....	673
Submitting DFSMSHsm commands using JCL and FTP.....	674
The DFSMSHsm LIST command.....	675
Naming conventions.....	675
Data set and partition naming conventions.....	675
Volume serial number naming conventions.....	676

Part 2. DFSMSHsm Storage Administration Reference.....677

Chapter 24. DFSMSHsm commands.....	681
User commands.....	681
Operator commands.....	681
Storage administrator commands.....	682
System programmer commands.....	683
Using RACF FACILITY class to control authorization of commands.....	684
Using DFSMSHsm commands.....	684
Positional parameters.....	684
Keyword parameters.....	685
Abbreviating commands and parameters.....	685
Delimiters.....	686
Line continuations.....	686
Data set naming conventions.....	686
Comments.....	686
TSO attention key.....	686
How to read syntax diagrams.....	687
Considering the command sequence.....	687

Submitting DFSMSHsm commands.....	687
Commands in a system without RACF: Method 1.....	688
Commands in a system without RACF: Method 2.....	689
Chapter 25. ABACKUP command: Backing up aggregated data sets.....	691
Syntax of the ABACKUP command.....	692
Required parameters of the ABACKUP command.....	692
agname: Specifying the aggregate group name.....	692
EXECUTE and VERIFY: Specifying whether data movement processing or data verification occurs.....	692
Optional parameters of the ABACKUP command.....	693
FILTEROUTPUTDATASET: Specifying the output data set.....	693
LIST: Listing data sets not backed up due to the SKIP parameter.....	693
MOVE: Removing the data sets.....	693
OPTIMIZE: Overriding the SETSYS ABARSOPTIMIZE command.....	694
PROCESSIONLY: Designating the categories to process.....	694
SKIP: Skipping data sets on volumes protected by PPRC or XRC.....	694
STACK and NOSTACK: Overriding the SETSYS ABARSTAPES setting.....	695
UNIT: Designating the output device.....	695
Examples of how to code the ABACKUP command.....	695
Chapter 26. ADDVOL command: Adding or changing DFSMSHsm volumes.....	697
Syntax of the ADDVOL command.....	698
Required parameters of the ADDVOL command.....	700
volser: Specifying the volume to be added or changed.....	700
BACKUP, DUMP, MIGRATION, and PRIMARY: Identifying the type of volume.....	700
UNIT: Specifying the type of device.....	700
Optional parameters of the ADDVOL command.....	701
AUTOBACKUP and NOAUTOBACKUP: Specifying whether the primary volume is eligible for automatic backup.....	701
AUTODUMP and NOAUTODUMP: Specifying whether a volume is eligible for automatic dumping to tape or cloud.....	702
AUTOMIGRATION and NOAUTOMIGRATION: Specifying whether the primary volume is eligible for automatic volume space management.....	702
AUTORECALL and NOAUTORECALL: Specifying whether the primary volume is eligible for automatic recall.....	703
BACKUPDEVICECATEGORY: Specifying where backup versions reside.....	703
DAILY and SPILL: Identifying the type of backup volume.....	704
DELETEBYAGE, DELETEIFBACKEDUP, and MIGRATE: Specifying the type of space management on a primary volume.....	704
DENSITY: Specifying the tape density.....	705
DRAIN and NODRAIN: Specifying whether a volume should be selected for migration output.....	706
DUMPClass: Restricting the specified volume to use a specific dump class.....	706
MIGRATIONLEVEL1 and MIGRATIONLEVEL2: Specifying the migration level.....	707
OVERFLOW and NOOVERFLOW: Specifying the usage of a level 1 volume for data set migration and backup versions.....	707
SMALLDATASETPACKING and NOSMALLDATASETPACKING: Specifying small data set packing on the level 1 volume.....	708
THRESHOLD: Specifying thresholds of occupancy.....	708
TRACKMANAGEDTHRESHOLD: Specifying the high and low threshold for the track-managed space of a non-SMS level 0 extended address volume.....	710
Selecting thresholds for volumes.....	710
Examples of how to code the ADDVOL command.....	711
Chapter 27. ALTERDS command: Changing the back-ups for a non-SMS-managed data set.....	715
Syntax of the ALTERDS command.....	715
Required parameters of the ALTERDS command.....	716
name: Specifying the name of the data set or UNIX file being altered.....	716

Optional parameters of the ALTERDS command.....	716
FREQUENCY and SYSFREQUENCY: Specifying the frequency of backup versions.....	716
VERSIONS and SYSVERSIONS: Specifying the number of backup versions to maintain.....	717
Examples of how to code the ALTERDS command.....	717
Chapter 28. ALTERPRI command: Altering the priority of queued requests.....	719
Syntax of the ALTERPRI command.....	720
Required parameters of the ALTERPRI command.....	720
REQUEST, USERID, and DATASETNAME, and FILENAME: Specifying the requests to be reprioritized.....	720
Optional parameters of the ALTERPRI command.....	720
HIGH and LOW: Specifying the new priority of a request.....	720
Examples of how to code the ALTERPRI command.....	721
Chapter 29. ARECOVER command: Recovering a specified aggregate group.....	723
Syntax of the ARECOVER command.....	724
Required parameters of the ARECOVER command.....	725
AGGREGATE: Specifying the aggregate group to be recovered.....	725
DATASETNAME: Specifying the control file.....	725
VOLUMES and XMIT: Specifying the volume type.....	725
EXECUTE, VERIFY, and PREPARE: Specifying whether data movement or verification should occur or whether an ABR record should be created.....	726
Optional parameters of the ARECOVER command.....	726
ACTIVITY: Recovering the activity log.....	727
DATASETCONFLICT: Resolving data set naming conflicts.....	727
DATE and VERSION: Specifying the specific aggregate group to be recovered.....	728
INSTRUCTION: Recovering the instruction data set.....	728
MENTITY: Specifying a model entity.....	728
MIGRATEDDATA: Specifying the migration level for recovering migrated data sets.....	728
NOBACKUPMIGRATED: Specifying that automatic backup does not occur for recovered migrated data sets.....	729
ONLYDATASET: Recover a single data set or a group of data sets from an aggregate group backup.....	729
PERCENTUTILIZED: Specifying the percentage of DASD space used during ARECOVER of data sets to non-SMS volumes.....	730
RECOVERNEWNAMEALL and RECOVERNEWNAMELEVEL: Specifying a high-level qualifier for recovered data sets.....	730
STACK and NOSTACK: Specifying whether or not the ABACKUP output is STACKED.....	731
TARGETUNIT: Specifying unit types to be used for ARECOVER.....	731
TGTGDS: Overriding the SETSYS ARECOVERTGTGDS command.....	732
UNIT: Specifying unit types used for ARECOVER.....	732
VOLCOUNT: Allowing data sets to expand to multivolume as necessary during ARECOVER....	733
How to create the ABR record.....	733
Examples of how to code the ARECOVER command.....	733
Chapter 30. AUDIT command: Auditing DFSMSHsm.....	737
Auditing control data sets.....	737
Auditing common queues.....	738
Using the AUDIT command.....	738
Syntax of the enhanced AUDIT commands.....	738
Syntax of the original AUDIT commands.....	740
Required and optional parameters of the AUDIT command.....	743
ABARSCONTROLS: Requesting an audit of ABR records in the BCDS.....	743
ALL, BACKUPCONTROLDATASET, MIGRATIONCONTROLDATASET, and OFFLINECONTROLDATASET: Requesting an audit of data sets.....	743
BACKUPTYPE, BACKUPVOLUMES, and VOLUMES: Requesting an audit of primary, migration, or backup volumes.....	744
COMMONQUEUE: Requesting an audit of common queues.....	745

COPYPOOLCONTROLS: Requesting an audit of fast replication CDS record relationships.....	746
DATASETCONTROLS: Requesting an audit of data set control records.....	746
DATASETNAMES, LEVELS, and RESUME: Requesting an audit of data sets.....	746
DIRECTORYCONTROLS VOLUMES: Comparing a VTOC or a TTOC to CDS records.....	747
FILECONTROLS: Requesting an audit of UNIX file control records.....	747
FILE, FILELEVEL, and RESUME: Requesting an audit of UNIX files.....	748
MASTERCATALOG and USERCATALOG: Requesting an audit of a catalog.....	748
MEDIACONTROLS VOLUMES, MEDIACONTROLS(SDSP) VOLUMES and MEDIACONTROLS(CLOUD): Requesting an audit of control information.....	748
VOLUMECONTROLS (BACKUP, MIGRATION, and RECOVERABLE): Requesting an audit of control information.....	749
Optional parameters of the AUDIT command.....	750
FIX and NOFIX: Specifying whether auditing repairs errors.....	750
OUTDATASET, SYSOUT, and TERMINAL: Specifying the output location for the report.....	751
REPORT: Specifying how much audit information to print.....	752
SERIALIZATION: Stabilizing control information.....	752
Using the AUDIT command.....	753
Chapter 31. AUTH command: Identifying authorized users.....	755
Syntax of the AUTH command.....	755
Required parameters of the AUTH command.....	755
userid: Specifying the user whose authorization is to be changed.....	756
DATABASEAUTHORITY and REVOKE: Giving authority to or revoking authority of a user.....	756
Examples of how to code the AUTH command.....	757
Chapter 32. BACKDS command: Backing up a specific data set.....	759
Syntax of the BACKDS command for a UNIX file.....	760
Required parameters of the BACKDS command.....	761
dsname: Specifying the name of the data set to be backed up.....	761
filepath: Specifying the name of the UNIX file to be backed up.....	761
Optional parameters for the BACKDS command.....	762
CC: Specifying concurrent copy for data set backup.....	762
CHANGEDONLY/TOTAL: Specifying whether to back up UNIX files that have a last backup date in their metadata	764
DELETE: Specifying whether to delete the UNIX file after a successful backup.....	764
EXCLUDE and EXCLUDEF: Specifying files or directories to be excluded from processing.....	764
INUSE: Tailoring backup processing of UNIX files in use at the time of backup.....	766
NEWNAME: Specifying a name to assign to the backup version.....	767
RECURSE: Examine subdirectories of the specified UNIX file path.....	769
RETAIN DAYS: Specifying the number of days to retain a backup copy.....	769
TARGET: Specifying a required target output device.....	770
UNIT: Specifying the type of device.....	770
VOLUME: Specifying the volume where the data set resides.....	770
Delete expired backup versions of UNIX files.....	770
Examples of how to code the BACKDS command.....	771
Chapter 33. BACKFILE command: Backing up a specific UNIX file.....	773
Chapter 34. BACKDS command: Backing up a specific data set.....	775
Chapter 35. BACKFILE command: Backing up a specific UNIX file.....	777
Chapter 36. BACKVOL command: Backing up data sets from volumes or dumping volumes.....	779
Backing up data sets from one or more SMS-managed volumes or dumping SMS-managed volumes.....	779
Syntax of the BACKVOL command with SMS-managed volumes.....	780
Required parameters for SMS-managed volumes.....	781
CONTROLDATASETS: Specifying the control data sets to be backed up.....	781

STORAGEGROUP: Specifying the storage group to back up or dump.....	781
VOLUMES: Specifying the volumes to be backed up or dumped.....	782
Optional parameters for SMS-managed control data sets.....	783
BACKUPDEVICECATEGORY: Specifying the backup device for the control data sets and journal.....	783
DATAMOVER: Specifying the type of data movement used when backing up the control data sets.....	783
NULLJOURNALONLY: Specifying that DFSMSHsm null the journal.....	784
Optional parameters for SMS-managed volumes.....	784
BACKUPDEVICECATEGORY: Specifying where the backup versions are to reside.....	784
DUMP, INCREMENTAL, and TOTAL: Specifying which data sets are backed up.....	785
DUMPCLASS: Specifying that each dump copy be targeted to a particular dump class.....	785
RETENTIONPERIOD: Requesting a specific retention period for the dump copy.....	786
STACK/MAXSTACK and MINSTACK: Specifying the maximum and minimum number of dump copies to be stacked for the specified dump classes.....	786
TERMINAL: Requesting that backup or dump messages appear at the terminal.....	788
Examples of how to code the BACKVOL command for SMS-managed volumes.....	788
Backing up data sets from one or more non-SMS-managed volumes or from all primary volumes or dumping non-SMS-managed volumes.....	789
Syntax of the BACKVOL command with non-SMS-managed volumes.....	790
Summary of parameters.....	790
Required parameters for non-SMS-managed volumes or all primary volumes.....	791
CONTROLDATASETS: Specifying the control data sets to be backed up.....	791
PRIMARY and VOLUMES: Specifying the volumes to be backed up or dumped.....	791
Optional parameters for non-SMS-managed control data sets.....	792
BACKUPDEVICECATEGORY: Specifying the backup device for the control data sets and journal.....	792
DATAMOVER: Specifying the type of data movement used when backing up the control data sets.....	792
NULLJOURNALONLY: Specifying that DFSMSHsm null the journal.....	792
Optional parameters for non-SMS-managed volumes or all primary volumes.....	793
BACKUPDEVICECATEGORY: Specifying where the backup versions reside.....	793
DUMP, INCREMENTAL, and TOTAL: Specifying which data sets are backed up.....	793
DUMPCLASS: Specifying that each dump copy be targeted to a particular dump class.....	794
FREQUENCY: Specifying the frequency of backup for data sets.....	794
RETENTIONPERIOD: Requesting a specific retention period for the dump copy.....	794
STACK/MAXSTACK and MINSTACK: Specifying the maximum and minimum number of dump copies to be stacked for the specified dump classes.....	795
TERMINAL: Requesting that backup or dump messages appear at the terminal.....	795
UNIT: Specifying the type of device.....	795
Examples of how to code the BACKVOL command for non-SMS-managed volumes.....	795
Chapter 37. BDELETE command: Deleting backup versions of a data set.....	797
Syntax of the BDELETE command.....	797
Required parameters of the BDELETE command.....	797
dsname: Specifying the name of the data set whose backup versions are to be deleted.....	797
pathname: Specifying the name of the UNIX file whose backup versions are to be deleted ...	798
ALL, DATE and TIME, and VERSIONS: Specifying which backup versions to delete.....	798
ALL: Deleting all backup versions.....	798
DATE and TIME: Specifying the date and time of the backup version to delete.....	798
VERSIONS: Specifying the backup versions to delete.....	799
Optional parameters of the BDELETE command.....	799
FROMVOLUME: Specifying the volume where the data set resided when the backup versions were created.....	799
Examples of how to code the BDELETE command.....	799
Chapter 38. CANCEL command: Canceling a queued or active DFSMSHsm request.....	801
Syntax of the CANCEL command.....	801

Optional parameters of the CANCEL command.....	801
DATASETNAME: Canceling a queued request by data set name.....	802
FILENAME: Canceling a queued request by UNIX file name.....	802
REQUEST: Canceling a queued request by request number.....	802
USERID: Canceling a queued request by user ID.....	802
SASINDEX: Canceling an active request by ABARS secondary address space.....	803
TCBADDRESS: Canceling an active request by TCB address.....	803
Examples of how to code the CANCEL command.....	803
Chapter 39. DDELETE Command – Deleting cloud dump copies.....	805
Syntax of the DDELETE command.....	805
Required parameters of the DDELETE command.....	805
STORAGEGROUP: Specifying the storage group whose dump copies are to be deleted.....	805
VOLUMES: Specifying the volumes whose dump copies are to be deleted.....	806
DATE and TIME: Specifying the particular cloud dump copy to delete.....	806
Optional parameters of the DDELETE command.....	806
LASTCOPY: Invalidating the only valid dump copy of a source volume.....	806
Examples of how to code the DDELETE command.....	806
Chapter 40. DEFINE command: Defining control structures for use by DFSMSHsm.....	809
Syntax of the DEFINE command.....	810
Optional parameters of the DEFINE command.....	813
ARPOOL: Identifying the volumes for aggregate recovery.....	813
BACKUP: Specifying the backup cycle.....	814
CYCLESTARTDATE: Specifying the start date of a cycle.....	814
DUMPCLASS: Adding or changing a volume dump class.....	815
DUMPCLASS(AUTOREUSE and NOAUTOREUSE): Specifies whether DUMP volumes are automatically available for reuse when invalidated.....	816
DUMPCLASS(CLOUD TAPE): Specifying where the dump copies to reside.....	817
DUMPCLASS(DATASETRESTORE and NODATASETRESTORE): Data set restore is allowed from a full-volume dump copy in this dump class.....	818
DUMPCLASS(DAY): Specifying the target day of the dump cycle for this class.....	819
DUMPCLASS(DISABLE): Specifying the volume dump class and making it unavailable as a target of any full-volume dump.....	820
DUMPCLASS(DISPOSITION): Specifying the disposition of dump volumes.....	820
DUMPCLASS(ENCRYPT): Specifying whether to encrypt the dump data with host based encryption.....	820
DUMPCLASS(FASTREPLICATIONDUMP): Specifying whether a dump class is required or optional.....	821
DUMPCLASS(FASTREPLICATIONRECOVER): Specifying whether a dump class copy is immediately available to be mounted for recovery.....	822
DUMPCLASS(FREQUENCY): Specifying the minimum number of days that must elapse between volume dumps of a volume to this class.....	822
DUMPCLASS(HWCOMPRESS): Specifying whether to compress the data to be encrypted.....	823
DUMPCLASS(RECOVERRESET): Specifying whether to reset the data-set-changed indicator in the VTOC entry.....	823
DUMPCLASS(RESET and NORESET): Specifying whether DFSMSdss resets the change bit for each data set following a full-volume dump.....	824
DUMPCLASS(RETENTIONPERIOD): Specifying the enforcement of retention periods for the dump copies in this class.....	824
DUMPCLASS(STACK/MAXSTACK) and DUMPCLASS(MINSTACK): Specifying the maximum and minimum number of dump copies that are stacked for this dump class.....	825
DUMPCLASS(TAPEEXPIRATIONDATE): Specifying a unique expiration date for all dump copies created in a dump.....	827
DUMPCLASS(UNIT): Specifying a default tape unit name.....	827
DUMPCLASS(UNASSIGNTAPEUNIT): Specifying unassigning empty dump volumes.....	828
DUMPCLASS(VTOCCOPIES): Specifying the number of dump VTOC copy data sets to keep for dump copies created in this class.....	828

DUMPCCLASS(ZCOMPRESS): Specifying whether to use zEDC Services on the dump data.....	829
DUMPCYCLE: Specifying the automatic dump cycle.....	829
MIGRATIONCLEANUPCYCLE and SECONDARYSPMGMTCYCLE: Specifying the cycle for automatic secondary space management functions.....	830
MIGRATIONLEVEL2: Specifying DASD level 2 volumes and their associated key range structure.....	831
POOL: Identifying the non-SMS-managed volumes where the data sets with the same first qualifier are recalled.....	832
PRIMARYSPMGMTCYCLE: Specifying the cycle for automatic primary space management.....	832
SWITCHTAPES: Releasing mounted tapes for command data set backup.....	833
VOLUMEPOOL: Identifying groups of volumes where data sets are recalled.....	834
DUMPCCLASS(TAPE CLOUD): Specifying where the dump copies to reside.....	835
Examples of how to code the DEFINE command.....	836
 Chapter 41. DELETE Command: Deleting a migrated data set from a migration volume.....	839
Syntax of the DELETE command.....	839
Required parameters of the DELETE command.....	839
dsname: Specifying the data set to be deleted.....	839
Optional parameters of the DELETE command.....	839
PURGE: Specifying deletion of a data set.....	840
Examples of how to code the DELETE command.....	840
 Chapter 42. DELVOL command: Removing a volume from DFSMSHsm management.....	841
Syntax of the DELVOL command.....	841
Required parameters of the DELVOL command.....	842
volser: Specifying the volume you are deleting.....	842
BACKUP, DUMP, MIGRATION, and PRIMARY: Specifying the type of volume.....	842
Optional parameters of the DELVOL command.....	843
COPYPOOLCOPY: Deleting a copy pool dump volume.....	843
LASTCOPY: Invalidating the only valid dump copy of a source volume.....	843
MARKFULL, PURGE, REASSIGN, and UNASSIGN: Specifying the options for deleting a backup, dump, or ML2 volume.....	843
Examples of how to code the DELVOL command.....	845
 Chapter 43. DISPLAY command: Displaying DFSMSHsm storage locations.....	849
 Chapter 44. EXPIREBV command: Deleting unwanted backup versions of data sets and UNIX files..	851
Syntax of the EXPIREBV command.....	852
Required parameters of the EXPIREBV command.....	852
DISPLAY and EXECUTE: Specifying whether to delete the backup versions.....	852
Optional parameters of the EXPIREBV command.....	853
ABARSVERSIONS: Specifying that only aggregate group versions are to be processed.....	853
NONSMSVERSIONS: Specifying expiration criteria for backup versions of data sets not managed by SMS.....	854
STARTKEY and RESUME: Determining the starting point for BCDS processing during command execution.....	855
UNIXPATH: Specifying the UNIX file, empty directory, or nonempty directory to analyze for backup.....	855
ENDKEY: Determining the ending point for BCDS processing during command execution.....	857
OUTDATASET and SYSOUT: Specifying the output location for EXPIREBV DISPLAY listing.....	858
Examples of how to code the EXPIREBV command.....	859
 Chapter 45. FIXCDS command: Displaying or modifying MCDS, BCDS, and OCDS records.....	863
 Chapter 46. FRBACKUP command: Requesting a fast replication backup or dump version.....	865
Syntax of the FRBACKUP command.....	865
Required parameter of the FRBACKUP command.....	866
COPYPOOL: Specifying the name of the copy pool for backup.....	866

Optional parameters of the FRBACKUP command.....	866
BACKUPSTORAGEGROUP: Overriding the copy pool backup storage group with an alternate storage group.....	866
DUMP: Creating a fast replication DASD version followed by a dump.....	866
DUMPCCLASS: Specifying unique dump classes for fast replication dump processing.....	867
EXECUTE, PREPARE, WITHDRAW, and DUMPONLY: Specifying the backup or dump action....	867
FCINCREMENTAL: Requesting DFSMSHsm to create a copy pool version using incremental FlashCopy.....	868
FCINCREMENTALLAST: Requesting a FlashCopy incremental copy with a subsequent withdraw.....	869
FORCE: Replacing a DASD backup copy that has an incomplete dump copy.....	869
MESSAGEDATASET: Specifying the date and time used in the message data set name.....	869
NOVTOCENQ: Requesting DFSMSHsm to not serialize the VTOC of processed volumes.....	870
TOKEN: Specifying an identifier for each copy pool backup version.....	870
Examples of how to code the FRBACKUP command.....	870
 Chapter 47. FRDELETE command: Deleting fast replication backup or dump versions.....	873
Syntax of the FRDELETE command.....	873
Required parameters of the FRDELETE command.....	873
COPYPOOL: Specifying the name of the copy pool for deletion.....	873
ALL, TOKEN, and VERSIONS: Specifying the delete action.....	873
Optional parameters of the FRDELETE command.....	874
DASDONLY, DUMPONLY, and BOTH: Specifying which copies to delete.....	874
Examples of how to code the FRDELETE command.....	874
 Chapter 48. FREEVOL command: Freeing DFSMSHsm-owned volumes.....	877
Syntax of the FREEVOL command.....	878
Required parameters of the FREEVOL command.....	878
MIGRATIONVOLUME and BACKUPVOLUME: Specifying a volume to free.....	878
ML1BACKUPVERSIONS: Moving backup versions off migration volumes.....	878
Optional parameters of the FREEVOL command.....	879
AGE: Specifying data set use.....	879
TARGETLEVEL: Specifying the target to which data sets are moved.....	879
RETAINNEWESTVERSION: Preventing movement of backup versions.....	880
Examples of how to code the FREEVOL command.....	881
 Chapter 49. FRRECOV command: Requesting a fast replication recovery.....	883
Syntax of the FRRECOV command.....	884
Required parameters of the FRRECOV command.....	885
COPYPOOL: Specifying the copy pool for which to recover all backup volumes.....	885
DSNAME: Specifying the data sets to recover from a copy pool backup version.....	886
TOVOLUME: Specifying a single volume within a copy pool to recover.....	886
Optional parameters of the FRRECOV command.....	886
ALLOWPPRCP: Specifying Metro Mirror and Global Mirror primary volumes and preserve mirror options for a FlashCopy operation.....	886
ALLOWXRCP: Specifying an extended remote copy primary volume option for a FlashCopy operation.....	887
DATE, GENERATION, TOKEN, VERSION: Specifying a specific backup version to recover.....	888
FASTREPLICATION: Indicating a recovery preference.....	888
FORCE: Allowing recovery from a DASD backup copy that has an incomplete dump copy.....	889
FROMCOPYPOOL: Recovering a volume or data set that belongs to multiple copy pools.....	889
FROMDASD: Recovering a volume from DASD.....	889
FROMDUMP: Recovering a volume from a dump tape.....	889
MESSAGEDATASET: Specifying the date and time used in the message data set name.....	890
NEWNAME: Specifying a new data set name for the recovered data set.....	891
NOCOPYPOOLBACKUP: Setting an alternate return code.....	891
REPLACE: Recovering a cataloged data set.....	891
RSA: Overriding the RSA key label	891

VERIFY: Verifying that no unexpected FlashCopy relationship exists for a volume in a copy pool.....	892
Examples of how to code the FRRECOV command.....	892
Chapter 50. HOLD command: Preventing DFSMSHsm functions from running.....	895
Syntax of the HOLD command.....	896
Optional parameters of the HOLD command.....	897
ABACKUP: Preventing aggregate backup.....	897
ALL: Preventing all DFSMSHsm functions.....	898
ARECOVER: Preventing aggregate recovery.....	898
AUDIT: Ending the audit function.....	899
AUTOEXPIRATION: Preventing data sets from expiring.....	899
AUTOMIGRATION: Preventing automatic volume and automatic secondary space management.....	900
BACKUP: Preventing backups.....	900
COMMONQUEUE: Preventing common queue functions.....	901
DUMP: Preventing dumps.....	902
ENDOFDATASET: Specifying when aggregate group processing should stop.....	903
ENDOFDATASET and ENDOFVOLUME: Specifying when volume processing should stop.....	904
EVENTDRIVENMIGRATION: Preventing Interval and On-demand Migration.....	904
EXPIREBV: Ending the expire backup function.....	905
FRBACKUP: Preventing fast replication backup or dump processing.....	905
FRRECOV: Preventing fast replication recovery processing.....	905
LIST: Ending the list function.....	907
LOG: Preventing logging.....	907
MIGRATION: Preventing space management.....	907
RECALL: Preventing recall and deletion.....	907
RECOVER: Preventing recovery and restore.....	908
RECYCLE: Ending the recycle function.....	908
REPORT: Ending the report function.....	909
TAPECOPY: Preventing DFSMSHsm from copying a tape.....	909
TAPEREP: Preventing DFSMSHsm from replacing an original tape volume with an alternate.....	909
Examples of how to code the HOLD command.....	909
Chapter 51. HSEND CMD command: Issuing DFSMSHsm authorized-user commands from a TSO terminal.....	913
Syntax of the HSEND CMD command.....	913
Required parameters of the HSEND CMD command.....	913
command: Specifying DFSMSHsm authorized command.....	913
Optional parameters of the HSEND CMD command.....	913
WAIT and NOWAIT: Specifying whether to wait for command completion.....	914
Examples of how to code the HSEND CMD command.....	915
Chapter 52. HSENDU command: Issuing DFSMSHsm authorized user commands to a Filemode DFSMSHsm host.....	917
Chapter 53. LIST command: Listing information from the MCDS, BCDS, and OCDS.....	919
Syntax of the LIST command.....	920
Required parameters of the LIST command.....	925
AGGREGATE: Listing the aggregate backup and recovery activity information.....	925
BACKUPVOLUME: Listing backup volume entries.....	925
CLOUD: Listing information from cloud storage.....	925
COPYPOOL: Listing information for each version of a copy pool.....	926
COPYPOOLBACKUPSTORAGEGROUP: Listing copy pool information for each volume in the specified copy pool backup storage group.....	926
DATASETNAME and LEVEL: Listing data set entries.....	926
DUMPCCLASS: Listing dump class profiles.....	927
DUMPVOLUME: Listing information about a single dump volume or a set of dump volumes....	927

FILENAME and FILELEVEL: Listing UNIX file entries.....	928
HOST: Listing all control data sets serialized by the DFSMSHsm host.....	928
MIGRATIONVOLUME, MIGRATIONLEVEL1, MIGRATIONLEVEL2, and VOLUME: Listing migration volumes.....	929
PRIMARYVOLUME: Listing primary volumes.....	929
TAPETABLEOFCONTENTS: Listing tape volume information from the OCDS.....	930
TAPETABLEOFCONTENTS(volser): Listing tape volume information and data set names from the TTOC.....	930
USER: Listing user entries.....	931
Optional parameters of the LIST command.....	931
ALLDUMPS and BACKUPCONTENTS: Determining existence of dumps or listing incremental backup contents.....	931
ALLVOLS, DATASETS, DUMPVOLS, FASTREPLICATIONVOLS, and NOVOLS: Listing a specific type of copy pool information.....	932
BACKUPCONTROLDATASET, MIGRATIONCONTROLDATASET, and BOTH: Controlling the source of information.....	933
DATASETINFORMATION and NODATASETINFORMATION: Including or not including data set information.....	933
DATE and VERSION: Specifying the date or version of the aggregate group.....	933
DUMPCONTENTS: Listing contents of a set of dump volumes.....	934
INCLUDEPRIMARY: Listing entries for any data set that has migrated.....	934
OUTDATASET, SYSOUT, and TERMINAL: Specifying the location of output for the list.....	934
RECURSE: Examine subdirectories of the specified UNIX file path.....	935
RESET: Requesting that the host ID field indicate no serialization.....	935
SELECT: Listing only those data set or volume entries that meet selection criteria.....	936
SELECT (ACTIVE and RETAIN_DAYS): Displaying information about active and retained backup copies.....	938
SELECT (AGE): Specifying data set use by age.....	938
SELECT (ALTERNATEVOLUME and NOALTERNATEVOLUME): Listing tape volumes that have alternate tape volumes.....	938
SELECT (AVAILABLE): Listing available dump volumes for dump output.....	939
SELECT (CAPACITYMODE): Identifying tape volumes by their ability to use available tape capacity.....	939
SELECT (CONNECTED and NOTCONNECTED): List volumes that contain spanning data sets .	939
SELECT (DISASTERALTERNATEVOLUMES): Listing tape volumes that have disaster alternate volumes.....	940
SELECT (DUMPCCLASS).....	940
SELECT (DUMPSTATE, FASTREPLICATIONSTATE, INCOMPLETERECOVERY, and NONPREFERREDFRVOLS): Listing specific information about copy pools.....	940
SELECT (EFMT1, EFMT2, EEFMT2, EFMT3, EEFMT3, EFMT4, EEFMT4, ENCRYPTED, and NOTENCRYPTED): Recording format.....	941
SELECT (EMPTY, FULL, NOTFULL, ASSOCIATED, NOTASSOCIATED, ERRORALTERNATE, and RECYCLETAKEAWAY): Identifying tape volume capacity.....	941
SELECT (EXCESSIVEVOLUMES): Listing volumes with data sets spanning four or more volumes.....	942
SELECT (EXPIRED): Listing expired dump volumes.....	942
SELECT (FAILEDCREATE): Listing all tape volumes that have incomplete data set information saved in the OCDS TTOC record.....	943
SELECT (FAILEDRECYCLE): Listing only those volumes that have failed recycle.....	943
SELECT (LIBRARY and NOLIBRARY): Listing tape volume library status.....	943
SELECT (MIGRATIONLEVEL1, MIGRATIONLEVEL2, VOLUME(volser), BACKUP, and BOTH): Requesting specific information for volumes.....	944
SELECT (MULTIPLEVOLUME): Listing multiple-volume non-VSAM data sets from backup or dump VTOC copy data sets.....	944
SELECT (NORETENTIONLIMIT): List dump volumes with no retention period.....	945
SELECT (OVERFLOW and NOOVERFLOW): List overflow or non-overflow ML1 volumes.....	945
SELECT (RECALLTAKEAWAY): Listing ML2 volumes taken away by recall or ABACKUP functions.....	945

SELECT (RESUMEAUDIT): Listing tape volumes that are capable of being resumed after AUDIT MEDIACONTROLS was interrupted.....	945
SELECT (SMALLDATASETPACKING and NOSMALLDATASETPACKING): Specifying whether data sets have migrated to SDSP.....	946
SELECT (UNAVAILABLE): Listing dump volumes with invalid contents.....	946
SELECT (UNEXPIRED): Listing dump volumes that have not reached their expiration date.....	946
SELECT (VSAM): Specifying the data set organization.....	947
SUMMARY: Requesting a summary.....	947
Examples of how to code commonly used list requests.....	947
Chapter 54. LOG command: Entering data into the DFSMSHsm log.....	953
Syntax of the LOG command.....	953
Required parameters of the LOG command.....	953
data: Specifying the data for the log.....	953
Example of how to code the LOG command.....	953
Chapter 55. MIGRATE command: Requesting a space management function.....	955
Requesting a space management function for SMS-managed data sets.....	956
Syntax of the MIGRATE command for SMS-managed data sets.....	956
Summary of parameters.....	956
Required parameters of the MIGRATE command for SMS-managed data sets.....	957
DATASETNAME: Specifying migration and/or class transition for a specific data set.....	958
MIGRATIONLEVEL1: Specifying migration of data sets from level 1 to level 2 volumes.....	958
VOLUME: Specifying the space management attribute for the eligible data sets on a volume.	958
STORAGEGROUP: Specifying the space management attribute for the eligible data sets in a storage group.....	959
Optional parameters of the MIGRATE command for SMS-managed data sets.....	960
BOTH: Specifying that transition or migration of data sets should be processed.....	960
CONVERT: Specifying movement of data sets from one volume to another.....	960
CLOUD: Specifying migration of a data set directly to cloud.....	960
DAYS: Specifying migration eligibility of data sets.....	961
KEEPMIGDATE: Specifying migration of a data set should keep its original migration date, if one exists.....	961
MIGRATIONLEVEL1: Specifying migration of a data set.....	962
MIGRATIONLEVEL2: Specifying migration of a data set directly to a level 2 volume.....	962
MIGRATIONONLY: Specifying that only migration of data sets should be processed.....	962
MOVE: Specifying that only moving of data sets should be processed.....	962
TERMINAL: Requesting that volume space management messages appear at the terminal....	963
TRANSITION: Specifying that only transitions of data set should be processed.....	963
TRANSITIONONLY: Specifying that only transition of data sets should be processed.....	963
Examples of how to code the MIGRATE command for SMS-managed data sets.....	964
Requesting a space management function for non-SMS-managed data sets.....	964
Syntax of the MIGRATE command for non-SMS-managed data sets.....	965
Summary of parameters.....	965
Required parameters of the MIGRATE command for non-SMS-managed data sets.....	966
DATASETNAME: Specifying migration of a specific data set.....	966
MIGRATIONLEVEL1: Specifying migration of data sets from level 1 to level 2 volumes.....	966
PRIMARY: Specifying space management of eligible data sets on all primary volumes.....	966
VOLUME: Specifying the space management attribute for the eligible data sets on a volume.	967
Optional parameters of the MIGRATE command for non-SMS-managed data sets.....	968
CONVERT: Specifying movement of data sets from one volume to another.....	968
DAYS: Specifying migration eligibility of data sets.....	969
MIGRATIONLEVEL1: Specifying migration of a data set directly to appropriate level of hierarchy.....	970
MIGRATIONLEVEL2: Specifying migration of a data set directly to a level 2 volume.....	970
TERMINAL: Requesting that volume space management messages appear at the terminal....	970
UNIT: Specifying the type of device.....	970
Examples of how to code the MIGRATE command for non-SMS-managed data sets.....	971

Chapter 56. ONLYIF command: Running commands in ARCCMDxx conditionally.....	973
Syntax of the ONLYIF command.....	973
Required parameters of the ONLYIF command.....	974
HSMHOST: Specifying the host ID of a specific host or hosts.....	974
SYSNAME: Specifying the system name where the next line should be processed.....	974
HMSYM: Specifying the HSM system symbol and its value where the next line should be processed.....	974
Optional parameters of the ONLYIF command.....	975
BEGIN: Begin a group of commands.....	975
END: End a group of commands.....	975
Examples of how to code the ONLYIF command.....	975
Chapter 57. PATCH command: Modifying storage within DFSMSHsm.....	979
Chapter 58. QUERY command: Displaying the status of parameters, statistics, and requests.....	981
Syntax of the QUERY command.....	982
Optional parameters of the QUERY command.....	983
ABARS: Displaying aggregate backup and recovery control parameters.....	983
ACTIVE: Displaying all active requests.....	983
ARPOOL: Displaying aggregate recovery pools.....	983
AUTOPROGRESS: Displaying an indication of volumes to be processed.....	984
BACKUP: Displaying backup parameters and backup volumes.....	985
CDSVERSIONBACKUP: Displaying the control data set multiple backup support parameters.....	985
COMMONQUEUE: Displaying common queue information.....	985
CONTROLDATASETS: Displaying control data set space-use information.....	986
COPYPOOL: Displaying copy pool information.....	986
CSALIMITS: Displaying the common service area storage limits.....	986
DATASETNAME, FILENAME, REQUEST, and USER: Displaying selected requests.....	986
IMAGE: Displaying information for the DFSMSHsm address spaces active within a single z/OS image.....	987
MIGRATIONLEVEL2: Displaying key ranges and migration level 2 volumes.....	987
NETWORK: Displaying cloud endpoint statistics known to DFSMSHsm.....	987
POOL: Displaying recall pools.....	987
RETAIN: Displaying data set levels prevented from space management.....	987
SECURITY: Displaying the type of command authorization in effect.....	987
SETSYS: Displaying control parameters.....	988
SPACE: Displaying volume space-use information.....	988
STARTUP: Displaying the startup information.....	988
STATISTICS: Displaying daily statistics.....	988
TRAPS: Displaying trap requests.....	988
VOLUMEPOOL: Displaying current volume pools.....	988
WAITING: Displaying the number of requests waiting to be processed.....	988
Examples of how to code the QUERY command.....	989
Chapter 59. RECALL command: Recalling a specific migrated data set.....	991
Syntax of the RECALL command.....	991
Required parameters of the RECALL command.....	991
dsname: Specifying the name of the data set to be recalled.....	991
Optional parameters of the RECALL command.....	992
DAOPTION: Selecting target volume track length.....	992
DFDSSOPTION: Pass optional parameters to DFSMSDss.....	993
FORCENONSMS: Recalling an SMS-managed data set to a non-SMS-managed volume.....	993
UNIT: Specifying the type of device.....	993
VOLUME: Specifying the volume to receive the recalled data set.....	994
Examples of how to code the RECALL command.....	994

Chapter 60. RECOVER command: Recovering a backup version or dump copy of a data set, or UNIX file, or a volume.....	997
Syntax of the RECOVER command.....	998
Required parameters of the RECOVER command.....	1000
<i>dsname</i> : Specifying the name of the data set to be recovered.....	1000
<i>pathname</i> : Specifying the name of the UNIX file to be recovered.....	1001
Optional parameters of the RECOVER command.....	1001
DAOPTION: Selecting target volume track length.....	1001
DATE, GENERATION, and VERSION: Specifying the particular backup version or dump copy to recover.....	1002
DFDSSOPTION: Pass optional parameters to DFSMSdss.....	1003
EXCLUDE and EXCLUDEF: Specifying files or directories to be excluded from processing.....	1004
FORCENONSMS: Recovering an SMS-managed data set to a non-SMS-managed volume.....	1005
FROMDUMP: Specifying which dump copy is used for restoring data sets.....	1005
FROMVOLUME: Specifying the volume from which the backup version of a data set has been created.....	1007
NEWDIR : Recover files to a different directory.....	1007
NEWNAME: Specifying a new data set name for the recovered data set.....	1007
RECURSE: Examine subdirectories of the specified UNIX file path.....	1008
REPLACE: Specifying that the recovered backup version or dump copy is to replace any existing data set of the same name.....	1009
RSA: Specifying that the RSA parameter is to override the RSA key label	1009
TARGETVOLUME: Specifying the volume to receive the restored data sets.....	1009
TERMINAL: Requesting that volume recovery messages appear at the terminal.....	1010
TOVOLUME: Specifying the volume to receive the recovered data set.....	1010
UNIT: Specifying the type of device.....	1011
Examples of how to code the RECOVER command.....	1011
Chapter 61. RECYCLE command: Consolidating valid data to one tape from other tapes.....	1015
Syntax of the generic RECYCLE command.....	1016
Syntax of the specific RECYCLE command.....	1016
Required parameters of the RECYCLE command.....	1017
ALL, BACKUP, DAILY, ML2, SPILL and VOLUME: Specifying the tape volumes to be recycled	1017
EXECUTE, VERIFY, and DISPLAY: Specifying whether to list or to recycle tape volumes.....	1018
Optional parameters of the RECYCLE command.....	1019
CHECKFIRST: Allowing connected set to be recycled sooner.....	1019
FORCE: Forcing DFSMSHsm to permit a tape recycle.....	1019
LIMIT: Quiescing recycle processing when the desired number of tapes have been returned to scratch.....	1020
OUTDATASET: Directing recycle messages.....	1020
PERCENTVALID: Specifying the percent valid criteria for recycle eligibility.....	1021
SELECT: Specifying tape eligibility for generic recycle.....	1022
TAPELIST: Listing volumes for pulling and mounting tapes.....	1022
Examples of how to code the RECYCLE command.....	1025
Chapter 62. RELEASE command: Releasing all or some DFSMSHsm functions for processing.....	1029
Syntax of the RELEASE command.....	1030
Optional parameters of the RELEASE command.....	1031
ABACKUP: Releasing aggregate backup.....	1031
ALL: Releasing all DFSMSHsm functions.....	1031
ARECOVER: Releasing aggregate recovery.....	1031
AUDIT: Releasing the audit function.....	1032
AUTOEXPIRATION: Releasing automatic expiration.....	1032
AUTOMIGRATION: Releasing only automatic volume and automatic secondary space management.....	1032
BACKUP: Releasing backup.....	1032
COMMONQUEUE: Releasing holds on the common queue.....	1033

DUMP: Releasing dump.....	1034
EVENTDRIVENMIGRATION: Releasing Interval and On-demand Migration.....	1034
EXPIREBV: Releasing the expire backup versions function.....	1035
FRBACKUP: Releasing the fast replication backup or dump function.....	1035
FRRECOV: Releasing the fast replication recover function.....	1035
HARDCOPY: Releasing the activity logs.....	1036
LIST: Releasing the list function.....	1036
LOG: Releasing logging.....	1036
MIGRATION: Releasing space management.....	1037
RECALL: Releasing recall and deletion.....	1037
RECOVER: Releasing recovery and restore.....	1038
RECYCLE: Releasing the recycle function.....	1038
REPORT: Releasing the report function.....	1038
RMM: Releasing the DFSMSrmm function.....	1039
TAPECOPY: Releasing the tape copy function.....	1039
TAPEREMPL: Releasing the tape replace function.....	1039
Examples of how to code the RELEASE command.....	1039
 Chapter 63. REPORT Command: Requesting reports based on DFSMSHsm statistics records.....	1043
Syntax of the REPORT command.....	1044
Summary of parameters.....	1045
Optional parameters of the REPORT command.....	1045
DAILY: Requesting daily statistics.....	1045
DELETE: Deleting statistics records used as input to the report.....	1046
FROMDATE: Requesting statistics of activity on or after the specified date.....	1046
FUNCTION and NOFUNCTION: Requesting statistics by function.....	1046
FUNCTION(BACKUP): Requesting statistics of the backup function.....	1047
FUNCTION(DELETE): Requesting statistics of the delete function.....	1047
FUNCTION(MIGRATION): Requesting statistics of the migration function.....	1047
FUNCTION(RECALL): Requesting statistics of the recall function.....	1048
FUNCTION(RECYCLE): Requesting statistics of the recycle function.....	1048
FUNCTION(RECOVER): Requesting statistics of the recovery function.....	1048
FUNCTION(SPILL): Requesting statistics of the spill function.....	1049
FUNCTION(TRANSITION): Requesting statistics of the transition function.....	1049
OUTDATASET and SYSOUT: Specifying the output location.....	1049
SUMMARY: Requesting only the totals of the statistics.....	1050
TODATE: Requesting statistics of activity on or before the specified date.....	1050
VOLUMES: Requesting volume statistics.....	1050
Examples of how to code the REPORT command.....	1051
 Chapter 64. SETMIG command: Changing the space management status of non-SMS-managed data sets or volumes.....	1053
Syntax of the SETMIG command.....	1053
Required parameters of the SETMIG command.....	1053
DATASETNAME, LEVEL, and VOLUME: Changing the space management status.....	1054
Optional parameters of the SETMIG command.....	1054
COMMANDMIGRATION, MIGRATION, and NOMIGRATION: Specifying the way space management is controlled.....	1054
Examples of how to code the SETMIG command.....	1055
 Chapter 65. SETSYS command: Establishing or changing the values of DFSMSHsm control parameters.....	1057
Syntax of the SETSYS command.....	1057
SETSYS fast-path for DFSMSHsm functions.....	1075
Relationship of SMS to SETSYS parameters.....	1082
Summary of DFSMSHsm default values.....	1084
Optional parameters of the SETSYS command.....	1094
ABARSACTLOGMSGSLVL: Specifying which DFSMSdss messages are written to the log.....	1094

ABARSACTLOGTYPE: Specifying the ABARS activity log output data set type.....	1094
ABARSBUFFERS: Specifying the number of I/O buffers for ABARS.....	1095
ABARSDELETEACTIVITY: Allowing DFSMSHsm to automatically delete the ABARS activity log.....	1095
ABARSKIP: Specifying which types of data sets ABACKUP should skip.....	1095
ABARSOPTIMIZE: Adjusting performance when backing up level 0 DASD specified in INCLUDE list.....	1096
ABARSPROCNAME: Specifying the base procedure name to start the ABARS secondary address space.....	1096
ABARSTAPES: Allowing DFSMSHsm to stack the ABACKUP output files.....	1096
ABARSUNITNAME: Specifying a unit name for ABARS.....	1097
ABARSVOLCOUNT: Specifying the method for determining volume count for SMS target data set allocations done by DFSMSdss.....	1097
ACCEPTPSCBUSERID and NOACCEPTPSCBUSERID: Allowing non-RACF installations to issue authorized commands in a TMP environment.....	1098
ACTLOGMSGVL: Specifying message level for recording data sets.....	1098
ACTLOGTYPE: Specifying the type of output data set to receive messages.....	1099
ARECOVERML2UNIT: Specifying the unit when recovering migrated data sets to nonspecific volumes.....	1100
ARECOVERPERCENTUTILIZED: Specifying the percentage of DASD space for non-SMS-managed volumes.....	1100
ARECOVERGTGDS: Passing GTGDS parameter to DFSMSdss during restore processing....	1100
ARECOVERUNITNAME: Specifying tape unit type for aggregate recovery.....	1100
AUTOBACKUPSTART: Specifying when automatic backup is to start.....	1101
AUTODUMPSTART: Specifying when automatic dump is to start.....	1102
AUTOMIGRATIONSTART: Specifying the start time for automatic primary space management.....	1103
BACKUP and NOBACKUP: Specifying whether you intend to use DFSMSHsm backup, recovery, and dump.....	1104
BACKUPPREFIX: Specifying the prefix for the backup version name or the dump copy data set name.....	1107
CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets.....	1109
CDSVERSIONBACKUP (BACKUPCOPIES): Specifying the number of backup copies to keep for each control data set and journal data set.....	1110
CDSVERSIONBACKUP (BACKUPDEVICECATEGORY): Specifying the backup device to which the control data sets and journal will be backed up.....	1110
CDSVERSIONBACKUP (DATAMOVER): Specifying the type of data movement used when performing CDSVERSIONBACKUP.....	1112
CDSVERSIONBACKUP (BCDSBACKUPDSN): Specifying the initial characters of the backup data set name for the backup control data set.....	1112
CDSVERSIONBACKUP (MCDSBACKUPDSN): Specifying the initial characters of the backup data set name for the migration control data set.....	1113
CDSVERSIONBACKUP (OCDSBACKUPDSN): Specifying the initial characters of the backup data set name for the offline control data set.....	1113
CDSVERSIONBACKUP (JRNLBKUPDSN): Specifying the initial characters of the backup data set name for the journal data set.....	1114
CLASSTRANSITIONS: Specifying if class transitions will be performed during event-driven migration.....	1114
CLOUDMIGRATION: Specifying the type of migration to CLOUD.....	1114
CLOUD: Specifying CLOUD properties.....	1115
COMMONQUEUE: Specifying settings for the common queues.....	1117
COMMONQUEUE (DUMP(CONNECT(<i>basename</i>))) : Specifying the XCF functional group name suffix for remote dump processing.....	1117
COMMONQUEUE(DUMP(DISCONNECT)): Specifying that a system disconnect itself from using the currently connected XCF functional group for dump processing.....	1118
COMMONQUEUE (DUMP(MASTERSCHEDULERCANDIDATE)): Specifying whether the host is a master scheduler candidate for grouped function of dump.....	1118

COMMONQUEUE (RECALL(CONNECT(<i>basename</i>))): Specifying the base name of the list structure for CRQ host connection.....	1119
COMMONQUEUE (RECALL(DISCONNECT)): Specifying disconnection from the CRQ.....	1119
COMMONQUEUE RECOVER(CONNECT(<i>basename</i>)): Specifying the XCF functional group name suffix for remote full volume recover from dump processing.....	1120
COMMONQUEUE RECOVER(DISCONNECT): Specifying that a system disconnect itself from using the currently connected XCF functional group for recover processing.....	1121
COMMONQUEUE RECOVER(MASTERSCHEDULERCANDIDATE): Specifying whether the host is a master scheduler candidate for grouped function of recover.....	1121
COMPACT: Specifying when compaction should be done.....	1122
COMPACT ALL and NONE: Specifying whether to compact data sets.....	1122
COMPACT DASDBACKUP and NODASDBACKUP: Specifying whether to compact data sets during backup to DASD.....	1123
COMPACT DASDMIGRATE and NODASDMIGRATE: Specifying whether to compact data sets during migration to DASD.....	1123
COMPACT TAPEBACKUP and NOTAPEBACKUP: Specifying whether to compact data sets during backup to tape.....	1124
COMPACT TAPEMIGRATE and NOTAPEMIGRATE: Specifying whether to compact data sets during migration to tape.....	1124
COMPACTPERCENT: Specifying the percentage of space saved for compaction.....	1125
CONVERSION and NOCONVERSION: Specifying whether reblocking is to occur.....	1125
CSALIMITS and NOCSALIMITS: Specifying whether to use CSA limiting for MWE allocation.....	1126
CSALIMITS (ACTIVE): Specifying the percentage of the maximum limit to be allocated to any type of MWE.....	1126
CSALIMITS (INACTIVE): Specifying the percentage of the maximum limit to be allocated to NOWAIT MWEs.....	1127
CSALIMITS (MAXIMUM): Specifying the maximum number of CSA bytes allocated to all types of MWEs.....	1127
CSALIMITS (MWE): Specifying maximum number of NOWAIT MWEs per address space.....	1127
DAYS: Specifying the minimum migration age of a data set.....	1128
DEBUG and NODEBUG: Specifying debug mode.....	1128
DEFERMOUNT: Specifying defer mounting of tapes.....	1129
DENSITY: Specifying the tape density.....	1129
DFHSMDATASETSERIALIZATION and USERDATASETSERIALIZATION: Specifying data set serialization.....	1129
DISASTERMODE: Specifying the tape operation mode for DFSMSHsm in disaster mode.....	1130
DSBACKUP: Controlling the command data set backup environment.....	1130
DSSXMMODE: Specifying whether DFSMSdss will be loaded in its own address space using the cross memory interface, or in the DFSMSHsm address space, for backup, CDS backup, dump, migration, and full-volume and data set recovery.....	1133
DUMPIO: Specifying the DFSMSdss DASD I/O buffering technique.....	1134
DUPLEX: Allowing DFSMSHsm to create two tapes concurrently.....	1135
EMERGENCY and NOEMERGENCY: Specifying no data set movement.....	1136
ERASEONSCRATCH and NOERASEONSCRATCH: Specifying to check for erasure of DFSMSHsm-owned data sets.....	1137
EVENTDRIVENMIGRATIONQUIESCE: Specifying quiesce window(s) for Interval and On-demand Migration.....	1138
EXITOFF: Specifying inactive installation exits.....	1138
EXITON: Specifying active installation exits.....	1139
EXITS: Specifying the installation exits to be taken.....	1141
EXPIREDDATASETS: Specifying whether to scratch data sets with expired expiration dates.....	1141
EXTENDEDTTTC: Specifying whether to use extended tape table of contents (TTTC).....	1141
FASTREPLICATION: Specifying fast replication methods.....	1142
FREQUENCY: Specifying the number of days between backup versions of a data set.....	1143
INCREMENTALBACKUP: Specifying whether to back up only changed data sets during incremental volume backup.....	1144
INPUTTAPEALLOCATION: Specifying whether to wait for the input tape to be allocated.....	1145

INTERVALMIGRATION and NOINTERVALMIGRATION: Specifying whether interval migration is performed.....	1145
ISSUEARC1902: Enable/disable the issue of ARC1902I message to the console.....	1146
JES2 and JES3: Specifying the job entry subsystem.....	1147
JOURNAL and NOJOURNAL: Specifying whether to write updated control data set records in the journal data set.....	1147
MAXABARSADDRESSSPACE: Specifying the maximum number of concurrent secondary address spaces supported.....	1148
MAXBACKUPTASKS: Specifying the maximum number of concurrent volume backup tasks.....	1149
MAXCOPYPOOLTASKS: Specifying the maximum number of concurrent DFSMSdss invocations for each fast replication command.....	1149
MAXDSRECOVERTASKS: Specifying the maximum number of data set recovery tasks from DASD and tape.....	1150
MAXDSTAPERECOVERTASKS: Specifying the maximum number of data set recovery tasks from tape.....	1150
MAXDUMPRECOVERTASKS: Specifying the maximum number of concurrent volume recovery from dump tasks.....	1151
MAXDUMPTASKS: Specifying the maximum number of concurrent volume dump tasks.....	1151
MAXEXTENTS: Specifying the number of extents to cause extent reduction.....	1152
MAXINTERVALTASKS: Specifying the maximum number of automatic interval migration or on-demand migration tasks that can run concurrently.....	1152
MAXMIGRATIONTASKS: Specifying the maximum number of automatic volume space management tasks that can run concurrently.....	1153
MAXRECALLTASKS: Specifying the maximum number of active recall tasks.....	1153
MAXRECYCLETASKS: Specifying the maximum number of active recycle tasks.....	1154
MAXSSMTASKS: Specifying the maximum number of concurrent automatic secondary space management tasks.....	1154
MIGDENSITY: Specifying the tape density when allocating scratch tapes for migration.....	1155
MIGRATEPREFIX: Specifying the prefix for the migration copy name.....	1155
MIGRATIONAUTOCLLOUD: Allow the host to execute Cloud migration.....	1156
MIGRATIONCLEANUPDAYS: Specifying the number of days to keep MCDS data set records and statistics records.....	1157
MIGRATIONLEVEL1DAYS: Specifying the inactive age for data sets migrating from level 1 volumes.....	1158
MIGRATEPSMEARLY: Activate the PSM earlier checking option.....	1158
MIGRATIONSUBTASKS: Enabling multiple migration subtasks.....	1159
MIGUNITNAME: Specifying the type of unit for allocating a scratch tape during migration...	1159
ML1OVERFLOW: Manage ML1 OVERFLOW volumes.....	1160
ML2PARTIALSNOTASSOCIATEDGOAL: Specifying the maximum number of allowed partial ML2 tapes.....	1161
ML2RECYCLEPERCENT: Specifying the maximum percentage of valid data on a migration tape.....	1161
MONITOR: Specifying which informational messages to print at the system console.....	1162
MONITOR (BACKUPCONTROLDATASET): Monitoring the space used in the BCDS.....	1163
MONITOR (JOURNAL): Monitoring the space used in the journal data set.....	1163
MONITOR (MIGRATIONCONTROLDATASET): Monitoring the space used in the MCDS.....	1163
MONITOR (OFFLINECONTROLDATASET): Monitoring the space used in the OCDS.....	1164
MONITOR (SPACE and NOSPACE): Printing volume space-use messages.....	1164
MONITOR (STARTUP and NOSTARTUP): Printing DFSMSHsm startup informational messages.....	1164
MONITOR (VOLUME and NOVOLUME): Print data set messages during volume processing...	1165
MOUNTWAITTIME: Specifying the time DFSMSHsm waits for a tape mount and open.....	1165
OBJECTNAMES: Specifying the compaction-control qualifier of the object data set.....	1166
ONDEMANDMIGRATION: Specifying whether on-demand migration should be performed..	1166
ODMNOTIFICATIONLIMIT: Specifying an on-demand migration number of concurrently queued volumes notification limit.....	1167
OPTIMUMDASDBLOCKING and NOOPTIMUMDASDBLOCKING: Specifying whether DFSMSHsm should use its defined optimum block size.....	1167

OUTPUTTAPEALLOCATION: Specifying if to wait for output tape to be allocated.....	1168
PARTIALTAPE: Specifying whether to reuse partially full tapes.....	1168
PDA: Specifying whether the problem determination aid is initiated.....	1169
PLEXNAME: Specifying a name for an HSMplex	1170
PRIMARYSPMGMTSTART: Specifying the start and stop times for automatic primary space management.....	1171
PROFILEBACKUP and NOPROFILEBACKUP: Specifying whether to create backup RACF discrete profiles for cataloged data sets.....	1172
PROMOTE: Specifying whether a host is eligible for promotion.....	1172
RACFIND and NORACFIND: Specifying whether to put RACF-indication on migration copies and backup versions.....	1173
RECALL: Specifying the target volume to receive recalled data sets.....	1173
RECYCLEINPUTDEALLOCFREQUENCY: Allowing periodic deallocation of input unit during recycle processing.....	1174
RECYCLEOUTPUT: Limiting the selection and allocation of an output volume during recycle processing.....	1176
RECYCLEPERCENT: Specifying the maximum percentage of valid data on a backup tape.....	1176
RECYCLETAKEAWAYRETRY: Specifying whether to automatically generate a new RECYCLE command for a tape.....	1177
RECYCLETAPEALLOCATION: Specifying if to wait for recycle tape to be allocated.....	1177
RECYCLETAPERESIDENCE: Specifying which IBM VTS tape volumes should be included in RECYCLE processing.....	1178
REMOVECOMPACTNAMES: Removing the compaction-control qualifier for source or object data sets.....	1178
REQUEST and NOREQUEST: Specifying whether the operator's permission is required.....	1179
SCRATCHFREQUENCY: Specifying how long to retain list data sets.....	1179
SECONDARYSPMGMTDIRECTIVE: Specifying Secondary Space Management controls pertaining to Cloud and non-Cloud data.....	1179
SECONDARYSPMGMTSTART: Specifying the start and stop time for automatic secondary space management.....	1180
SELECTVOLUME: Specifying whether DFSMSHsm should select a scratch tape volume or a DFSMSHsm tape volume.....	1181
SKIPABPRIMARY and NOSKIPABPRIMARY: Specifying whether to skip automatic backup of primary volumes.....	1183
SMALLDATASETPACKING and NOSMALLDATASETPACKING: Specifying whether DFSMSHsm should use small data set packing.....	1183
SMF and NOSMF: Specifying whether DFSMSHsm writes SMF records.....	1185
SOURCENAMES: Specifying the compaction-control qualifier of the source data set.....	1185
SPIII and NOSPIII: Specifying whether DFSMSHsm does spill or cleanup processing on full DASD daily backup volumes.....	1186
STORAGEGROUFAFFINITY: Define the set of host names that have affinity to process a storage group cloud migration.....	1187
SWAP and NOSWAP: Specifying whether the DFSMSHsm address space can be swapped....	1187
SYSOUT: Directing DFSMSHsm output.....	1187
SYS1DUMP and NOSYS1DUMP: Specifying where DFSMSHsm dumps should be written.....	1188
TAPEDATASETORDER: Specifying order for processing tape.....	1188
TAPEDELETION: Specifying what to do with tapes that no longer contain valid data.....	1189
TAPEHARDWARECOMPACT and NOTAPEHARDWARECOMPACT: Specifying the data format used when writing to a 3480X tape device.....	1190
TAPEINPUTPROMPT: Specifying whether to suppress input tape messages.....	1191
TAPELIBRARYPRESELECT: Preselecting output tape volumes in IBM Tape Library tape environment.....	1191
TAPEMAXRECALLTASKS: Specifying the maximum number of concurrent tape recall tasks..	1192
TAPEMIGRATION: Specifying the type of migration to tape.....	1192
TAPEOUTPUTPROMPT: Specifying whether to suppress output tape messages.....	1194
TAPERECALLLIMITS: Specifying limits to continuously mounted ML2 tapes during multiple recalls.....	1195
TAPESECURITY: Specifying the type of security for tape volumes.....	1195

TAPESPANSIZE: Reducing tape volume spanning.....	1197
TAPETYPE: Specifying whether to use WORM media for non-ABARS functions.....	1198
TAPEUTILIZATION: Specifying the desired tape utilization.....	1198
UNITNAME: Specifying the type of unit for mounting a scratch tape during backup or dump.....	1201
UNLOAD and NOUNLOAD: Specifying whether to unload MSS virtual DASD volumes.....	1202
USECYLINDERMANAGEDSPACE: Specifying whether to allow use of cylinder managed space for migration copies and backup versions.....	1202
USERUNITTABLE and NOUSERUNITTABLE: Specifying esoteric tape unit names to DFSMSHsm.....	1202
VERSIONS: Specifying the maximum number of backup versions to keep for any data set...	1205
VOLCOUNT: Specifying the method for determining volume count for SMS target data set allocations when performed by DFSMSdss.....	1206
VOLUMEDUMP: Specifying concurrent copy options for dump volumes.....	1206
ZCOMPRESS: Specifying when compression with zEDC should be done.....	1207
ZCOMPRESS (NONE ALL): Specifying whether to use zEDC Services for data sets.....	1208
ZCOMPRESS(CLOUDMIGRATE(NO YES)): Specifying whether to use CDA compression for data sets during migration to DIRECT cloud connections.....	1208
ZCOMPRESS (DASDBACKUP (NO YES)) : Specifying whether to use zEDC Services for data sets during backup to DASD.....	1209
ZCOMPRESS (DASDMIGRATE (NO YES)) : Specifying whether to use zEDC Services for data sets during migration to DASD.....	1210
ZCOMPRESS (TAPEBACKUP (NO YES)) : Specifying whether to use zEDC Services for data sets during backup to tape.....	1210
ZCOMPRESS (TAPEMIGRATE (NO YES)) : Specifying whether to use zEDC Services for data sets during migration to tape.....	1211
Examples of how to code the SETSYS command.....	1211
Chapter 66. STOP command: Shutting down DFSMSHsm.....	1223
Syntax of the STOP command.....	1223
Optional parameters of the STOP command.....	1223
DUMP: Requesting a dump of the DFSMSHsm address space.....	1223
PROMOTE: Specifying that a specific host in a sysplex is eligible to be promoted for another host.....	1224
Example of how to code the STOP command.....	1224
Chapter 67. SWAPLOG command: Switching between the DFSMSHsm log or PDA data sets.....	1225
Syntax of the SWAPLOG command.....	1225
Optional parameters of the SWAPLOG command.....	1225
LOG: Specifying to swap the log data sets.....	1225
PDA: Specifying to swap the PDA data sets.....	1226
Examples of how to code the SWAPLOG command.....	1226
Chapter 68. TAPECOPY command: Making copies of selected cartridge-type tape volumes.....	1227
Syntax of the TAPECOPY command.....	1227
Required parameters of the TAPECOPY command.....	1228
ALL: Specifying to copy all eligible cartridge-type single-file tape volumes.....	1228
MIGRATIONLEVEL2: Specifying to create copies of all eligible migration level 2 volumes.....	1228
BACKUP: Specifying to create copies of all eligible backup tape volumes.....	1228
ORIGINALVOLUMES: Copying specified volumes.....	1228
INDATASET: Specifying to use a data set containing a volume list.....	1228
Optional parameters of the TAPECOPY command.....	1230
ALTERNATEUNITNAME: Specifying alternate unit names for a tape copy.....	1230
ALTERNATE3590UNITNAME: Specifying 3590 alternate unit name for a tape copy.....	1231
ALTERNATEVOLUMES: Specifying which volumes to copy.....	1231
EXPDT and RETPD: Specifying a tape expiration date or a retention period.....	1232
Examples of how to code the TAPECOPY command.....	1233

Chapter 69. TAPEREP command: Replacing cartridge-type tape volumes with their alternate volumes.....	1235
Syntax of the TAPEREP command.....	1235
Required parameters of the TAPEREP command.....	1235
ALL: Specifying replacement of all cartridge-type single-file tape volumes.....	1235
BACKUP: Specifying replacement of all cartridge-type single-file backup tape volumes.....	1235
INDATASET: Specifying the use of a data set containing a volume list.....	1236
MIGRATION: Specifying replacement of all cartridge-type single-file migration tape volumes.....	1236
ONLYDISASTERALTERNATES: Converting disaster alternate volumes to original volumes....	1237
ORIGINALVOLUMES: Specifying volumes to be replaced.....	1237
Optional parameters of the TAPEREP command.....	1237
ALTERNATEUNITNAME: Specifying an alternate unit name for a tape replacement.....	1237
ALTERNATEVOLUMES: Specifying the replacement volumes to use.....	1238
DISASTERALTERNATEVOLUMES: Identifying the alternate volume as the disaster alternate volume.....	1238
Examples of how to code the TAPEREP command.....	1238
Chapter 70. TRAP command: Requesting a dump when a specified error occurs.....	1241
Chapter 71. UPDATEC command: Recovering the control data sets.....	1243
Syntax of the UPDATEC command.....	1243
Required parameters of the UPDATEC command.....	1243
ALL, BACKUPCONTROLDATASET, MIGRATIONCONTROLDATASET, and OFFLINECONTROLDATASET: Specifying which control data set to update.....	1243
Optional parameters of the UPDATEC command.....	1244
JOURNAL: Specifying the journal data set to use.....	1244
Examples of how to code the UPDATEC command.....	1244
Chapter 72. Storage administrator mode for UNIX commands.....	1245
Chapter 73. Java <i>hbackup</i> command for Incremental UNIX file backup.....	1247
Syntax of the Java <i>hbackup</i> command.....	1247
Required parameters of the Java <i>hbackup</i> command	1247
directory_name: Specifying the name of the directory.....	1247
Optional parameters of the Java <i>hbackup</i> command.....	1247
Specifying the environment configuration.....	1250
Example of how to code the Java <i>hbackup</i> command	1251
Chapter 74. UPDTCDS Command: Updating the control data sets.....	1253
Syntax of the UPDTCDS command.....	1253
Required parameters of the UPDTCDS command.....	1253
COPYPOOL: Specifying the name of the copy pool.....	1253
VERSION: Specifying the version.....	1253
DUMPEXPIRATION: Indicating that the copy pool dump expiration is to be modified.....	1253
Example of how to code the UPDTCDS command.....	1254
Chapter 75. Using the AUDIT command.....	1255
Using the enhanced audit command to audit DFSMSshm control information.....	1256
Using the AUDIT ABARSCONTROLS command.....	1257
Using the AUDIT COMMONQUEUE command.....	1258
Using the AUDIT COPYPOOLCONTROLS command.....	1259
Using the AUDIT DATASETCONTROLS (BACKUP) command.....	1260
Using the AUDIT DATASETCONTROLS (MIGRATION) command.....	1261
Using the AUDIT DIRECTORYCONTROLS command.....	1263
Using the AUDIT FILECONTROLS(BACKUP) command.....	1266
Using the AUDIT MEDIACONTROLS (CLOUD) command.....	1267

Using the AUDIT MEDIACONTROLS (SDSP) VOLUMES command.....	1268
Using the AUDIT MEDIACONTROLS VOLUMES command.....	1269
Using the AUDIT VOLUMECONTROLS (BACKUP) command.....	1272
Using the AUDIT VOLUMECONTROLS (MIGRATION) command.....	1273
Using the AUDIT VOLUMECONTROLS (RECOVERABLE) command.....	1274
Summary of errors detected by the enhanced AUDIT commands.....	1276
Using the original AUDIT command to audit DFSMSHsm data set information.....	1280
Using the AUDIT DATASET NAMES and AUDIT LEVELS commands.....	1282
Using the AUDIT VOLUMES command.....	1284
Using the AUDIT BACKUPVOLUMES and AUDIT BACKUPTYPE commands.....	1286
Using the AUDIT MASTERCATALOG and AUDIT USERCATALOG commands.....	1288
Using the AUDIT MIGRATIONCONTROL DATASET command.....	1290
Using the AUDIT BACKUPCONTROL DATASET command.....	1291
Using the AUDIT OFFLINECONTROL DATASET command.....	1292
Summary of errors detected by the original AUDIT commands.....	1294
Error codes (*ERR) and diagnosis.....	1295
Unexpected errors during an audit.....	1337
Unexpected error messages.....	1337
Errors encountered during AUDIT VOLUMES, AUDIT BACKUPVOLUMES, or AUDIT BACKUPTYPE.....	1339
Other considerations when using the AUDIT command.....	1340
Multiple DFSMSHsm host considerations when using the AUDIT command.....	1340
Printing information from the AUDIT command.....	1340
Chapter 76. Using the LIST command.....	1341
Requesting information for aggregate backup and recovery activity information.....	1341
Requesting information for backup volumes.....	1345
Requesting information for cloud storage.....	1346
Requesting information for copy pools.....	1348
Requesting information for copy pool backup storage groups.....	1351
Requesting information for data sets.....	1352
Requesting data set information from the MCDS.....	1352
Requesting data set information from the BCDS.....	1355
Requesting data set information from the MCDS and BCDS.....	1359
Requesting data set information for the latest backup version.....	1360
Requesting information for dump classes.....	1361
Requesting information for dump volumes.....	1364
Requesting dump content information for dump volumes.....	1366
Requesting information for primary and migration volumes.....	1371
Requesting primary and migration volume information from the MCDS.....	1371
Requesting primary and migration volume information from the BCDS.....	1376
Requesting primary volume dump information from the BCDS.....	1378
Requesting contents of backup VTOC for a primary volume from the BCDS.....	1379
Requesting a list of records serialized by DFSMSHsm host ID.....	1381
Requesting information from the tape table of contents (TTOC).....	1381
Requesting information for all user entries or a specific user entry.....	1386
Chapter 77. Using the QUERY command.....	1389
Chapter 78. Using the REPORT command.....	1401
Requesting a report of the daily statistics for all functions.....	1401
Requesting a report of the volume statistics for all functions.....	1405
Requesting a summary of the daily or volume statistics.....	1410

Appendix A. Accessibility.....1413

Notices.....1415

Terms and conditions for product documentation.....	1416
IBM Online Privacy Statement.....	1417
Policy for unsupported hardware.....	1417
Minimum supported hardware.....	1417
Programming interface information.....	1418
Trademarks.....	1418
Glossary.....	1419
Index.....	1445

Figures

1. Data Set Flow in DFSMSHsm.....	10
2. Example System of SMS-Managed Volumes.....	17
3. Example System of Non-SMS-Managed Volumes.....	18
4. Comprehensive Example System.....	21
5. Storage Group Attributes for Space Management.....	34
6. Example System of SMS-managed volumes before start of automatic space management.....	57
7. Example System of SMS-Managed Volumes after Migration Cleanup.....	61
8. Example System of SMS-Managed Volumes after Level 1 to Level 2 Migration.....	64
9. Example System of SMS-Managed Volumes after DFSMSHsm-Managed Volume Management Pass 1.....	70
10. Component Name Limitations when Recalling Migrated VSAM Data Sets.....	76
11. Example System of SMS-Managed Volumes after DFSMSHsm-Managed Volume Management Pass 2.....	80
12. Example commands used to migrate data sets from SMS-managed volumes.....	92
13. Multiple-Processing Units in a JES3 and DFSMSHsm Environment.....	108
14. Example System of Non-SMS-Managed Volumes before Automatic Space Management.....	110
15. Example System of Non-SMS-Managed Volumes after Primary Volume Management Pass 1.....	113
16. Component Name Limitations When Recalling Migrated VSAM Data Sets.....	116
17. Example System of Non-SMS-Managed Volumes after Primary Volume Management Pass 2.....	118
18. Example System of Non-SMS-Managed Volumes after Undirected Recall of Data Sets.....	123
19. The Common Recall Queue.....	130
20. SDSP Contention Environment.....	144
21. Part 1—Ideal Multitasking Migration.....	147
22. Part 2—Recall Processing Has a Higher Priority than Migration.....	148

23. Part 3—All SDSP Data Sets Are Full.....	148
24. Storage Group Dump Attributes for Availability Management.....	167
25. COPY POOL DEFINE and COPY POOL ALTER panels, Page 1.....	168
26. DFSMSHsm Automatic Dump Commands for SMS-Managed Storage.....	177
27. Storage Group Attributes for Backup.....	178
28. Backup Attributes for STANDMC.....	186
29. Backup Attributes for GDSMC.....	187
30. Backup Attributes for LARGEMC.....	187
31. Backup Attributes for NOMIGMC.....	188
32. DFSMSHsm Automatic Backup Commands for SMS-Managed Storage.....	193
33. COPY POOL DEFINE/ALTER panel, Page 2.....	263
34. Aggregate Backup and Recovery Support (part 1 of 2).....	320
35. Aggregate Backup and Recovery Support (part 2 of 2).....	321
36. Aggregate Backup Information Flow.....	324
37. Example of an Instruction Data Set.....	335
38. Aggregate Group ISMF Panel 1.....	337
39. Aggregate Group ISMF Panel 2.....	337
40. Aggregate Backup Invocation.....	347
41. Example of an Aggregate Backup Activity Log.....	353
42. Example of DISPLAY ABARSVERSIONS Parameter Output.....	355
43. Stacking Aggregates - Sample JCL (Part 1 of 2).....	363
44. Stacking Aggregates - Sample JCL (Part 2 of 2).....	364
45. Page 5 of the ISMF Management Class Alter panel with # Versions parameter.....	365
46. Sample JCL to recatalog an entire aggregate onto the original ABACKUP file.....	368
47. Sample JCL to copy files from an existing tape to a new scratch tape and new device type, NOSTACK mode.....	370

48. Sample JCL to copy files from an existing tape to a new scratch tape and new device type, after ABACKUP STACK.....	371
49. Aggregate Recovery Information Flow.....	372
50. Example of a Retrieved Instruction Data Set.....	381
51. Aggregate Recovery Invocation.....	392
52. Conflict Resolution Data Set.....	403
53. Aggregate Group Application Selection Panel.....	414
54. Aggregate Group List Panel.....	414
55. Aggregate Group Backup Panel.....	415
56. Aggregate Group Application Selection Panel.....	416
57. Aggregate Group Backup Panel.....	417
58. Aggregate Group Application Selection Panel.....	418
59. Aggregate Group Recovery Panels.....	419
60. Aggregate Recovery Panel DFQDCAG5.....	420
61. Aggregate Recovery Panel DFQDCAG6.....	421
62. Aggregate Recovery Panel DFQDCAG7.....	422
63. Aggregate Recovery Panel DFQDCAG8.....	423
64. Aggregate Recovery Panel DFQDCAG9.....	424
65. Aggregate Recovery Panel DFQDCAGA.....	424
66. Aggregate Recovery Panel DFQDCAGC.....	425
67. Aggregate Recovery Panel DFQDCAGB.....	426
68. NetView FTP Main Task Selection Panel.....	428
69. Main Transfer Parameters Panel.....	428
70. Main Data Set Parameters Panel.....	429
71. Additional Transfer Parameters Panel.....	429
72. Additional Parameters for the Sending Data Set Panel.....	430

73. Example 2: Additional Parameters for the Receiving Data Set Panel.....	430
74. Sample Batch Job Used to Transmit Aggregate Backup Files.....	431
75. Example of Inline Backup Job Stream.....	449
76. Example of Messages Resulting from Inline Backup Job Stream.....	449
77. Common Dump Queue.....	467
78. CDQ -- Flexible Configurations.....	468
79. CDQ Host Types.....	469
80. Common Recover Queue.....	475
81. CVQ -- Flexible Configurations.....	476
82. CVQ Host Types.....	478
83. Example System for MIGRATE CONVERT.....	529
84. SMS Volume Status.....	530
85. DFSMSHsm Log and Edit Log Overview.....	607
86. Example of a List from the ARCPRLOG Program.....	612
87. Example of a List from the ARCPEDIT Program.....	613
88. Example of a Summary List from the ARCPEDIT Program.....	614
89. Handwritten note from Steve to Rick.....	629
90. Consolidated Storage Group Attributes.....	659
91. Sample Activity Log Created When You Specify ABACKUP with the VERIFY Parameter.....	696
92. ARECOVER sample activity log output.....	735
93. Identifying Authorized DFSMSHsm Users.....	757
94. Sample Printer List of Data Sets When You Specify EXPIREBV ABARSVERSIONS and DISPLAY.....	861
95. Sample Printer List of Data Sets When You Specify EXPIREBV and DISPLAY.....	861
96. Sample List of Output for AUDIT DATASETNAMES.....	1283
97. Sample Terminal Output for AUDIT DATASETNAMES.....	1283

98. Sample List of Output for AUDIT VOLUMES.....	1285
99. Sample Terminal Output for AUDIT VOLUMES.....	1285
100. Sample List of Output for AUDIT BACKUPVOLUMES.....	1287
101. Sample Terminal List for AUDIT BACKUPVOLUMES.....	1288
102. Sample List of Output for AUDIT MASTERCATALOG.....	1289
103. Sample List of Output for AUDIT USERCATALOG.....	1289
104. Sample List of Output for AUDIT MCDS.....	1290
105. Sample Terminal List for AUDIT MCDS.....	1291
106. Sample List of Output for AUDIT BCDS.....	1292
107. Sample Terminal List for AUDIT BCDS.....	1292
108. Sample List of Output for AUDIT OCDS(ALL).....	1294
109. Sample Terminal List for AUDIT OCDS(ML2).....	1294
110. Sample Terminal List when You Specify AGGREGATE(PAY1), VERSION(002) and TERMINAL.....	1344
111. Sample Printer List when You Specify AGGREGATE(PAY1), VERSION(001) and ODS.....	1344
112. Sample Printer List of All the Backup Volumes when You Specify BACKUPVOLUME.....	1346
113. Sample Printer List of a specific cloud when you specify the CLOUD(cloud_network_connection_name).....	1347
114. Sample Printer List of specific objects when you specify the CLOUD(cloud_network_connection_name) CONTAINER(container_name).....	1348
115. Sample Printer List of specific objects when you specify the CLOUD(cloud_network_connection_name) CONTAINER(container_name) PREFIX(prefix_name)...	1348
116. Sample Printer List of All the Copy Pools when You Specify COPYPOOL.....	1350
117. Sample Printer List of a Specific Copy Pool when You Specify COPYPOOL(cpname).....	1350
118. Sample Output for the LIST COPYPOOL(cpname) DATASETS command.....	1351
119. Sample printer list of a specific copy pool when you specify the COPYPOOL(cpname) parameter and one of the dump versions resides on cloud object storage.....	1351
120. Sample Printer List for COPYPOOLBACKUPSTORAGEGROUP.....	1352
121. Sample Printer List when You Specify DATASETNAME and MIGRATIONCONTROLDATASET.....	1354

122. Sample Printer List of VSAM Data Sets when You Specify DATASETNAME, SELECT(VSAM), and MIGRATIONCONTROLDATASET.....	1354
123. Sample Printer List when You Specify DATASETNAME and SUMMARY.....	1354
124. Sample Printer List Output for the LIST DATASETNAME(dsname) BACKUPCONTROLDATASET Command.....	1358
125. Sample Terminal Output for the LIST DATASETNAME(dsname) BACKUPCONTROLDATASET Command.....	1358
126. Sample Printer List Output for the LIST LEVEL(qualifier) BACKUPCONTROLDATASET Command..	1359
127. Sample Terminal Output for the LIST LEVEL(qualifier) BACKUPCONTROLDATASET Command.....	1359
128. Sample Printer List of All Data Sets when You Specify DATASETNAME, BOTH, and INCLUDEPRIMARY.....	1360
129. Sample Printer List when you specify DATASETNAME and SELECT(VOLUME(volser)).....	1361
130. Sample Terminal List when you specify DATASETNAME and SELECT(VOLUME(volser)).....	1361
131. Sample Printer List of Dump Classes of Dump Volumes when You Specify DUMPCLASS.....	1363
132. Sample Printer List of a specific dump class when you specify the DUMPCLASS parameter.....	1363
133.	1364
134. Sample Printer List for DUMPVOLUME(volser).....	1366
135. Sample Printer List for all Available Volumes when You Specify SELECT(AVAILABLE).....	1366
136. Sample Printer List of all Dump Volumes when You Specify DUMPVOLUME without (volser).....	1366
137. Sample Printer List when You Specify DUMPVOLUME(volser) and DUMPCONTENTS(srcvol).....	1370
138. Sample Printer List of cloud dump information for a specific source volume when you specify DUMPVOLUME(%CLOUD) and DUMPCONTENTS(srvol).....	1370
139. Sample Printer List of Primary Volumes from the MCDS when You Specify PRIMARYVOLUME and MIGRATIONCONTROLDATASET.....	1374
140. Sample Printer List of Migration Volumes from the MCDS when You Specify MIGRATIONVOLUME and MIGRATIONCONTROLDATASET.....	1374
141. Sample Terminal List of a Volume from the MCDS when You Specify VOLUME, MIGRATIONCONTROLDATASET, and TERMINAL.....	1374
142. Sample Printer List of Tape Migration Level 2 Volumes when You Specify MIGRATIONLEVEL2(TAPE) and MIGRATIONCONTROLDATASET.....	1376

143. Sample Printer List from the BCDS when You Specify PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET.....	1377
144. Sample Terminal List from the BCDS when You Specify VOLUME, BACKUPCONTROLDATASET, and TERMINAL.....	1378
145. Sample Printer List from the BCDS for a Specific Primary Volume with Cloud Dump Classes when You Specify PRIMARYVOLUME(volser), BACKUPCONTROLDATASET.....	1378
146. Sample Printer List from the BCDS when You Specify ALLDUMPS with PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET.....	1379
147. Sample Printer List of the cloud dump information for a specific primary volume when you specify ALLDUMPS with PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET.....	1379
148. Sample Printer List of dump information for a specific primary volume when you specify ALLDUMPS with PRIMARYVOLUME(volser) and BCDS.....	1379
149. Sample Printer List from the BCDS when You Specify PRIMARYVOLUME(volser), BACKUPCONTENTS, and BACKUPCONTROLDATASET.....	1381
150. Sample Printer List when You Specify HOST(hostid) and RESET.....	1381
151. Sample Printer List when You Specify TAPETABLEOFCONTENTS SELECT(CONNECTED).....	1384
152. Sample Terminal List when You Specify TAPETABLEOFCONTENTS, NODATASETINFORMATION and TERMINAL.....	1384
153. Sample Printer List when You Specify TAPETABLEOFCONTENTS with DATASETINFORMATION...	1385
154. Sample Terminal List when You Specify TAPETABLEOFCONTENTS, DATASETINFORMATION and TERMINAL.....	1385
155. Sample Printer List when You Specify TAPETABLEOFCONTENTS, SELECT, NOALTERNATEVOLUMES, and LIBRARY.....	1385
156. Sample Printer List when You Specify TAPETABLEOFCONTENTS, SELECT, and DISASTERALTERNATEVOLUMES.....	1385
157. Sample Printer List when You Specify TAPETABLEOFCONTENTS, SELECT, ML2 and FAILEDRECYCLE.....	1386
158. Sample Printer List when You Specify TAPETABLEOFCONTENTS and FAILEDCREATE.....	1386
159. Sample Printer List when You Specify EXCESSIVEVOLUMES.....	1386
160. Sample Printer List when You Specify SELECT(NOTASSOCIATED).....	1386
161. Sample Printer List of User Entries when You Specify USER.....	1387
162. Sample report of daily statistics by function.....	1405

163. Sample report of volume statistics by function.....	1410
--	------

Tables

1. Syntax examples.....	lv
2. Standard Interval Migration Processing Compared to TMM Processing.....	31
3. Storage group Auto Migrate attribute values.....	33
4. Data set characteristics for automatic space management example.....	57
5. Rules for migration versus class transition, based on eligibility and return code from migration exit.....	72
6. Last Successful Class Transition Date (LSCTD) setting rules during class transitions and recalls.....	84
7. Recall Destination Summary.....	84
8. Commands for Deleting Migrated Data Sets.....	89
9. Conditions that can be encountered when obtaining a list of management class definitions during migration of an SMS-managed data set.....	91
10. Data Set Characteristics for Automatic Space Management Example (Non-SMS-Managed Volumes).....	109
11. Current Backup Version of Data Sets.....	114
12. Eligibility Criteria for Migration.....	128
13. Volume Contention in a Single DFSMSHsm-Host Environment.....	138
14. Volume Contention in a Multiple DFSMSHsm-Host Environment.....	140
15. SDSP Contention in Single and Multiple DFSMSHsm-Host Environments.....	145
16. RACF Authority Required for Space Management Functions.....	150
17. Restarting primary space management, secondary space management, or event-driven migration.	161
18. Example of Selecting Tasks and Assigning Volumes.....	170
19. Generations of Example Dumps.....	175
20. The state of the data set backup versions just before the fourth data set backup version is created on October 31st.....	184
21. The state of data set backup versions after roll-off processing completes. The second and the third versions rolled off.....	184

22. The state of data set backup versions just before the fourth data set backup version is created on October 31st.....	185
23. The state of data sets after roll-off processing is complete. The oldest version rolled-off and became a retained backup copy.....	185
24. Retained backup copies after roll-off processing completes.....	185
25. Data Set Characteristics for Availability Management Example.....	220
26. Determining Data Set Sizes and Their Task Targets.....	227
27. Allowable combinations of fast replication options.....	256
28. Wildcard support on the FRRECOV command.....	272
29. QUERY command parameters used to obtain information about fast replication functions.....	285
30. LIST command parameters used to obtain information about fast replication functions.....	286
31. REPORT command parameters used to obtain information about fast replication functions.....	288
32. Backup Version Data Set Selection for Recovery.....	301
33. Backup Version Data Set Selection for Recovery.....	317
34. Data Sets at the Example Aggregate Backup Site.....	325
35. Aggregate Backup Checklist.....	327
36. Acceptable Generic Job Names.....	358
37. DD Statements Used by ABA.....	359
38. The sequence of files associated with ABACKUP Version 1 on the tape volser A00040.....	361
39. The sequence of files associated with ABACKUP Version 2 on the tape volser A00041.....	361
40. File sequence after ABACKUP 2 files are copied onto the tape containing ABACKUP 1 files. Tape's volser is A00040.....	362
41. Aggregate Recovery Data Set Processing.....	373
42. Aggregate Recovery Checklist.....	375
43. Expected Results for Level 0 Data Sets in the INCLUDE List (Processed by DFDSS).....	388
44. Expected Results for Migrated Data Sets in the INCLUDE List.....	389
45. Volume Contention in Single and Multiple DFSMSHsm Host Environments.....	445

46. Resource Access Authority Required for DFSMSHsm Functions.....	460
47. Recovering RACF-Protected Data Sets Backed Up under DFHSM before Version 2 Release 2.0.....	463
48. CVQ Scope.....	474
49. ARCBJRNL Return Codes.....	512
50. ARCIMPRT Return Codes.....	512
51. Cartridge Percentages, by Output Device and Length.....	542
52. Report Types Supplied by DFSMSHsm.....	579
53. RACF Authorization Required for DFSMSHsm Functions.....	620
54. SETSYS Parameters Not Applicable to SMS-Managed Data Sets.....	679
55. DFSMSHsm Commands Not Used for SMS Storage Management.....	679
56. Command sequence.....	687
57. Headings of Output When You Specify EXPIREBV and DISPLAY.....	861
58. Availability management functions and SETSYS parameters.....	1076
59. Space management functions and SETSYS parameters.....	1078
60. Supporting functions and SETSYS parameters.....	1080
61. SMS relationship abbreviations and their meaning.....	1082
62. SETSYS parameter relationship to SMS-managed volumes and data sets.....	1083
63. Summary of DFSMSHsm Default Values.....	1084
64. Determining data set task targets.....	1131
65. Summary of DFSMSHsm default values.....	1157
66. Tape Utilization Capacity for MEDIA3 or Greater.	1200
67. Compatibility between original volume unit names and alternate volume unit names.....	1230
68. DFSMSHsm Processing by Condition.....	1232
69. Compatibility between original volume unit names and alternate volume unit names.....	1237
70. Comparison of AUDIT Options.....	1255

71. AUDIT Commands that Audit and Report Errors.....	1256
72. Summary of Errors Detected by the Listed AUDIT Options.....	1276
73. Original AUDIT Commands that Audit and Report Errors.....	1281
74. Headings of Output When You Audit Data Set Information.....	1281
75. Summary of errors detected or corrected by the listed AUDIT options.....	1294
76. Error codes used in AUDIT reports.....	1295
77. Output labels when you request information for aggregate backup and recovery.....	1341
78. Output labels when you request information for backup volumes.....	1345
79. Output labels for List of Cloud Storage Contents.....	1346
80. Output labels when you request information for copy pools.....	1348
81. Output labels when you request MCDS information for data sets.....	1352
82. Output labels when you request a summary of the data sets in the MCDS.....	1354
83. Output labels when you request data set information from the BCDS.....	1355
84. Output labels when you request data set information for the latest backup version.....	1360
85. Output labels when you request dump class information.....	1361
86. Output labels when you request information for dump volumes.....	1364
87. Output labels when you request dump content information for dump volumes.....	1367
88. Output labels when you request information from the MCDS for primary and DASD migration volumes.....	1371
89. Output labels when you request information from the MCDS for tape migration level 2 volumes....	1375
90. Output labels when you request information from the BCDS for primary and migration volumes....	1377
91. Output labels when you request dump information from the BCDS for primary volumes.....	1378
92. Output labels when you request contents of backup VTOC information from the BCDS for primary volumes.....	1379
93. Output labels when you request information about records serialized by DFSMSHsm host ID.....	1381
94. Output labels when you request information from the tape table of contents.....	1381
95. Headings of output when you request a daily statistics report.....	1401

96. Headings of output when you request a volume statistics report.....	1406
---	------

About this document

This document describes the tasks that must be performed to use IBM z/OS DFSMSHsm, a component of DFSMS that manages your storage in a hierarchical manner. It contains examples of the commands needed to perform the tasks, descriptions of the processing that DFSMSHsm performs, and examples of the results of that processing.

For information about the accessibility features of z/OS, for users who have a physical disability, see [Appendix A, “Accessibility,” on page 1413](#).

Who should read this document

This document is intended for use by anyone responsible for managing storage for a computer center. It is designed to be tutorial in nature. Information presented in one topic is often based on knowledge obtained in a preceding topic.

Major divisions of this document

This document is divided into two parts, which are divided into topics. The divisions of this document are:

- **[Part 1, “DFSMSHsm Storage Administration Guide,” on page 1](#)**
 - The following topic provides an overview of DFSMSHsm.
 - [Chapter 1, “Welcome to DFSMSHsm,” on page 7](#), which describes the functions of DFSMSHsm and explains how to use this document.
 - The following topics introduce the tasks necessary to control space management functions:
 - [Chapter 2, “Space management of SMS-managed storage,” on page 23](#), which describes the tasks to be performed and the results of processing for space management of data sets managed by the storage management subsystem (SMS).
 - [Chapter 3, “Space management of non-SMS-managed storage,” on page 99](#), which describes the tasks to be performed and the results of processing for space management of data sets not managed by SMS.
 - [Chapter 4, “Other space management considerations,” on page 127](#), which describes aspects of space management that deserve consideration but are peripheral to the discussions in the two preceding topics.
 - [Chapter 5, “Space management procedures,” on page 161](#), which describes procedures for starting and stopping space management under varying conditions.
 - The following topics introduce the tasks necessary to control availability management functions:
 - [Chapter 6, “Availability management of SMS-managed storage,” on page 165](#), which describes the tasks to be performed and the results of processing for availability management of data sets managed by SMS.
 - [Chapter 7, “Availability management of non-SMS-managed storage,” on page 303](#), which describes the tasks to be performed and the results of processing for availability management of data sets not managed by SMS.
 - [Chapter 8, “Aggregate backup and recovery support \(ABARS\),” on page 319](#), which describes the command-driven functions that back up and recover a user-defined group (aggregate group) of data sets.
 - [Chapter 9, “Making disaster backup copies of DFSMSHsm-owned tape volumes,” on page 435](#), which describes how to make backup copies of DFSMSHsm-owned, cartridge-type, single-file-format tape volumes and how to recover those volumes.

- [Chapter 10, “Other availability management considerations,” on page 445](#), which describes aspects of availability management that deserve consideration but are peripheral to the discussions in the preceding topics.
- [Chapter 11, “Availability management procedures,” on page 485](#), which describes procedures for starting and stopping availability management under varying conditions.
- The following topics introduce the activities necessary to maintain DFSMSHsm-owned data sets and volumes.
 - [Chapter 12, “Maintaining DFSMSHsm control data sets,” on page 493](#), which describes the activities necessary to ensure that you can recover from damaged or lost DFSMSHsm control data sets.
 - [Chapter 13, “Managing DFSMSHsm media,” on page 527](#), which describes the activities necessary to manage data stored on DFSMSHsm-owned volumes.
 - [Chapter 14, “Obtaining information from DFSMSHsm,” on page 559](#), which describes the information that you can obtain from DFSMSHsm about its operations and the data sets and volumes that it is managing.
 - [Chapter 15, “DFSMSHsm tape libraries,” on page 589](#), which briefly explains DFSMSHsm automated and manual tape libraries.
 - [Chapter 16, “Using DFSMSHsm with DFSMSrmm,” on page 591](#), which explains the interaction between DFSMSHsm and DFSMSrmm when managing DFSMSHsm tape volumes.
 - [Chapter 17, “Operating procedures for DFSMSHsm,” on page 593](#), which describes procedures to be performed by the computing system operator.
 - [Chapter 18, “RACF considerations,” on page 617](#), which discusses pertinent RACF® information. RACF is a component of the Security Server for z/OS. This information includes the use of discrete and generic profiles with DFSMSHsm, and the use of FACILITY class profiles with regard to command control access in DFSMSHsm.
 - [Chapter 19, “Health Checker for DFSMSHsm,” on page 627](#), which describes information about Health Checker for DFSMSHsm.
- The following topic introduces data recovery scenarios.
 - [Chapter 20, “Data recovery scenarios,” on page 629](#), which describes real-life examples of data loss and recovery situations
- The following topics provide additional DFSMSHsm information:
 - [Chapter 21, “Consolidated commands, storage group definitions, and management classes for the example system,” on page 653](#)
 - [Chapter 22, “Preventive methods for avoiding data loss,” on page 661](#)
 - [Chapter 23, “Linux-z/OS DFSMSHsm dump or restore how-to,” on page 665](#)
- **[Part 2, “DFSMSHsm Storage Administration Reference,” on page 677](#)**
 - [Chapter 24, “DFSMSHsm commands,” on page 681](#) provides an overview of the DFSMSHsm commands.
 - The following topics provide detailed explanations of the operator, storage administrator, and system programmer commands, listed in alphabetical order, as follows:
 - [Chapter 25, “ABACKUP command: Backing up aggregated data sets,” on page 691](#)
 - [Chapter 26, “ADDVOL command: Adding or changing DFSMSHsm volumes,” on page 697.](#)
 - [Chapter 27, “ALTERDS command: Changing the back-ups for a non-SMS-managed data set,” on page 715.](#)
 - [Chapter 28, “ALTERPRI command: Altering the priority of queued requests,” on page 719.](#)
 - [Chapter 29, “ARECOVER command: Recovering a specified aggregate group,” on page 723.](#)
 - [Chapter 30, “AUDIT command: Auditing DFSMSHsm,” on page 737.](#)
 - [Chapter 31, “AUTH command: Identifying authorized users,” on page 755.](#)
 - [Chapter 32, “BACKDS command: Backing up a specific data set,” on page 759.](#)

- [Chapter 36, “BACKVOL command: Backing up data sets from volumes or dumping volumes,” on page 779.](#)
- [Chapter 37, “BDELETE command: Deleting backup versions of a data set,” on page 797.](#)
- [Chapter 38, “CANCEL command: Canceling a queued or active DFSMSHsm request,” on page 801.](#)
- [Chapter 40, “DEFINE command: Defining control structures for use by DFSMSHsm,” on page 809.](#)
- [Chapter 41, “DELETE Command: Deleting a migrated data set from a migration volume,” on page 839.](#)
- [Chapter 42, “DELVOL command: Removing a volume from DFSMSHsm management,” on page 841.](#)
- [Chapter 43, “DISPLAY command: Displaying DFSMSHsm storage locations,” on page 849.](#)
- [Chapter 44, “EXPIREBV command: Deleting unwanted backup versions of data sets and UNIX files,” on page 851.](#)
- [Chapter 45, “FIXCDS command: Displaying or modifying MCDS, BCDS, and OCDS records,” on page 863.](#)
- [Chapter 46, “FRBACKUP command: Requesting a fast replication backup or dump version,” on page 865.](#)
- [Chapter 47, “FRDELETE command: Deleting fast replication backup or dump versions,” on page 873.](#)
- [Chapter 48, “FREEVOL command: Freeing DFSMSHsm-owned volumes,” on page 877.](#)
- [Chapter 49, “FRRECOV command: Requesting a fast replication recovery,” on page 883.](#)
- [Chapter 50, “HOLD command: Preventing DFSMSHsm functions from running,” on page 895.](#)
- [Chapter 51, “HSEND CMD command: Issuing DFSMSHsm authorized-user commands from a TSO terminal,” on page 913.](#)
- [Chapter 53, “LIST command: Listing information from the MCDS, BCDS, and OCDS,” on page 919.](#)
- [Chapter 54, “LOG command: Entering data into the DFSMSHsm log,” on page 953.](#)
- [Chapter 55, “MIGRATE command: Requesting a space management function,” on page 955.](#)
- [Chapter 56, “ONLYIF command: Running commands in ARCCMDxx conditionally,” on page 973.](#)
- [Chapter 57, “PATCH command: Modifying storage within DFSMSHsm,” on page 979.](#)
- [Chapter 58, “QUERY command: Displaying the status of parameters, statistics, and requests,” on page 981.](#)
- [Chapter 59, “RECALL command: Recalling a specific migrated data set,” on page 991.](#)
- [Chapter 60, “RECOVER command: Recovering a backup version or dump copy of a data set, or UNIX file, or a volume,” on page 997.](#)
- [Chapter 61, “RECYCLE command: Consolidating valid data to one tape from other tapes,” on page 1015.](#)
- [Chapter 62, “RELEASE command: Releasing all or some DFSMSHsm functions for processing,” on page 1029.](#)
- [Chapter 63, “REPORT Command: Requesting reports based on DFSMSHsm statistics records,” on page 1043.](#)
- [Chapter 64, “SETMIG command: Changing the space management status of non-SMS-managed data sets or volumes,” on page 1053.](#)
- [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.](#)
- [Chapter 66, “STOP command: Shutting down DFSMSHsm,” on page 1223.](#)
- [Chapter 67, “SWAPLOG command: Switching between the DFSMSHsm log or PDA data sets,” on page 1225.](#)
- [Chapter 68, “TAPECOPY command: Making copies of selected cartridge-type tape volumes,” on page 1227.](#)

- [Chapter 69, “TAPERREPL command: Replacing cartridge-type tape volumes with their alternate volumes,” on page 1235.](#)
- [Chapter 70, “TRAP command: Requesting a dump when a specified error occurs,” on page 1241.](#)
- [Chapter 71, “UPDATEC command: Recovering the control data sets,” on page 1243.](#)
- [Chapter 75, “Using the AUDIT command,” on page 1255](#) contains diagnosis, modification, and tuning information.
- You use the LIST command to determine the status of DFSMSHsm and SMS-managed volumes and data sets. [Chapter 76, “Using the LIST command,” on page 1341](#) describes the categories of information you can list, explains the output of the LIST command, and contains sample lists.
- You use the QUERY command, described in [Chapter 77, “Using the QUERY command,” on page 1389](#), to determine what values you specified with the DFSMSHsm commands, to list DFSMSHsm statistics, or to list the status of pending requests.
- You use the REPORT command, described in [Chapter 78, “Using the REPORT command,” on page 1401](#), to generate reports based on the DFSMSHsm statistics records in the MCDS.

Required product knowledge

You should have a systems background either as a system programmer or storage administrator. You should also be familiar with the basic concepts of DFSMS described in [z/OS DFSMS Introduction](#). You are presumed to have a background in using TSO and an understanding of z/OS concepts and terms.

z/OS information

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

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

How to read syntax diagrams

This section describes how to read syntax diagrams. It defines syntax diagram symbols, items that may be contained within the diagrams (keywords, variables, delimiters, operators, fragment references, operands) and provides syntax examples that contain these items.

Syntax diagrams pictorially display the order and parts (options and arguments) that comprise a command statement. They are read from left to right and from top to bottom, following the main path of the horizontal line.

For users accessing the Information Center using a screen reader, syntax diagrams are provided in dotted decimal format.

Symbols

The following symbols may be displayed in syntax diagrams:

Symbol	Definition
--------	------------

- | | |
|-----|--|
| ▶ — | Indicates the beginning of the syntax diagram. |
| —▶ | Indicates that the syntax diagram is continued to the next line. |
| ▶ — | Indicates that the syntax is continued from the previous line. |
| —▶ | Indicates the end of the syntax diagram. |

Syntax items

Syntax diagrams contain many different items. Syntax items include:

- Keywords - a command name or any other literal information.
- Variables - variables are italicized, appear in lowercase, and represent the name of values you can supply.
- Delimiters - delimiters indicate the start or end of keywords, variables, or operators. For example, a left parenthesis is a delimiter.
- Operators - operators include add (+), subtract (-), multiply (*), divide (/), equal (=), and other mathematical operations that may need to be performed.
- Fragment references - a part of a syntax diagram, separated from the diagram to show greater detail.
- Separators - a separator separates keywords, variables or operators. For example, a comma (,) is a separator.

Note: If a syntax diagram shows a character that is not alphanumeric (for example, parentheses, periods, commas, equal signs, a blank space), enter the character as part of the syntax.

Keywords, variables, and operators may be displayed as required, optional, or default. Fragments, separators, and delimiters may be displayed as required or optional.

Item type

Definition

Required

Required items are displayed on the main path of the horizontal line.

Optional

Optional items are displayed below the main path of the horizontal line.

Default

Default items are displayed above the main path of the horizontal line.

Syntax examples

The following table provides syntax examples.

Table 1. Syntax examples	
Item	Syntax example
Required item. Required items appear on the main path of the horizontal line. You must specify these items.	►► KEYWORD — required_item ►◄
Required choice. A required choice (two or more items) appears in a vertical stack on the main path of the horizontal line. You must choose one of the items in the stack.	►► KEYWORD — <div>required_choice1 required_choice2</div> ►◄
Optional item. Optional items appear below the main path of the horizontal line.	►► KEYWORD — <div>optional_item</div> ►◄
Optional choice. An optional choice (two or more items) appears in a vertical stack below the main path of the horizontal line. You may choose one of the items in the stack.	►► KEYWORD — <div>optional_choice1 optional_choice2</div> ►◄

Table 1. Syntax examples (continued)

Item	Syntax example
<p>Default.</p> <p>Default items appear above the main path of the horizontal line. The remaining items (required or optional) appear on (required) or below (optional) the main path of the horizontal line. The following example displays a default with optional items.</p>	
<p>Variable.</p> <p>Variables appear in lowercase italics. They represent names or values.</p>	
<p>Repeatable item.</p> <p>An arrow returning to the left above the main path of the horizontal line indicates an item that can be repeated.</p> <p>A character within the arrow means you must separate repeated items with that character.</p> <p>An arrow returning to the left above a group of repeatable items indicates that one of the items can be selected, or a single item can be repeated.</p>	
<p>Fragment.</p> <p>The fragment symbol indicates that a labelled group is described below the main syntax diagram. Syntax is occasionally broken into fragments if the inclusion of the fragment would overly complicate the main syntax diagram.</p>	

How to provide feedback to IBM

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

Summary of changes

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

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

Summary of changes for z/OS 3.2

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

New

The following content is new.

September 2025 release

Added the following details as per [OA66198: NEW FUNCTION - DFSMSHSM S3 SUPPORT FOR TCT AND DIRECT TO CLOUD](http://www.ibm.com/support/pages/apar/OA66198) (www.ibm.com/support/pages/apar/OA66198) (APAR OA66198, which applies from 3.2 and above)

- SMSPROVIDER. For more information, see [“Syntax of the SETSYS command”](#) on page 1057.
- CDAPROVIDER and SMSPROVIDER. For more information, see [“CLOUD: Specifying CLOUD properties”](#) on page 1115.
- [“ZCOMPRESS\(CLOUDMIGRATE\(NO | YES\)\)”](#): Specifying whether to use CDA compression for data sets during migration to DIRECT cloud connections” on page 1208.
- Column: CPCTL, Error Number: 205, 206, 209, 215, 218, 219, and 231. For more information, see [“Summary of errors detected by the enhanced AUDIT commands”](#) on page 1276.
- Error 231. For more information, see [“Error codes \(*ERR\) and diagnosis”](#) on page 1295.
- NETWORK parameter. For more information, see [Chapter 77, “Using the QUERY command,”](#) on page 1389.

Changed

The following content is changed.

September 2025 release

Updated the following details as per [OA66198: NEW FUNCTION - DFSMSHSM S3 SUPPORT FOR TCT AND DIRECT TO CLOUD](http://www.ibm.com/support/pages/apar/OA66198) (www.ibm.com/support/pages/apar/OA66198) (APAR OA66198, which applies from 3.2 and above)

- [“Compaction or zEDC compression during migration”](#) on page 75.
- [“Reconnecting data sets using fast subsequent migration”](#) on page 94.
- [“Object storage in the cloud”](#) on page 156.
- [“System set up to use the cloud”](#) on page 156.
- [“DASD volumes”](#) on page 156.
- [“Cloud password”](#) on page 156.
- Title only. For more information, see [“Workload distribution with TCT”](#) on page 157.
- Fast subsequent migration reconnect is attempted requirements. For more information, see [“Reconnecting data sets using fast subsequent migration”](#) on page 94.
- [“DUMPCCLASS\(TAPE | CLOUD\)”](#): Specifying where the dump copies to reside” on page 835.

- [“DUMPCCLASS\(ZCOMPRESS\): Specifying whether to use zEDC Services on the dump data”](#) on page 829.
- [“NETWORK: Displaying cloud endpoint statistics known to DFSMSHsm”](#) on page 987.
- O: ZCOMPRESS optional parameters. For more information, see [“Syntax of the SETSYS command”](#) on page 1057.
- Name(cloud_network_connection_name) description. For more information, see [“CLOUD: Specifying CLOUD properties”](#) on page 1115.
- Note. For more information, see [“DSSXMMODE: Specifying whether DFSMSdss will be loaded in its own address space using the cross memory interface, or in the DFSMSHsm address space, for backup, CDS backup, dump, migration, and full-volume and data set recovery”](#) on page 1133.
- Note. For more information, see [“DUMPIO: Specifying the DFSMSdss DASD I/O buffering technique”](#) on page 1134.
- Optional sub parameters. For more information, see [“ZCOMPRESS: Specifying when compression with zEDC should be done”](#) on page 1207.
- If there is a data set record in the MCDS section. For more information, see [“Errors detected when using AUDIT MEDIACONTROLS for cloud storage”](#) on page 1267.
- Error 215 description. For more information, see [“Error codes \(*ERR\) and diagnosis”](#) on page 1295.
- SETSYS parameter. For more information, see [Chapter 77, “Using the QUERY command,”](#) on page 1389.
- Updated REFRESH as per [OA68113: MSGADR617E GDKWRITE RC800, HWTHRQST RC106 DIAGREA: FOPEN\(//DD:SYS00001\) DURING MIGRATION TO CDA DIRECT CLOUD](#) (www.ibm.com/support/pages/apar/OA68113) (APAR OA68113, which applies from 3.1 and above). For more information, see [“CLOUD: Specifying CLOUD properties”](#) on page 1115

Deleted

The following content is deleted.

September 2025 release

- None.

Summary of changes for z/OS 3.1

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

New

The following content is new.

June 2025 release

- Added REPLACE parameter. For more information, see [“REPLACE: Specifying that the recovered backup version or dump copy is to replace any existing data set of the same name”](#) on page 1009.

April 2025 release

- Added Notes 3 and 4. For more information, see [“PLEXNAME: Specifying a name for an HSMplex ”](#) on page 1170.
- The unexpected error MSG 924, MSG 925, and MSG 926 is added. For more information, see [“MSG 924”](#) on page 1339, [“MSG 925”](#) on page 1339, and [“MSG 926”](#) on page 1339.

March 2025 release

- Added Note. For more information see, [“Dump processing”](#) on page 193

February 2025 release

- Added Note. For more information, see [“MIGRATIONCLEANUPCYCLE and SECONDARYSPMGMTCYCLE: Specifying the cycle for automatic secondary space management functions”](#) on page 830.
- Added TAPELIBRARYPRESELECT parameter. For more information, see [“TAPELIBRARYPRESELECT: Preselecting output tape volumes in IBM Tape Library tape environment”](#) on page 1191.
- Added TAPELIBRARYPRESELECT parameter under SETSYS. For more information, see [Chapter 77, “Using the QUERY command,”](#) on page 1389.
- Added a new optional parameter, TAPELIBRARYPRESELECT. For more information, see [“Syntax of the SETSYS command”](#) on page 1057.
- Added TAPELIBRARYPRESELECT under Tape Environment. For more information, see [“SETSYS fast-path for DFSMSHsm functions”](#) on page 1075.

June 2024 release

- Added a new optional parameter, ISSUEARC1902. For more information, see [“ISSUEARC1902: Enable/disable the issue of ARC1902I message to the console”](#) on page 1146, [“Summary of DFSMSHsm default values”](#) on page 1084 and [“Syntax of the SETSYS command”](#) on page 1057.

February 2024 release

- Added the following restrictions to NEWNAME option with BACKDS command and added a link to the restrictions. For more information, see [“Data set backup using NEWNAME”](#) on page 454 and [“NEWNAME: Specifying a name to assign to the backup version”](#) on page 767.

September 2023 release

- None.

Changed

The following content is changed.

June 2025 release

- Updated Explanation and Note. For more information see, [“TAPETYPE: Specifying whether to use WORM media for non-ABARS functions”](#) on page 1198
- Updated Explanation. For more information see, [“BACKUPSTORAGEGROUP: Overriding the copy pool backup storage group with an alternate storage group”](#) on page 866
- Removed obsolete text. For more information see, [Chapter 23, “Linux-z/OS DFSMSHsm dump or restore how-to,”](#) on page 665

April 2025 release

- Updated MEDIACONTROLS(CLOUD(*cloud_network_connection_name*)). For more information see, [“MEDIACONTROLS VOLUMES, MEDIACONTROLS\(SDSP\) VOLUMES and MEDIACONTROLS\(CLOUD\): Requesting an audit of control information”](#) on page 748.

March 2025 release

- Updated Note 4. For more information see, [“DUMPCYCLE: Specifying the automatic dump cycle”](#) on page 829
- Updated Note 2. For more information see, [“Specifying the day on which a dump occurs”](#) on page 170

February 2025 release

- Updated Explanation section. For more information, see [“FORCE: Replacing a DASD backup copy that has an incomplete dump copy”](#) on page 869
- Updated ARC0747I to ARC0747E. For more information, see:
 - [“Considerations when using multicluster control data sets”](#) on page 502

- [“Changing backup device from DASD to tape” on page 504](#)
- [“Changing the names of the backup data sets” on page 503](#)
- [“WAIT and NOWAIT: Specifying whether to wait for command completion” on page 914](#)
- Updated [“Maintenance and use of the data set date-last-referenced field for space management” on page 78](#)
- Updated ERR 32 userdsn—EXTRANEIOUS MIGRATED DATA SET RECORD EXTENSION details in [“Error codes \(*ERR\) and diagnosis” on page 1295](#)

December 2024 release

- Recovering from a large-scale cyber event is updated. For more information, see:
 - [“Case 12: Recovering from a large-scale cyber event” on page 650](#)

October 2024 release

- Under the section SETSYS, ARC0151I is updated. For more information, see:
 - [Chapter 77, “Using the QUERY command,” on page 1389](#)
 (APAR OA66239, which also applies to z/OS 2.4 and 2.5.)

June 2024 release

- The AUDIT command is updated. For more information, see:
 - [“Diagnosis and repair action for AUDIT MEDIACONTROLS for CLOUD” on page 1268](#)
 - [“Summary of errors detected by the enhanced AUDIT commands” on page 1276](#)
 - [“Errors detected when using AUDIT MEDIACONTROLS for cloud storage” on page 1267](#)
 - [“Error codes \(*ERR\) and diagnosis” on page 1295](#)
 (APAR OA61709, which also applies to z/OS 2.5.)

January 2024 release

- The HBACKUP command is updated. For more information, see:
 - [“Syntax of the Java hbackup command” on page 1247](#)
 - [“Optional parameters of the Java hbackup command” on page 1247](#)
 (APAR OA65319, which also applies to z/OS V2R5.)

September 2023 release

- The HSEND CMD command is updated. For more information, see:
 - [“Syntax of the HSEND CMD command” on page 913](#)
 - [“WAIT and NOWAIT: Specifying whether to wait for command completion” on page 914](#)
 - [“Examples of how to code the HSEND CMD command” on page 915](#)
 (APAR OA64893)

November 2023 release

- The AUDIT FILECONTROLS(BACKUP) command has been updated. For more information, see:
 - [“Error codes \(*ERR\) and diagnosis” on page 1295](#)
 - [“Using the AUDIT FILECONTROLS\(BACKUP\) command” on page 1266](#)
 (APAR OA63126, which also applies to z/OS V2R4 and V2R5.)
- Transparent Cloud Tiering Disaster Recovery Enhancement has been updated. For more information, see:
 - [“Error codes \(*ERR\) and diagnosis” on page 1295](#)
 - [“Using the AUDIT DATASETCONTROLS \(MIGRATION\) command” on page 1261](#)

- [“Configuring migration cleanup” on page 54](#)
- [“MIGRATIONCLEANUPDAYS: Specifying the number of days to keep MCDS data set records and statistics records” on page 1157](#)
- [“Syntax of the SETSYS command” on page 1057](#)
- [“Summary of DFSMSHsm default values” on page 1084](#)
- [Chapter 77, “Using the QUERY command,” on page 1389](#)

(APAR OA63678, which also applies to z/OS V2R4 and V2R5.)

- The INUSE keyword description was updated for Chapter 32. BACKDS command. For more information, see [“INUSE: Tailoring backup processing of UNIX files in use at the time of backup” on page 766](#). (APAR OA65396, which also applies to z/OS V2R4 and V2R5.)
- The -I option description was updated for Chapter 73. Java hbackup command. For more information, see [“Optional parameters of the Java hbackup command” on page 1247](#). (APAR OA65396, which also applies to z/OS V2R4 and V2R5.)

Part 1. DFSMSHsm Storage Administration Guide

This topic describes the tasks that must be performed to use DFSMSHsm, a component of DFSMS that manages your storage in a hierarchical manner. It contains examples of the commands needed to perform the tasks, descriptions of the processing that DFSMSHsm performs, and examples of the results of that processing.

Introduction

Chapter 1, “Welcome to DFSMSHsm,” on page 7 explains how to use this document and describes the functions of the DFSMSHsm program, highlighting the following information:

- The tasks that you perform to make DFSMSHsm do useful work.
- An overview of how DFSMSHsm moves data.
- How to use this document. You are introduced to the example system, which is referred to throughout the rest of this document.

Space management

The following topics introduce the tasks necessary to control space management functions. Space management is the DFSMSHsm program function that you use to ensure that your customers have space available on DASD volumes to allocate new data sets or to extend old ones. You can make the space available by deleting data sets that have outlived their usefulness, removing unused allocated space from data sets, moving low-activity data sets from level 0 volumes to other DASD or tape volumes, and returning the moved data sets to the level 0 devices when the data sets are needed.

- Chapter 2, “Space management of SMS-managed storage,” on page 23, describes the tasks to be performed and the results of processing for space management of data sets managed by the storage management subsystem (SMS).
- Chapter 3, “Space management of non-SMS-managed storage,” on page 99, describes the tasks to be performed and the results of processing for space management of data sets not managed by SMS.
- Chapter 4, “Other space management considerations,” on page 127, describes aspects of space management that deserve consideration but are peripheral to the discussions in the two preceding topics.
- Chapter 5, “Space management procedures,” on page 161, describes procedures for starting and stopping space management under varying conditions.

Functions of space management:

- Automatically delete, remove unused space from, and move data sets on DFSMSHsm-managed volumes to provide a specified amount of free space on each volume. The space management functions begin at a specified time of day and only on a specific day of a cycle. This operation is called *automatic primary space management*. The process of removing unused space from the data sets is called *space reduction*. Moving the data sets (as opposed to deleting or removing unused space from them) is called *migration*.
- Reconnect unchanged data sets that are recalled from ML2 tapes to their original ML2 tapes with the fast subsequent migration function.
- Ensure that a specified amount of space is available on DFSMSHsm-managed volumes. If DFSMSHsm finds any volume without the specified amount of space, it performs migration from all such volumes (known as *event-driven migration*) following a space check.
- Automatically clean up migration volumes and the migration control data set (MCDS). This function and the following one are known as *automatic secondary space management*.
- Automatically, or by command, migrate data sets from level 1 migration volumes to level 2 migration volumes.

- Automatically recall needed migrated data sets to level 0 volumes.
- Recall migrated data sets to level 0 volumes by command.
- Delete eligible data sets on a non-SMS-managed volume by command.
- Delete eligible non-SMS migrated data sets on a migration volume by command.
- Migrate individual data sets or eligible data sets on a volume by command.

Availability management

The following topics introduce the tasks necessary to control availability management functions. Availability management is the function of the DFSMSHsm program that you use to ensure that your customers can retrieve usable copies of their data sets should their online copies become lost or damaged. You can make the copies available by making daily incremental backup copies of changed data sets, making periodic dump copies of the DFSMSHsm-managed and ML1 volumes, making aggregate backup copies of data sets that your operation will need if your installation is damaged, and making fast replication backup versions for sets of storage groups.

- [Chapter 6, “Availability management of SMS-managed storage,” on page 165](#), which describes the tasks to be performed and the results of processing for availability management of data sets managed by SMS.
- [Chapter 7, “Availability management of non-SMS-managed storage,” on page 303](#), which describes the tasks to be performed and the results of processing for availability management of data sets not managed by SMS.
- [Chapter 8, “Aggregate backup and recovery support \(ABARS\),” on page 319](#), which describes the command-driven functions that back up and recover a user-defined group (aggregate group) of data sets.
- [Chapter 9, “Making disaster backup copies of DFSMSHsm-owned tape volumes,” on page 435](#), which describes how to make backup copies of DFSMSHsm-owned, cartridge-type, single-file-format tape volumes and how to recover those volumes.
- [Chapter 10, “Other availability management considerations,” on page 445](#), which describes aspects of availability management that deserve consideration but are peripheral to the discussions in the preceding topics.
- [Chapter 11, “Availability management procedures,” on page 485](#), which describes procedures for starting and stopping availability management under varying conditions.

Functions of availability management:

- Automatically make backup copies of individual changed data sets on DFSMSHsm-managed volumes. This is known as incremental backup. You can specify how frequently to back up data sets on a data set basis for SMS-managed data sets and on a system-wide basis for non-SMS-managed volumes. You can specify on a system-wide basis how often backup runs.
- Automatically make dump copies of DFSMSHsm-managed level 0 volumes and ML1 volumes on a specified schedule for day and time. You can dump different groups of volumes on different days with different periods for the number of days between dumps.
- Allow your customers to issue commands to recover their own data sets.
- Allow your customer to issue the HBACKDS command from a batch environment. See [“Inline backup” on page 447](#).
- By command, back up user data sets.
- By command, back up data sets of an application to tape so they can be taken to another computer site for recovery.
- By command, recover user data sets or data sets of an application to their original system environment at another computer site.
- By command, restore a volume from a dump copy and update the restored volume from later incremental backup versions.

- By command, recover a specific data set from either a dump copy or an incremental backup version.
- By command, create a fast replication backup of a copy pool.
- Automatically or by command, create a dump copy of a fast replication DASD backup version
- By command, recover volumes that have a fast replication backup version.
- By command, recover data sets from fast replication backup versions

Backup function

Backup is the process of copying a data set from a level 0 or an ML1 volume to daily backup volume. This copy is called a backup version. The purpose of backup is to have copies of data sets in case something happens to the original data sets. The difference between dump and backup is that the dump function backs up the entire allocated space on a volume, whereas the DFSMSHsm backup function backs up individual data sets.

DFSMSHsm can create backup versions of data sets either automatically or by command. DFSMSHsm automatically creates backup versions of data sets on specified days beginning at a specified time of day. The data sets must meet eligibility criteria and must be on DFSMSHsm-managed volumes that have been designated for automatic backup.

Fast replication function

Fast replication uses volume-level fast replication to create backup versions for sets of storage groups. You can define a set of storage groups through the SMS copy pool construct. The DFSMSHsm FRBACKUP command creates a fast replication backup version for each volume in every storage group defined within a copy pool. The fast replication backup versions can be dumped to tape. The dump copies can be created by using the FRBACKUP command or with Automatic Dump processing. You can use FRBACKUP to dump an existing fast replication backup version, or when no backup version exists, create a fast replication backup version and dump it to tape. In all cases, the dump copies are associated with the original source DASD volumes in the copy pool, but not with the target DASD volumes that are actually dumped. The dump time stamps reflect when the fast replication backup version is made, not the actual time of the dumps. Recovery from a fast replication backup version can be performed at the copy pool level from a disk copy, at the individual volume level from a disk or tape copy, or at the data set level from a disk or tape copy. This Fast Replication function enables the backup and recovery of a large set of volumes to occur within a small time frame.

The term fast replication refers to the FlashCopy® function supported by IBM System Storage disk and the SnapShot function supported by IBM RAMAC Virtual Array (RVA) disk.

Dump function

Dump is the process of copying all data from a DASD volume to dump tape volumes. Full-volume dump is an extension of DFSMSHsm's availability management that invokes DFSMSdss through the DFSMSdss application interface. Full-volume dump backs up the entire allocated space of DFSMSHsm-managed DASD volumes and ML1 volumes either automatically or by command. Non-DFSMSHsm-managed volumes are dumped only by command. The purpose of the full-volume dump is to expedite the recovery process when an entire volume is lost or damaged and to supplement the incremental recovery process.

The full-volume-dump process can be one volume in and one dump copy out. It can also be one volume in and multiple dump copies out. Each dump copy is a complete image of the dumped volume, not just an incremental backup of selected data sets. Each successive full-volume dump of a volume, regardless of the number of dump copies, is a generation. Each dump copy in a generation is associated with a different dump class, which specifies how the dump copy is to be managed. DFSMSHsm allows from one to five dump copies to be made concurrently for any one full-volume dump. DFSMSdss discontinues writing to dump copies on which errors occur. DFSMSdss creates the multiple copies and continues its full-volume-dump process as long as one output copy is good. DFSMSHsm discards the contents of only the bad copies and issues an appropriate message. If all copies fail, the full-volume dump is failed.

DFSMSHsm keeps 100 generations of dump copies for any given volume unless all the copies in a generation expire. When all dump copies in any generation for a volume reach their expiration dates, DFSMSHsm deletes that generation from its records. When the 100 generations for a particular volume have been reached and the next full-volume dump for that volume is performed, the control records for the oldest generation are discarded, regardless of the retention periods of the individual copies.

Types of backup volumes

You can define the following types of tape or DASD backup volumes to DFSMSHsm:

- **Daily backup volumes:** Daily backup volumes are assigned to a specific day in the backup cycle and contain the backup versions created on that day. DFSMSHsm creates these backup versions during the backup of a level 0 volume or during the backup of migrated data sets. Daily backup volumes can also contain backup versions created by the BACKDS command. However, the daily backup volume used to contain these manually backed up versions does not necessarily correspond to the day in the backup cycle when DFSMSHsm created the backup version. Instead, the daily backup volume corresponds to the day in the backup cycle that DFSMSHsm moved the backup version from its temporary location on an ML1 volume to a daily backup volume.
- **Fast replication target volumes:** Fast replication target volumes contain the fast replication backup copies of DFSMSHsm-managed volumes. Fast replication target volumes are defined with the SMS copy pool backup storage group type.
- **Spill backup volumes:** Spill backup volumes contain older backup versions of data sets. These backup versions are moved to spill backup volumes when DFSMSHsm needs more space on DASD daily backup volumes. Tape spill volumes also receive all valid backup versions of data sets when a tape backup volume is recycled.
- **Unassigned backup volumes:** DFSMSHsm uses unassigned volumes whenever it needs more daily or spill backup volumes. There are two types of unassigned volumes: unassigned daily backup volumes and unassigned backup volumes. When DFSMSHsm selects an unassigned volume during backup processing, DFSMSHsm assigns it as a daily or spill backup volume.
 - Unassigned daily backup volumes are those volumes defined to DFSMSHsm as daily backup volumes but not associated with any day in the backup cycle. DFSMSHsm can use these volumes only as daily backup volumes, but DFSMSHsm determines which day in the backup cycle the volume should be assigned to when it first uses the volume.
 - Unassigned backup volumes are those volumes defined to DFSMSHsm as backup volumes but not specified as daily or spill backup volumes. When it first uses unassigned backup volumes, DFSMSHsm assigns the volumes as daily or spill backup volumes. If DFSMSHsm uses a volume as a daily backup volume, DFSMSHsm determines which day in the backup cycle the volume should be assigned to.

When DFSMSHsm needs another daily backup volume, it chooses an unassigned daily backup volume before it chooses an unassigned backup volume.

Recovery and restore of data sets

Recovery and restore are processes that are requested only by command, not automatically, for backed up data sets. Recovery is the process of retrieving a full-volume dump or a backup version of a data set or a volume. Restore signifies that DFSMSdss is used to retrieve dumped data from dump volumes. You can use restore or recover processing to:

- Recover a data set that has been lost or damaged.
- Recover an earlier version of the data set without deleting the current version.
- Restore a volume from a full-volume dump and update the volume from later incremental backup versions.
- Restore a data set from a dump copy.
- Restore a volume from a full-volume dump.
- Recover a volume from DFSMSHsm backup versions.

- Recover a pool of volumes from the fast replication backup versions on DASD.
- Recover a single volume from the fast replication backup versions on DASD or tape.
- Recover a data set from the fast replication backup versions on DASD or tape.

DFSMSHsm volume recovery can use incremental backups or full-volume dumps or both. One DFSMSHsm RECOVER command can be used to request both a volume restore and an incremental volume recovery.

DFSMSHsm can recover data sets from a DFSMSHsm incremental backup version or from a DFSMSHsm dump copy. DFSMSHsm automatically chooses the more recent copy of the data set if you allow users to perform data set restores (see [“Controlling restoring of individual data sets from dump tapes”](#) on page 172).

There are special considerations for full volume restores of volumes that were dumped using the BACKVOL DUMP command and whose data sets contain RLS attributes.

DFSMSHsm support activities

The following topics introduce the activities necessary to maintain DFSMSHsm-owned data sets and volumes. In addition to the activities that cause the DFSMSHsm program to perform its space management and availability management functions, certain other activities maintain the security and efficiency of DFSMSHsm itself. For example, backup copies of the control data sets should be kept, and the DFSMSHsm-owned volumes must be maintained.

- Chapter 12, [“Maintaining DFSMSHsm control data sets,”](#) on page 493, describes the activities necessary to ensure that you can recover from damaged or lost DFSMSHsm control data sets.
- Chapter 13, [“Managing DFSMSHsm media,”](#) on page 527, describes the activities necessary to manage data stored on DFSMSHsm-owned volumes.
- Chapter 14, [“Obtaining information from DFSMSHsm,”](#) on page 559, describes the information that you can obtain from DFSMSHsm about its operations and the data sets and volumes that it is managing.
- Chapter 15, [“DFSMSHsm tape libraries,”](#) on page 589, explains DFSMSHsm automated and manual tape libraries.
- Chapter 16, [“Using DFSMSHsm with DFSMSrmm,”](#) on page 591, explains the interaction between DFSMSHsm and DFSMSrmm when managing DFSMSHsm tape volumes.
- Chapter 17, [“Operating procedures for DFSMSHsm,”](#) on page 593, describes procedures to be performed by the computing system operator.
- Chapter 18, [“RACF considerations,”](#) on page 617, discusses pertinent RACF information. RACF is a component of the Security Server for z/OS. This information includes the use of discrete and generic profiles with DFSMSHsm, and the use of FACILITY class profiles with regard to command control access in DFSMSHsm.
- Chapter 19, [“Health Checker for DFSMSHsm,”](#) on page 627, describes information about Health Checker for DFSMSHsm.

Data recovery scenarios

Chapter 20, [“Data recovery scenarios,”](#) on page 629, describes real-life examples of data loss and recovery situations. It describes a variety of situations where data is lost, and what steps to take to recover that data. It also includes situations where the lost data is a control data set or journal.

Chapter 1. Welcome to DFSMShsm

DFSMShsm is a functional component of the DFSMS family, which provides facilities for managing your storage devices. DFSMShsm is a DASD storage management and productivity tool for managing low-activity and inactive data. It relieves you from manual storage management tasks and improves DASD use by automatically managing both space and data availability in a storage hierarchy. DFSMShsm cooperates with the other products in the DFSMSdftp family to provide efficient and effective storage management.

DFSMShsm provides a storage management subsystem (SMS) that allows storage administrators to control the use of storage. The storage management subsystem provides storage groups, storage classes, management classes, and data classes that control the allocation parameters and management attributes of data sets. DFSMShsm performs space management and availability management of each data set as directed by the management class attributes of that data set. In addition, the storage group controls the allocation of the data set when DFSMShsm returns the data set to level 0 (L0) storage.

Purpose of space management

Space management is the function of DFSMShsm that allows you to keep DASD space available for users in order to meet the service-level objectives for your system. The purpose of space management is to manage your DASD storage efficiently. To do this, space management automatically and periodically performs functions that (1) move low-activity data sets from user-accessible volumes to DFSMShsm volumes, (2) move data sets from one class of user-accessible volumes to another class of user-accessible volumes, and (3) reduce the space occupied by data on both the user-accessible volumes and the DFSMShsm volumes. The DFSMShsm space management functions are:

- Automatic primary space management of DFSMShsm-managed volumes, which includes:
 - Deletion of temporary data sets
 - Deletion of expired data sets
 - Release of unused, over-allocated space
 - Migration to DFSMShsm-owned migration volumes
 - Transitions between different classes of user-accessible volumes
- Automatic secondary space management of DFSMShsm-owned volumes, which includes:
 - Migration level cleanup, including deletion of expired migrated data sets and some migration control data set (MCDS) records
 - Moving migration copies from migration level 1 (ML1) to migration level 2 (ML2) volumes
- Event-driven migration and transitions, initiated when a DFSMShsm-managed volume exceeds a specified threshold
- Automatic recall of user data sets back to DASD volumes
- Space management by command
- Space-saving functions, which include:
 - Small data set packing
 - Partitioned data set compression
 - Data compaction and zEDC compression
 - Data set reblocking

Purpose of availability management

Availability management ensures that a recent copy of your DASD data set exists. Availability management ensures that lost or damaged data sets can be retrieved at the most current possible level.

To do this, availability management automatically and periodically performs functions that (1) copy all the data sets on DASD volumes to tape volumes and that (2) copy the changed data sets on DASD volumes either to other DASD volumes or to tape volumes. By copying the data sets, DFSMSHsm minimizes the space that is occupied by the data set on the backup volume. Availability management comprises the following functions:

- Aggregate backup and recovery
- Automatic physical full-volume dump
- Automatic incremental backup
- Automatic control data set backup
- Command dump and backup
- Command recovery
- Disaster backup
- Expiration of backup versions
- Fast replication backup and recovery

What you are expected to do

Making DFSMSHsm perform useful work in your system requires four sets of tasks:

- Installing DFSMSHsm and its required data sets in the system
- Specifying to DFSMSHsm the automatic and command activities that it must perform
- Monitoring and tuning the operations of DFSMSHsm
- Partitioning the DFSMSHsm activities between systems (z/OS images) that share storage which must be managed. Establishing one or more DFSMSHsm address spaces (hosts) for each of those z/OS images

The first of these sets of tasks is described in the [*z/OS DFSMSHsm Implementation and Customization Guide*](#). DFSMSHsm provides a starter set of DFSMSHsm commands used at each DFSMSHsm startup in SYS1.SAMPLIB to help you get started using DFSMSHsm.

This guide describes the second, third, and fourth sets of tasks. That is, it describes how to use DFSMSHsm to manage your storage, how to monitor and tune DFSMSHsm operations, and how to partition DFSMSHsm activities between all of the DFSMSHsm hosts. Because this document is designed as a textbook, you can learn about DFSMSHsm tasks by reading this document from beginning to end. However, because this document is organized by tasks, once you understand which tasks you want to perform, you can go to specific topics to learn all of the necessary activities for that specific task.

Related reading

- For more information about the starter set, see [*z/OS DFSMSHsm Implementation and Customization Guide*](#).
- For more information about whether and how you need to partition DFSMSHsm activities, see [Chapter 17, “Operating procedures for DFSMSHsm,”](#) on page 593.

Tasks for managing storage with DFSMSHsm

The following set of tasks for using DFSMSHsm to manage storage has been defined:

- Setting overall DFSMSHsm system parameters. This task consists of defining those parameters that DFSMSHsm uses, no matter which of its functions it is to perform.
- Performing space management. This task consists of:
 - Defining the parameters that DFSMSHsm uses for automatic space management
 - Issuing commands that cause DFSMSHsm to perform specific space management functions

- Performing availability management. This task consists of:
 - Defining the parameters that DFSMSHsm uses for automatic availability management
 - Issuing commands that cause DFSMSHsm to perform specific availability management functions
- Maintaining the control data sets. This task consists of backing up and maintaining correct contents for the data sets that DFSMSHsm uses to keep track of the data sets that it is managing.
- Managing DFSMSHsm media. This task consists of eliminating not valid data sets from DFSMSHsm volumes and in using DFSMSHsm to move data sets when new kinds of volumes are introduced into your system.
- Determining what occurs in DFSMSHsm. This task consists of using commands that cause DFSMSHsm to issue reports of what has occurred under specific conditions and of reading and interpreting the reports.
- Running DFSMSHsm in a simulation mode. This task consists of using DFSMSHsm as though you were managing storage, but having DFSMSHsm only issue a report instead of actually moving data sets. The purpose of the task is to analyze the results that would be obtained by using DFSMSHsm commands in the way you planned to use them.
- Operating DFSMSHsm. This task consists of directing the operations that start, stop, and interrupt DFSMSHsm.

Overview of DFSMSHsm functions

Figure 1 on page 10 shows the generalized flow of data in DFSMSHsm. The user's data is stored on level 0 volumes, either SMS-managed or non-SMS-managed.

Your objective in using DFSMSHsm is to make the most efficient use of all your DASD storage, primarily by making better use of the level 0 volumes. Better use of the level 0 volumes results when:

- You specify thresholds for DFSMSHsm-managed volumes, and you can assure the users that space will be available for extending old data sets or allocating new data sets (space management).
- You can assure the users that a copy of their data sets will be available if the data sets should accidentally be lost from the level 0 volumes (availability management). Such assurance that you can provide backup copies may encourage users not to unnecessarily maintain their own backup copies on level 0 volumes.

DFSMSHsm allows you to perform both availability management and space management automatically, periodically, and by issuing specific commands when manual operations are necessary or desirable. The dump and backup copies are made to tape (for dump) or to DASD or tape (for backup).

Rule: If a data set is larger than the free space on the DASD backup or migration volume, then the migration or backup will fail.

DFSMSHsm can then restore the volumes and recover the data sets when they are needed to replace destroyed or damaged data sets.

Note:

1. The term "level 0" refers to all DASD volumes that contain user-accessible data. While there is a single level 0 tier within the overall storage hierarchy, it is generally comprised of many different classes of devices. For example, solid state drives (SSD) and SATA drives represent two unique classes. For simplicity, all classes will be generally referred to singularly as level 0.
2. Only level 0 DASD volumes can be SMS managed. Migration DASD and backup DASD should never be managed by SMS. Migration, backup, and dump tapes can be SMS-managed in tape libraries.
3. DFSMSHsm allows use of the cylinder-managed space on extended address volumes (EAVs) for migration copies and backup versions. Use the SETSYS USECYLINDERMANAGEDSPACE command to manage allocation of ML1, ML2, or backup data sets on cylinder managed space. See Chapter 65, "SETSYS command: Establishing or changing the values of DFSMSHsm control parameters," on page 1057 for more information.
4. Although DFSMSHsm performs the same functions of copying, moving, and deleting data sets for both SMS-managed and non-SMS-managed volumes, the kinds and levels of control are different.

Therefore, these DFSMSShsm functions are described separately for SMS-managed volumes and for non-SMS-managed volumes.

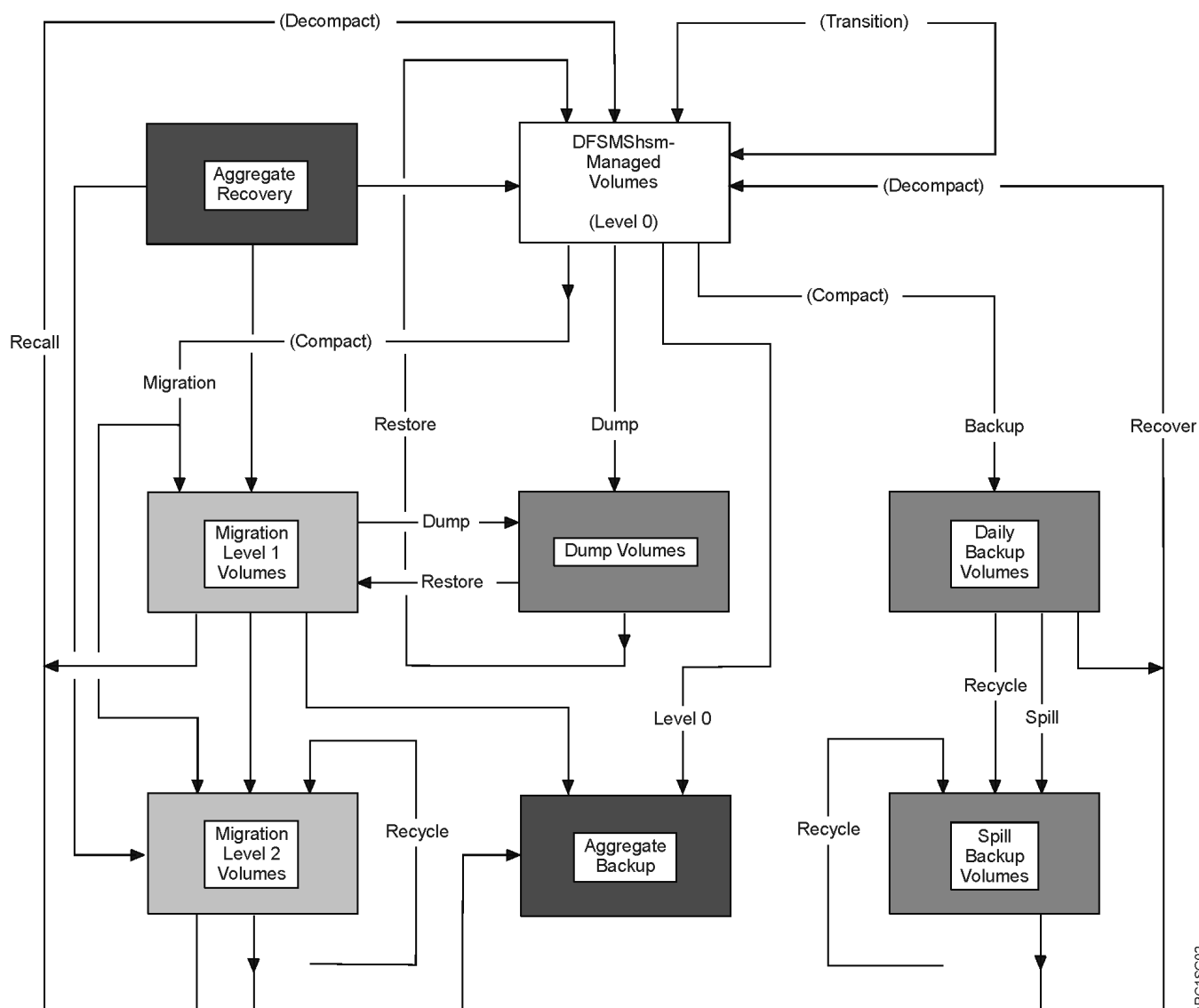


Figure 1. Data Set Flow in DFSMSShsm

DFSMSShsm functions for SMS-managed volumes

For SMS-managed level 0 volumes, the attributes specified in the storage groups and the management classes govern how DFSMSShsm performs automatic space management and incremental backup. You can override many of the attributes in the storage groups and management classes when the functions are performed by issuing specific commands. See [Figure 1 on page 10](#) when reading the following descriptions.

Note: If you have converted a non-SMS-managed L0 volume to an SMS-managed volume, you should:

- Add the volume to a storage group having the desired SMS attributes.
- DELVOL the volume to remove non-SMS attributes. Unexpected failures, for instance in autobackup processing, might occur if the volume is not DELVOLed.

Related reading

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

Use of storage groups and management classes

By managing storage with SMS, you establish the following:

- Storage classes to define levels of service provided to data sets
- Storage groups to govern the volumes to which data sets are allocated and subject to management by DFSMSHsm
- Storage groups to define which volumes are used as target volumes of the fast replication backup versions
- Management classes to define how DFSMSHsm manages the data sets
- Data classes to govern values for initially allocating the data sets
- Automatic class selection (ACS) routines that automatically select storage classes, management classes, data classes, and storage groups to govern the data sets

DFSMSHsm is concerned mainly with DASD storage groups and with management classes. DASD storage groups allow you to pool volumes for the purpose of defining how and whether:

- Automatic dumps are performed on the volumes and to what dump classes
- Automatic backups are performed on the volumes
- Automatic volume space management is performed on the volumes and at what level of occupancy

Management classes allow you to group data sets logically and specify how they are managed by space management and availability management.

As users allocate data sets, the automatic class selection routine associates them with the appropriate storage class, storage group, management class, and data class (data and management classes are optional). If a data set is not SMS managed, SMS does not associate any class or storage group with the data set.

The storage group and the management class are related only by the needs of particular data sets. That is, SMS can allocate data sets that are associated with different management classes to the volumes assigned to a single storage group. Thus, in using DFSMSHsm to manage SMS-managed storage, you must think in terms of managing data sets rather than in terms of managing volumes. (An exception to the preceding statement is the automatic dump function, which operates on a volume basis.)

Dump and restore functions

Because automatic dump is a volume function, the attributes that govern it are taken from the storage group. As [Figure 1 on page 10](#) shows, DFSMSHsm copies the data sets from level 0 volumes to dump volumes. Dump can be performed only on complete volumes.

Restore, the reverse of dump, moves dumped data sets from dump volumes to level 0 volumes. There are conditions under which a data set, in contrast to a volume, can be restored.

Related reading

- For more information about restoring a non-VSAM data set, see [“Restoring a non-VSAM data set from a dump copy” on page 247](#).
- For more information about restoring a VSAM data set, see [“Restoring a VSAM data set from a dump copy” on page 247](#).

Backup and recovery functions

The backup function is a data-set-level function when DFSMSHsm is processing SMS-managed volumes. That is, the management class attributes define how the data set is treated for creation and retention of backup versions. Thus, on the same volume you can have data sets with different:

- Permissions for automatic backup
- Conditions for command backup

- Minimum frequencies of backup
- Numbers of backup versions
- Retention periods for backup versions

For backup, DFSMSHsm makes copies (versions) of changed data sets residing on level 0 volumes onto backup volumes. DFSMSHsm uses the management class attributes and the guaranteed backup frequency attribute of the storage group for each data set to determine whether to copy the data set. After the data sets have been backed up, DFSMSHsm determines from the management class attributes for each data set how many backup versions to keep and how long to keep them.

Backup can occur to either DASD or tape, whichever you specify. If backup is performed to DASD, your DASD volumes can become filled with the current backup versions and earlier backup versions that have not been discarded. When DASD backup volumes become full, DFSMSHsm transfers the old (non-current but valid) backup versions to spill backup volumes. Transfer of valid versions from DASD backup volumes can be either to tape or to DASD.

If the backup or the spill processes are performed to tape, the tapes eventually contain many not valid versions that have been superseded by the more current versions that have been made. When tape volumes have many not valid versions, they are selected by the recycle process, which moves valid versions to spill backup volumes.

Recovery, the reverse of backup, returns data sets from the daily backup volumes or the spill backup volumes to the level 0 volumes. Recovery can be performed only by command, for individual data sets or for complete volumes.

Aggregate backup and recovery functions

The aggregate backup and recovery support (ABARS) allows you to back up and recover a user-defined group of data sets. The user-defined group of data sets can be those belonging to an application or any combination of data sets that you want treated as a separate entity. With the aggregate backup and aggregate recovery functions, you can:

- Define the components of an aggregate
- Back up data sets by aggregate
- Recover data sets by aggregate
- Duplicate your aggregates at a remote site
- Resume business at a remote location if necessary

When the NetView File Transfer Program is installed on any two nodes, it can be used to transfer copies of aggregate backup files by running a batch job or using NetView File Transfer Program input panels. The NetView File Transfer Program is referred to as NetView FTP in this book.

Related reading

For more information about ABARS, see [Chapter 8, “Aggregate backup and recovery support \(ABARS\),” on page 319](#).

Extended address volume (EAV) considerations

SMS will determine the volume selection logic for SMS managed data sets and EAVs. See [“Controlling allocations to extended address volumes \(EAVs\)” on page 129](#) and [Modifying the SYS1.PARMLIB IGDSMSxx member in *z/OS DFSMSdfp Storage Administration*](#).

Fast replication function

DFSMS provides a volume-level fast replication function that allows you to create a backup that is managed by DFSMSHsm with minimal application outage. The fast replication function supports the FlashCopy and SnapShot functions.

With a copy pool SMS construct, you can define a set of storage groups that DFSMSHsm processes collectively for fast replication functions. For recovery purposes, you can restore the fast replication backup versions for an entire volume or copy pool, or you can recover individual data sets from the fast replication backups.

Related reading

For more information, see [“Managing volume backups with fast replication” on page 254.](#)

Space management function

Space management consists of removing appropriate data sets from the level 0 volumes and reducing the space occupied on the level 0 volumes by the data sets that remain there. In removing the data sets, you can have DFSMSHsm either delete them entirely, migrate them to migration DASD or tape volumes, or transition them to other user-accessible DASD volumes. If you move the data sets to migration DASD or tape volumes (ML1 or ML2), they can still be automatically deleted from the migration volumes when they expire.

Rule: If a data set is larger than the free space on the DASD backup or migration volume, then the migration or backup will fail.

Note: DFSMSHsm allows use of the cylinder-managed space on extended address volumes (EAVs) for migration copies and backup versions. Use the SETSYS USECYLINDERMANAGEDSPACE command to manage allocation of ML1, ML2, or backup data sets on cylinder managed space. See [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#) for more information.

The space management functions of DFSMSHsm for SMS-managed volumes are performed on a data set level basis. That is, for each data set on a volume to be managed, DFSMSHsm uses the management class attributes of the management class with which the data set is associated to determine:

- Whether to release unused overallocated space
- Whether to delete the data set based on:
 - The amount of time since the data set was created
 - The amount of time since the data set was last used
 - An explicit date
- The minimum time to leave the data set unopened on the level 0 volume before migrating it
- Whether a data set on a level 0 volume is eligible to move to an ML1 volume or directly to an ML2 volume
- Whether to move the data sets on ML1 volumes to ML2 volumes
- For data sets on ML1 volumes, the minimum time to leave the data set unopened on both the level 0 volume and the ML1 volume before moving it to an ML2 volume
- A capability to migrate older generation data groups (GDGs)
- Whether to delete or migrate rolled-off generations if the data set was a member of a GDG
- Whether to transition the data set from one class of level 0 volumes to another, based on creation date, date since last referenced, or a specific periodic timeframe

DFSMSHsm uses the attributes from the storage group to determine which level 0 volumes to process for space management, how full the volume must be before data sets are migrated or transitioned from the volume, and how much free space should be on the volume after space management has finished.

As space management processes each volume, it first tries to make space on the volume by deleting expired and temporary data sets and by releasing over-allocated space in data sets that cannot be deleted. During extent reduction, DFSMSHsm releases any unused space in the data set, both internal and on the end, and places the data set on the same volume in as few extents as possible. If deleting data sets and releasing space does not make enough space available, DFSMSHsm uses the management class

to determine which data sets are eligible to transition to a different level 0 volume and which data sets are eligible to move to a migration volume.

Rule: All ML1 volumes must be DASD. If ML1 volumes become sufficiently full, DFSMSHsm moves migrated data sets from ML1 volumes to ML2 volumes as specified by either:

- The management class attributes for each data set
- The THRESHOLD parameter on the SETSYS ML1OVERFLOW command if you are using ML1 OVERFLOW volumes. See [“ML1OVERFLOW: Manage ML1 OVERFLOW volumes”](#) on page 1160.

ML2 volumes can be either DASD or tape. You can issue a DFSMSHsm command to move data sets from DASD ML2 volumes to tape ML2 volumes. As data sets on tape ML2 volumes become not valid, you can recycle the valid data sets to other tape ML2 volumes.

Recall, the reverse of migration, moves data sets from ML1 or ML2 volumes to level 0 volumes. With the recall function, DFSMSHsm returns data sets to storage groups as specified by SMS, whether the data set migrated from a volume in a storage group or not. This allows data sets that migrated before they were SMS-managed to be put under SMS management during recall. If a data set is recalled to an SMS-managed EAV volume, then the EATTR attribute of the data set is taken into consideration for level 0 volume selection.

DFSMS sysplex name support for DFSMSHsm automatic functions

Because DFSMSHsm supports the use of DFSMS sysplex names for the MIGRATE SYSTEM/SYSPLEX NAME, BACKUP SYSTEM/SYSPLEX NAME, and DUMP SYSTEM/SYSPLEX NAME storage group system name attributes, it handles each storage group for automatic space management, data availability management, and automatic dump processing as follows. How the storage groups are handled is referred to as the "system affinity" or "host affinity."

- If the storage group system name is a system name, the storage group is processed by that specific system.
- If the storage group system name is blank, the storage group can be processed by any system.
- If the storage group system name is a sysplex name, the storage group is processed by a subset of systems.

Each automatic function (automatic space management, data availability management, or automatic dump processing) is processed independently. Therefore, the various systems used by one automatic function will not affect the other automatic functions.

DFSMSHsm functions for non-SMS-managed volumes

For non-SMS-managed volumes, DFSMSHsm uses volume parameters to determine how each data set on the volume is managed.

Dump and restore functions

The dump function for non-SMS-managed volumes is governed by parameters of the DFSMSHsm commands, but it is performed the same way as for SMS-managed volumes. However, in addition to performing the automatic dump function for level 0 volumes, DFSMSHsm also performs automatic dump for ML1 volumes.

A data set dumped from a non-SMS-managed volume can be restored to an SMS-managed volume if the data set does not already exist, if SMS is active, and if the data set is indicated by the ACS routines to be SMS-managed.

Rule: Because the ML1 volumes are not managed by SMS—even in an environment in which all level 0 volumes are SMS-managed—you must use non-SMS-managed volume parameters to perform automatic dump for ML1 volumes.

Backup and recovery functions

The volume-basis operation of DFSMSHsm for non-SMS-managed volumes causes differences in how backup is performed. For non-SMS-managed volumes:

- If automatic backup is in effect for the volume, all changed data sets that DFSMSHsm can manage are backed up.
- All data sets processed by a particular DFSMSHsm host have the same backup frequency unless the frequency has been modified for that data set by a DFSMSHsm command.
- All data sets processed by a particular DFSMSHsm host have the same number of backup versions, unless the number of backup versions has been modified for that data set by a DFSMSHsm command.

In addition, you can specify the number of days to retain backup versions of non-SMS-managed data sets only by command parameters for the entire system complex.

A data set backed up from a non-SMS-managed volume can be recovered to an SMS-managed volume if the data set does not already exist, if SMS is active, and if the data set is indicated by the ACS routine to be SMS-managed.

How the system selects non-SMS extended address volume (EAV)

DFSMSHsm checks the data set level attribute EATTR when performing non-SMS volume selection. The EATTR data set level attribute specifies whether a data set can have extended attributes (format 8 and 9 DSCBs) and optionally reside in EAS on an extended address volume (EAV). The EATTR attribute is optional. If you do specify it, valid values are NO and OPT. For more information on the EATTR attribute, see [ALLOCATE](#) and [DEFINE CLUSTER](#) in *z/OS DFSMS Access Method Services Commands*.

If an EATTR value is not specified for a data set, the following default data set EATTR values are used:

- The default behavior for VSAM data sets is OPT.
- The default behavior for non-VSAM data sets is NO.

Preference for volume selection corresponds to the EATTR values for each data set:

- When EATTR = NO, the data set has no preference to any volume. For recalls, recovers and ARECOVERs the volume with the most amount of free space will be selected but only the track-managed free space will be used for determining the free space calculations of an eligible EAV. The data set will be allocated with a format 1 DSCB indicating that it is not eligible to have extents in the EAS.

If an EAV or non-EAV is chosen for allocation, **and** the value is EATTR=NO **and** the data set had a format 8 or 9 DSCB with vendor attributes, then the data set is allocated with a format 1 DSCB, and the system issues message ARC0784I to indicate that extended attributes have not been retained for this recall, recovery or ARECOVER. In this situation format 9 information of the data set will be lost on the recall, recovery or ARECOVER.

- When EATTR = OPT, the data set will prefer EAV volumes if the size of the allocation is equal to or greater than the BreakPointValue (BPV) specified. If the requested size of the data set allocation is smaller than the BPV then EAVs and non-EAVs will have equal preference.
 - If an EAV is chosen for allocation and the data set type is EAS-eligible, then the data set will be allocated with format 8 or 9 DSCBs in either the track-managed space or the EAS.
 - If a non-EAV is chosen for the allocation and the data set had format 8 or 9 DSCBs with vendor attributes, then the data set will be allocated with a format 1 and DFSMSHsm issues message ARC0784I signifying that extended attributes have not been retained for this recall, recovery or ARECOVER. The EATTR value will always remain in the format 1 DSCB.

EAV considerations for recovery functions

You can specify the EATTR value when allocating a data set to define whether the data set can have extended attributes (format 8 and 9 DSCBs) and optionally reside in EAS. In the event that a data set is being directed to a pre-allocated data set on a RECOVER or ARECOVER function, the system uses the

EATTR value of the pre-allocated data rather than the EATTR value specified for the data set source recovery data.

ARECOVER and EATTR considerations

The following methods can be used to allow the EATTR information from the ABACKUP site to be restored, along with the dataset:

- Use the DSCONFLICT(RENAMETARGET) parameter on the ARECOVER command, then manually delete the existing dataset. Since the ARECOVER resolved the conflict by renaming the existing dataset before restoring the ABARS copy, both copies will have their own EATTR. You only need to get rid of the renamed existing dataset.
- Use the DSCONFLICT(BYPASS) parameter on the ARECOVER command, manually delete the existing dataset and existing conflict resolution data set, then rerun the ARECOVER. Only the datasets that were bypassed or failed for some other reason on the first ARECOVER will be processed. Since no conflict exists anymore, you will get the ABARS copy, complete with its EATTR.

For more information, see the EATTR parameter of the [ALLOCATE](#) command in *z/OS DFSMS Access Method Services Commands*.

Space management function

Space management consists of removing appropriate data sets from the level 0 volumes and reducing the space occupied on the level 0 volumes by the data sets that remain there. In removing the data sets, you can have DFSMSHsm either delete them entirely or move them to other DASD or tape volumes. If you move the data sets to other DASD or tape volumes (ML1 or ML2), they can be automatically deleted from the migration volumes when they expire.

Differences in the way that DFSMSHsm performs space management for non-SMS-managed data sets include:

- The deletion age and management techniques are the same for all data sets on a volume managed by the deletion (delete by age) technique.
- The deletion age and management techniques are the same for all data sets on a volume managed by the retirement (delete if backed up) technique.
- The migration age and management techniques are the same for all data sets on a volume managed by the migration technique.
- The time for the data set to remain unopened before migration from ML1 to ML2 volumes is the same for all non-SMS-managed data sets in the system complex.
- Class transitions are not supported.

A data set migrated from a non-SMS-managed volume can be recalled to an SMS-managed volume if SMS is active and if the data set meets the conditions for selecting SMS management for that data set. DFSMSHsm recalls a non-SMS-managed data set to a non-SMS-managed volume only if the data set does not meet the conditions to become an SMS-managed data set, or if a command forces the recall to a non-SMS-managed volume.

Devices supported by DFSMSHsm

DFSMSHsm supports devices other than those specifically recommended in the examples in this document. For example, ML2 volumes can be either tape or DASD. Daily backup and spill backup volumes can also be either tape or DASD.

DFSMSHsm can be used on the following device types:

- All models of 3390, 9345, and 3380 DASD
- IBM 3494 and 3495 Tape Library Dataservers
- All models of 3590-1, 3490, 3480, 3480XF (with or without cartridge loaders), and 3592 (including Model J and Model E06, Model E06, and Model E07 tape drives)

- Cache storage controls for DFSMSHsm control data sets:
 - IBM 2105
 - IBM 2107
 - IBM 3990
 - IBM 3880 Model 23
 - IBM 3880 Model 13 (only indexes)
- Object Storage with OpenStack Swift API using Keystone or tempauth authorization

Note: DFSMSHsm normally uses only non-WORM media for non-ABARS functions. DFSMSHsm uses all media, including physical and logical WORM, for ABACKUP output processing. DFSMSHsm can use WORM media for non-ABARS processing and ARECOVER to tape, if specifically allowed by your installation. See the SETSYS TAPETYPE command for details.

Tip: For improved performance, the DFSMSHsm control data sets should be on cache storage controllers.

How to use the example system

Figure 2 on page 17 and Figure 3 on page 18 show two aspects of the example system used to explain DFSMSHsm. The figures are separated into SMS-related and non-SMS-related examples to simplify the discussions. Figure 4 on page 21 shows the unified system. The following major features affect the way in which the DFSMSHsm system commands are structured.

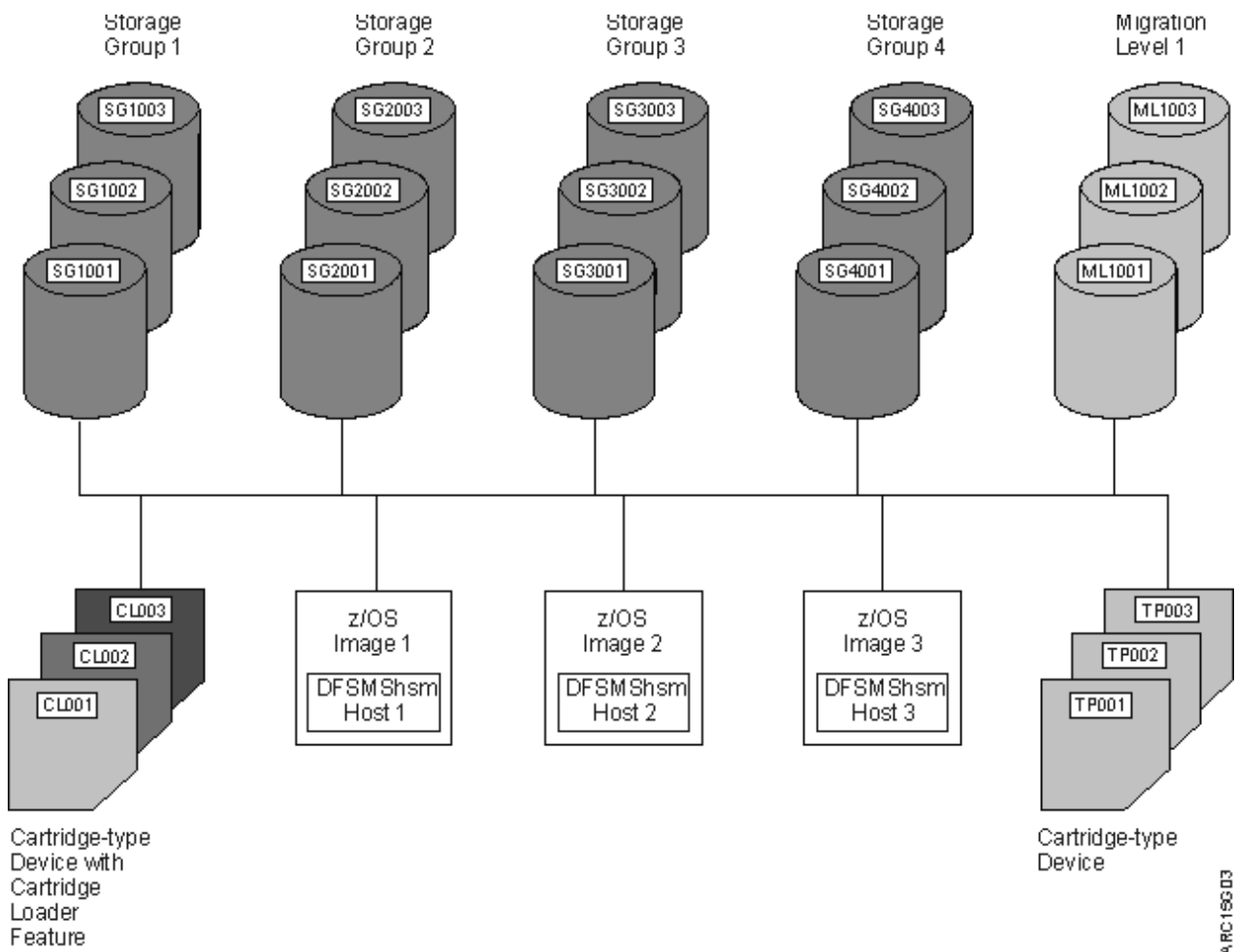


Figure 2. Example System of SMS-Managed Volumes

Note: Cartridge-type devices may be in a manual tape library (MTL) or in an automated tape library (ATL).

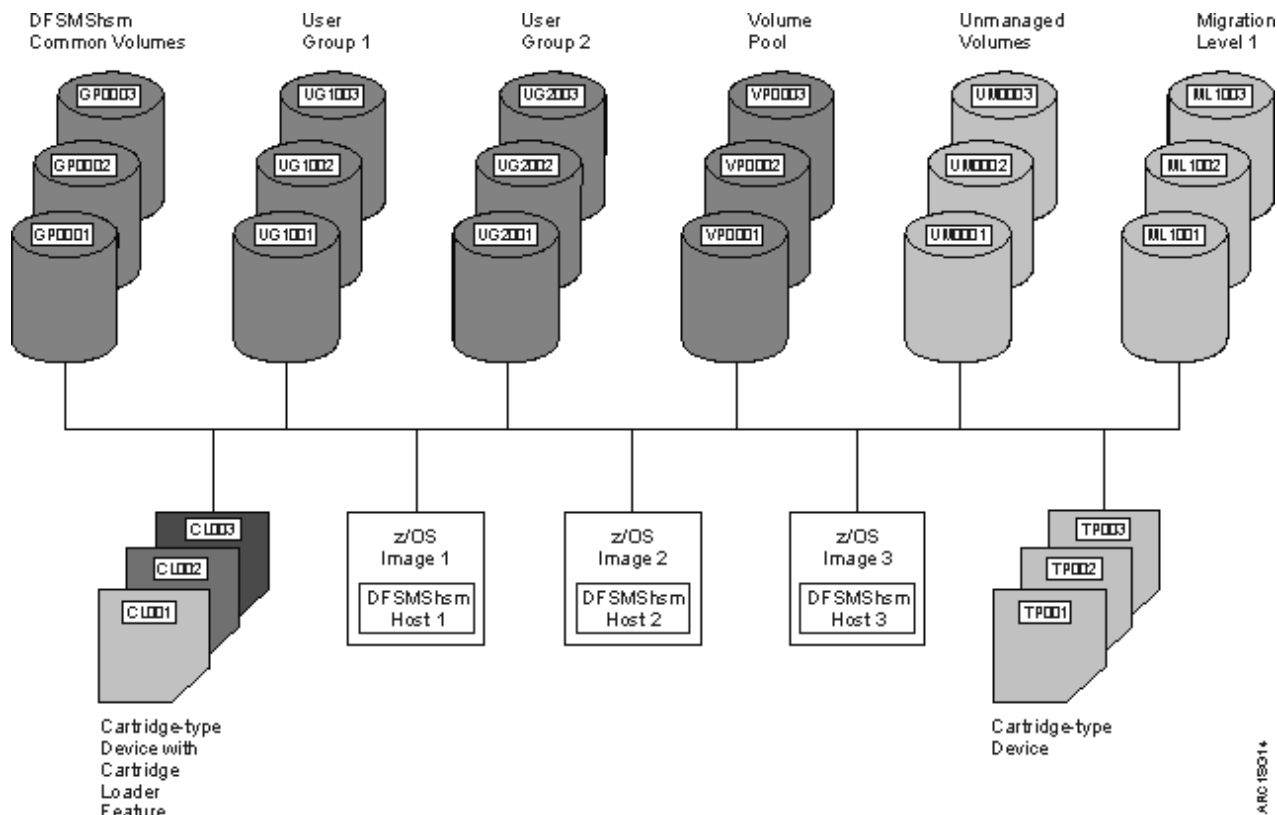


Figure 3. Example System of Non-SMS-Managed Volumes

Related reading

For more information about how to direct DFSMSHsm functions to a system-managed library, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Multiple z/OS images in the example system

Using multiple z/OS images allows the processing of a single DFSMSHsm host on one z/OS image or multiple DFSMSHsm hosts on more than one z/OS image. In the example system, there is one DFSMSHsm host per z/OS image. DFSMSHsm is shown running under JES2. However, when the use of a single z/OS image or of JES3 affects the selection of DFSMSHsm commands or the results attained with those commands, the differences that result from using JES3 instead of JES2 are explained.

Note: The z/OS images, tapes, and ML1 volumes shown in [Figure 2 on page 17](#) and [Figure 3 on page 18](#) are duplicated in both figures.

DASD storage in the example system

The example system shows only a limited number of DASD units in each classification. You can, of course, have any number of units in any of the classifications (up to the maximum channel and addressing capabilities of your system). All of the DASD is shared among all of the z/OS images. Where other sharing arrangements cause differences in the use of commands or in the results obtained, the differences are explained.

Storage groups 1 through 4

The example system uses DFSMSdftp to manage its storage. [Figure 2 on page 17](#) shows those elements of the system that are essential for using the SMS capability of DFSMSdftp. Storage groups 1 through 4 are the storage that is actually under control of the SMS at this time. The data sets on these volumes are the

ones referred to as SMS-managed. Expectations are that some of the DASD volumes shown in both this figure and in [Figure 3 on page 18](#) will be brought under SMS control as time goes on. This expectation applies only to the level 0 volumes, not to the DFSMSHsm-owned volumes.

User groups 1 and 2

User groups 1 and 2 ([Figure 3 on page 18](#)) are non-SMS-managed level 0 volumes that DFSMSHsm manages and that are reserved for particular groups of data sets. Data sets with particular high-level qualifiers in the data set names are directed to these volumes. Groups of volumes to which data sets are directed by their high-level qualifiers are known as *data set pools*.

Volume pool

The volume pool ([Figure 3 on page 18](#)) consists of non-SMS-managed volumes that are differentiated on some basis other than that of data set names. Data sets that are assigned to these volumes return to these volumes if they are migrated and recalled.

DFSMSHsm common volumes

The DFSMSHsm common volumes ([Figure 3 on page 18](#)) are the DFSMSHsm-managed volumes that contain non-SMS-managed data sets that do not belong on the volumes in either the user groups or the volume pools.

Unmanaged volumes

The unmanaged volumes ([Figure 3 on page 18](#)) are level 0 volumes that are not managed by either SMS or DFSMSHsm. DFSMSHsm operations can be performed on these volumes, but only by issuing commands. Automatic DFSMSHsm operations are not performed on these volumes.

Migration level 1

The ML1 volumes ([Figure 2 on page 17](#) and [Figure 3 on page 18](#)) are DFSMSHsm-owned volumes on which migrated data sets are stored. ML1 volumes are always DASD. As with the z/OS images, the ML1 volumes shown in both figures are the same volumes.

Tape devices in the example system

In the example system, there are two groups of tape devices, again duplicated in both figures. These devices are used for dump volumes, ML2 volumes, and backup volumes. ML2 and backup volumes can also use DASD.

Figures [Figure 2 on page 17](#), [Figure 3 on page 18](#), and [Figure 4 on page 21](#) show two groups of cartridge-type devices: those with cartridge loaders, and those without. The cartridge-type devices without cartridge loaders are used (in the example) mainly as input devices for recall and recovery because recall or recovery is always from a specific tape. The cartridge-type devices with cartridge loaders are used for output to backup, dump, and ML2 tapes because output can use scratch tapes (tapes that no longer contain valid data, and have been returned to the global scratch pool) and take advantage of the cartridge loader's ability to change tapes automatically.

Note: Cartridge-type devices may be in an MTL or an ATL.

Related reading

For more information about manual tape libraries (MTLs) and automated tape libraries (ATLs), see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Other assumptions in the example system

In order to understand how to use the example system, other assumptions must be made concerning your knowledge and concerning security systems for the example system.

Your knowledge

There is an assumption that you know about z/OS, DFSMSdfp, TSO, and RACF, a component of the Security Server for z/OS.

Under this assumption, if you are told to perform an action related to these products, you know how to perform that action without a discussion being necessary.

Security systems

So that you can be shown what DFSMSHsm does in relation to all types of data set protection, it is assumed that the example system provides the following:

- RACF or equivalent security system
- Expiration date protection for tapes

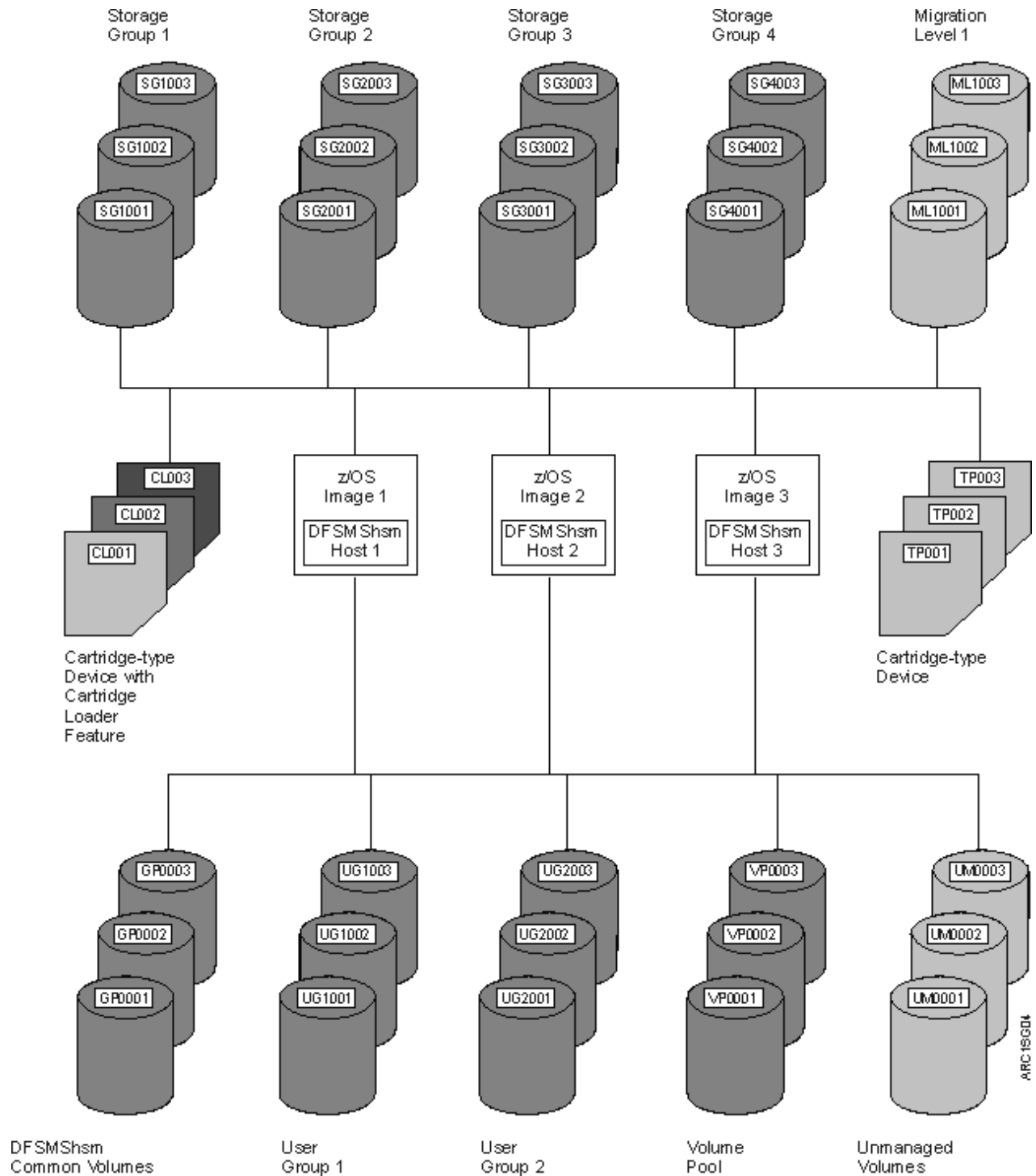


Figure 4. Comprehensive Example System

Chapter 2. Space management of SMS-managed storage

DFSMSHsm manages SMS-managed storage by data sets. That is, the specification for how to treat each data set on a volume is contained in the management class associated with that data set. The storage management subsystem determines to which volume the data set is recalled.

Related reading

To view the consolidation of all of the DFSMSHsm commands, storage group attributes, and management class attributes that show the automatic storage management of the example system, see [Chapter 21, “Consolidated commands, storage group definitions, and management classes for the example system,”](#) on page 653.

Space management functions

This topic describes the space management functions for managing SMS-managed storage:

- [“Automatic primary space management”](#) on page 23
- [“Automatic secondary space management”](#) on page 24
- [“Event-driven migration”](#) on page 24
- [“Automatic recall”](#) on page 32
- [“Automatic dynamic volume expansion space management”](#) on page 32
- [“Command space management”](#) on page 32

Related reading

Before performing space management functions on extended address volumes (EAVs) you need to understand how EAV space is managed. For details, see [“Space management of extended address volumes \(EAVs\)”](#) on page 127 and review recommendations in [“Recommendations for using extended address volumes \(EAVs\)”](#) on page 129.

Automatic primary space management

During automatic primary space management:

- All DFSMSHsm hosts delete temporary data sets and expired data sets from the DFSMSHsm-managed volumes they are processing. This is done under the control of the management class that is associated with each data set on the volume or the expiration date in the data set's volume table of contents (VTOC) entry.
- Under control of the data set management classes, all DFSMSHsm hosts release unused allocated space in physical sequential, partitioned, and extended format Virtual Storage Access Method (VSAM) data sets.
- During data set and volume processing, fast subsequent migration reconnects eligible data sets to the ML2 tape or cloud from which they were most recently recalled.
- Under the extent reduction function, all DFSMSHsm hosts also reduce the number of extents of physical sequential, partitioned, and direct access data sets that have exceeded a specified number of extents. During the process of extent reduction, they also release any unused space in the data sets and compress partitioned data sets.
- Data sets that are eligible for a class transition are transitioned. If a data set is eligible for both a class transition and migration, the data set will only be migrated. This prevents the data set from being moved from one level 0 volume to another for a transition, only to be later migrated to a migration volume.

- Data sets that are eligible for migration are migrated.

Primary space management continues until the SMS-managed volumes have the specified amount of free space. However, if deletion of expired data sets, fast subsequent migration, and space reduction of the remaining data sets achieves the specified volume free space low threshold, no actual data sets are moved.

If data sets are to be migrated during primary space management, they will be migrated in compacted form if possible. Data sets expected to be smaller than 110 KB (where 1 KB is 1024 bytes) after they are compacted are candidates for small data set packing (SDSP) if enabled.

Related reading

- For more information about fast subsequent migration, see [“Reconnecting data sets using fast subsequent migration”](#) on page 94.
- For other space management considerations, see [Chapter 4, “Other space management considerations,”](#) on page 127.
- For more information about space management of extended address volumes, see [“Space management of extended address volumes \(EAVs\)”](#) on page 127.

Automatic secondary space management

Automatic secondary space management schedules TAPECOPY commands for migration tape copy needed (TCN) records, deletes expired data sets from the *migration volumes*, deletes obsolete MCDs, VSRs, and DSRs during migration cleanup, and moves data sets (under control of the management class) from ML1 to ML2 volumes.

On the first day of the secondary space management cycle, DFSMSHsm will examine the containers in the clouds that are defined to DFSMSHsm and look for containers created using the current HSMplex name that do not have objects. Empty containers are deleted. If the end of the **Secondary Space Management** window has been reached and not all containers have been examined, the processing will continue in the next **Secondary Space Management** window.

Event-driven migration

Event-driven migration performs space management functions similar to primary space management, except it is event-driven instead of scheduled. Event-driven migration automatically and periodically moves low-activity data sets from DFSMSHsm-managed volumes to DFSMSHsm-owned volumes, transitions data sets from one class of user-accessible volumes to another, and reduces the space occupied by data sets on DFSMSHsm-managed volumes.

There are two event-driven migration methods:

- Interval migration. See [“Using interval migration”](#) on page 25.
- On-demand migration. See [“Using on-demand migration”](#) on page 27.

Related reading

- For a comparison of interval migration and tape mount management, see [“Comparison of interval migration and tape mount management”](#) on page 30.
- For an illustrative event-driven migration scenario, see [“Event-driven migration example scenario”](#) on page 81.
- For other space management considerations, see [Chapter 4, “Other space management considerations,”](#) on page 127.
- For more information about space management extended address volumes, see [“Space management of extended address volumes \(EAVs\)”](#) on page 127.
- For more information about how to quiesce event-driven migration during specific time windows, see [“EVENTDRIVENMIGRATION: Releasing Interval and On-demand Migration”](#) on page 1034.

Using interval migration

Interval migration is a space management function that performs *hourly* space management on eligible volumes that exceed specified volume threshold requirements. Interval migration migrates data sets to ML1 or ML2 volumes and transitions data sets from one class of user-accessible volumes to another until the low threshold is reached or there are no eligible data sets remaining.

During interval migration:

- Information about the SMS configuration is obtained:
 - A list of all volumes in all DFSMSHsm-managed storage groups is obtained.
 - A list of active management classes is obtained.
- The percent of allocated space on each volume in the obtained list is checked.
- A message is issued for unmatched data sets.
- Temporary data sets are deleted.
- Expired data sets are deleted.
- Data sets that are eligible for a class transition are transitioned. If a data set is eligible for both a class transition and migration, the data set will only be migrated. This prevents the data set from being moved for a transition, only to be later migrated.
- Data sets that are eligible for migration are migrated.

The following functions that are performed during primary space management are not performed during interval migration:

- Partial release.
- Extent reduction.

Specific interval migration eligibility requirements must be satisfied both at the volume level and the data set level. See [Interval migration eligibility requirements](#).

Enabling interval migration

Interval migration is enabled by default and there are two methods of explicitly enabling interval migration:

- Add the SETSYS INTERVALMIGRATION command to the ARCCMDxx parmlib member of SYS1.PARMLIB. Then restart DFSMSHsm.
- Issue the SETSYS INTERVALMIGRATION command after DFSMSHsm is started.

To determine if interval migration is enabled or disabled, issue QUERY SETSYS. Message ARC0151I is issued with the query response.

To use interval migration after it is enabled, eligibility requirements must be satisfied. See [Interval migration eligibility requirements](#).

Disabling interval migration

There are two methods of disabling interval migration:

- Add the SETSYS NOINTERVALMIGRATION command to the ARCCMDxx parmlib member of SYS1.PARMLIB. Then restart DFSMSHsm.
- Issue the SETSYS NOINTERVALMIGRATION command after DFSMSHsm is started.

Interval migration eligibility requirements

There are two sets of requirements that must be satisfied in order for a volume to be eligible for interval migration. The first set of requirements must be satisfied at the volume level. The second set of requirements must be satisfied at the data set level.

Requirements that must be satisfied at the *volume* level for interval migration to be performed are:

- The volume is either an SMS-managed or non-SMS-managed volume.
- The volume belongs to a storage group defined with AM=I (Auto Migrate, interval) or AM=Y (Auto Migrate, yes).
 - If the storage group is defined with AM=I, the volume must exceed the midpoint between the low and high threshold to be eligible.
 - If the storage group is defined with AM=Y, the volume must exceed the high threshold to be eligible.
 - If the storage group is defined with AM=Y, on-demand migration must not be enabled for the volume to be eligible. Otherwise, on-demand migration is performed.
- The SETSYS INTERVALMIGRATION setting is in effect.

Requirements that must be satisfied at the *data set* level for a data set to be transitioned or migrated from an eligible volume are:

- The management class for the data set is configured to allow both command or auto migration (COMMAND OR AUTO MIGRATE=BOTH)
- - For transitions:
 - The age of the data set has exceeded the management class TIME SINCE CREATION attribute value.
 - The last referenced date has met or exceeded the management class TIME SINCE LAST USE attribute value.
 - The current date has met or exceeded the PERIODIC time period specified in the management class.
- For migration, the inactive age of the data set has exceeded the management class PRIMARY DAYS NON-USAGE attribute value.
- If the data set is a candidate to be backed up:
 - It is not eligible for migration if the migration of the data set is directed to tape.
 - It is not eligible for a transition if the management class TRANSITION COPY TECHNIQUE specifies a technique other than STANDARD.

Interval migration usage notes and other considerations

1. Neither interval migration or on-demand migration is performed on SMS-managed volumes that belong to a storage group defined with AM=P (Auto Migrate, primary) or AM=N (Auto Migrate, no).
2. Interval migration is performed on volumes that belong to storage groups defined with AM=I regardless of whether interval migration is enabled or disabled.
3. When on-demand migration is enabled, on-demand migration is performed instead of interval migration on eligible SMS-managed volumes that belong to storage groups defined with AM=Y. For on-demand eligibility requirements, see [On-demand migration eligibility requirements](#).
4. If you want to require operator permission before allowing interval migration to run, issue the SETSYS REQUEST command.
5. If you want the operator to be notified of the space allocation of each volume during the interval migration space check, issue the SETSYS MONITOR(SPACE) command.
6. Data sets on SMS-managed volumes are processed in the following order:
 - a. Data sets eligible for a class transition and migration will be migrated.
 - b. Data sets eligible for a class transition will be transitioned.
 - c. Data sets eligible for migration only will be migrated. For this phase, data set size is a determining factor for the order and priority in which data sets are migrated. The largest eligible data set

migrates first, followed by the next largest. The priority for migrating data sets from volumes that belong to storage groups defined with AM=I applies when the low threshold exceeds 5% only.

7. In order for a data set to be migrated or transitioned by TIME SINCE LAST USE, the date of the current backup copy should be greater than the date-last-referenced by at least one day.
8. The current time is outside the time window(s) specified in SETSYS EVENTDRIVENMIGRATIONQUIESCE.

Related reading

- For more information about on-demand migration, see [“Using on-demand migration” on page 27](#).
- For more information about the QUERY command and responses, see [Chapter 58, “QUERY command: Displaying the status of parameters, statistics, and requests,” on page 981](#) and [Chapter 77, “Using the QUERY command,” on page 1389](#).
- For more information about the SETSYS commands used, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#).
- For a comparison of interval migration and tape mount management, see [“Comparison of interval migration and tape mount management” on page 30](#).
- For more information about managing storage and space requirements, see [“Specifying the storage to be managed and the space requirements” on page 32](#).
- For more information about exceptions for which management classes are not included in the obtained list, see [“DFSMSHsm retrieval of management classes and storage groups” on page 154](#).
- For an illustrative event-driven migration scenario, see [“Event-driven migration example scenario” on page 81](#).
- For other space management considerations, see [Chapter 4, “Other space management considerations,” on page 127](#).
- For more information about space management of extended address volumes, see [“Space management of extended address volumes \(EAVs\)” on page 127](#).
- For more information about storage groups and management classes, see [z/OS DFSMSdfp Storage Administration](#).

Using on-demand migration

On-demand migration is a space management function that performs *near immediate* space management on eligible SMS-managed volumes that exceed the specified volume high threshold, instead of waiting for interval migration to run at the top of each hour. On-demand migration migrates data sets to ML1 or ML2 volumes and transitions data sets from one class of user-accessible volumes to another until the low threshold is reached or there are no eligible data sets remaining.

During on-demand migration:

- DFSMSHsm listens for an ENF72 event code signal from SMS indicating a volume has exceeded the high threshold and queues the volume for processing.
- An eligible volume is selected from the queue for space management.
- The percent of allocated space on the selected volume is checked.
- A message is issued for unmatched data sets.
- Temporary data sets are deleted.
- Data sets that are eligible for class transitions are transitioned. If a data set is eligible for both a class transition and migration, the data set will only be migrated. This prevents the data set from being moved for a transition, only to be later migrated.
- Expired data sets are deleted.
- Data sets that are eligible for migration are migrated.

The following functions that are performed during primary space management are not performed during on-demand migration:

- Partial release.
- Extent reduction.

Specific on-demand migration eligibility requirements must be satisfied both at the volume level and the data set level. See [On-demand migration eligibility requirements](#).

Enabling on-demand migration

There are two methods of enabling on-demand migration:

- Add the SETSYS ONDEMANDMIGRATION(Y) command to the ARCCMDxx parmlib member of SYS1.PARMLIB. Then restart DFSMSHsm.
- Issue the SETSYS ONDEMANDMIGRATION(Y) command after DFSMSHsm is started.

On-demand migration is disabled by default.

To determine if on-demand migration is enabled or disabled, issue QUERY SETSYS. Message ARC0139I is issued with the query response.

To use on-demand migration after it is enabled, eligibility requirements must be satisfied. See [On-demand migration eligibility requirements](#).

Disabling on-demand migration

There are two methods of disabling on-demand migration:

- Add the SETSYS ONDEMANDMIGRATION(N) command to the ARCCMDxx parmlib member of SYS1.PARMLIB. Then restart DFSMSHsm.
- Issue the SETSYS ONDEMANDMIGRATION(N) command after DFSMSHsm is started.

On-demand migration eligibility requirements

There are two sets of requirements that must be satisfied in order for a volume to be eligible for on-demand migration. The first set of requirements must be satisfied at the volume level. The second set of requirements must be satisfied at the data set level.

Requirements that must be satisfied at the *volume* level for on-demand migration to be performed are:

- The volume is an SMS-managed volume.
- The volume belongs to a storage group defined with AM=Y (Auto Migrate, yes).
- The volume exceeds the high threshold.
- The SETSYS ONDEMANDMIGRATION(Y) setting is in effect.

Requirements that must be satisfied at the *data set* level for a data set to be migrated or transitioned from an eligible volume are:

- The management class for the data set is configured to allow both command or auto migration (COMMAND OR AUTO MIGRATE=BOTH)
- For transitions:
 - The age of the data set has exceeded the management class TIME SINCE CREATION attribute value.
 - The last referenced date has met or exceeded the management class TIME SINCE LAST USE attribute value.
 - The current date has met or exceeded the PERIODIC time period specified in the management class.

- For migrations, the inactive age of the data set has exceeded the management class PRIMARY DAYS NON-USAGE attribute value.
- If the data set is a candidate to be backed up:
 - It is not eligible for migration if the migration of the data set is directed to tape.
 - It is not eligible for a transition if the management class TRANSITION COPY TECHNIQUE specifies a technique other than STANDARD.

On-demand migration is **not** performed on non-SMS-managed volumes or SMS-managed volumes that belong to a storage group defined with AM=I (Auto Migrate, interval). Interval migration is performed instead.

On-demand migration in a multiple DFSMSHsm host environment

In a multiple DFSMSHsm host environment:

- Each host configured to perform on-demand migration selects eligible volumes for on-demand migration up to the configured maximum number of migration tasks (that is, the SETSYS MAXINTERVALTASKS command). As volumes are processed, each host selects from the remaining volumes until all volumes have been processed.

Note: MAXEVENTDRIVENMIGTASKS or the MAXEDMT alias can be used instead of the MAXINTERVALTASKS parameter.

- You should distribute on-demand migration workload evenly across all hosts configured to perform on-demand migration.
- You should ensure there are enough hosts to perform other critical DFSMSHsm tasks.

On-demand migration usage notes and other considerations

1. Neither interval migration or on-demand migration is performed on SMS-managed volumes that belong to a storage group defined with AM=P (Auto Migrate, primary) or AM=N (Auto Migrate, no).
2. On-demand migration is paused when interval migration or primary space management are started on the same DFSMSHsm host performing on-demand migration. On-demand migration will start again after interval migration or primary space management has ended, if there are eligible volumes to be processed.
3. Notification of the number of volumes concurrently queued for on-demand migration is issued to the operator's console (highlighted message ARC1901E). The default notification limit is 100 volumes. For more information about configuring the notification limit, see [“Specifying an on-demand migration number of concurrently queued volumes notification limit” on page 52](#)
4. After a volume is processed using on-demand migration it can remain at or above the high threshold. This happens when there are not enough eligible data sets on the volume to be space managed. To prevent continuously processing a volume that remains above the high threshold, an on-demand migration timer is started. This timer prevents on-demand migration from running again for 24 hours (default) on all volumes that remained at or above the high threshold during its duration. When the timer expires, on-demand migration is performed again on all of the volumes that remain at or above the high threshold. A patch is provided for [Changing the frequency](#) at which volumes at or above the high threshold are selected for on-demand migration again. See [z/OS DFSMSHsm Implementation and Customization Guide](#).
5. When on-demand migration is enabled, the SETSYS INTERVALMIGRATION setting and the SETSYS NOINTERVALMIGRATION setting are ignored for on-demand migration eligible volumes.
6. If you want to perform on-demand migration on TMM storage groups, you must convert AM=I (Auto Migrate, interval) TMM storage groups to AM=Y.
7. If you want the operator to be notified of the space allocation of each volume during the on-demand migration space check, issue the SETSYS MONITOR(SPACE) command.
8. Data sets on SMS-managed volumes are processed in the following order:

- a. Data sets eligible for a class transition and migration will be migrated.
 - b. Data sets eligible for a class transition will be transitioned.
 - c. Data sets eligible for migration only will be migrated. For this phase, data set size is a determining factor for the order and priority in which data sets are migrated. The largest eligible data set migrates first, followed by the next largest. The priority for migrating data sets from volumes that belong to storage groups defined with AM=I applies when the low threshold exceeds 5% only.
9. In order for a data set to be migrated or transitioned, the date of the current backup copy should be greater than the date-last-referenced by at least one day.
10. The current time is outside the time window(s) specified in SETSYS
EVENTDRIVENMIGRATIONQUIESCE.

Related reading

- For more information about interval migration, see [“Using interval migration”](#) on page 25.
- For more information about the QUERY command and responses, see Chapter 58, “QUERY command: Displaying the status of parameters, statistics, and requests,” on page 981 and Chapter 77, “Using the QUERY command,” on page 1389.
- For more information about the SETSYS commands used, see Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.
- For more information about managing storage and space requirements, see [“Specifying the storage to be managed and the space requirements”](#) on page 32.
- For an illustrative event-driven migration scenario, see [“Event-driven migration example scenario”](#) on page 81.
- For other space management considerations, see Chapter 4, “Other space management considerations,” on page 127.
- For more information about space management of extended address volumes, see [“Space management of extended address volumes \(EAVs\)”](#) on page 127.
- For more information about storage groups and management classes, see [z/OS DFSMSdfp Storage Administration](#).

Comparison of interval migration and tape mount management

Tape mount management (TMM) is a methodology that is used to optimize tape subsystem operation and use. It consists of hardware and software facilities, which manage tape data efficiently. By using DFSMSHsm in conjunction with your ACS routines to implement the TMM methodology, you can redirect multiple output data sets to DASD, then later have them moved to a single tape, with a single tape mount. This not only improves tape usage, but also greatly reduces the number of tape mounts that are required.

The following table illustrates the differences between normal interval migration processing and TMM processing:

Table 2. Standard Interval Migration Processing Compared to TMM Processing

DFSMShsm Options or Controls	Interval Migration	TMM with SMS
Hourly option Note: Hourly migration is not performed on a host that is currently running primary space management.	Storage group having AM=Y (Auto Migrate, yes) Note: Specifying AM=Y for a storage group requests that primary space management and event-driven migration both be performed on volumes in this storage group. An option of AM=P (Auto Migrate, primary) indicates only primary space management is performed.	Storage group having AM=I (Auto Migrate, interval) Note: This option is referred to as tape mount management (TMM).
SETSYS INTERVALMIGRATION	Volumes meeting trigger threshold may be processed hourly on this host.	Volumes meeting trigger threshold may be processed hourly on this host.
SETSYS NOINTERVALMIGRATION	Volumes are not processed hourly on this host.	Volumes meeting trigger threshold may be processed hourly on this host.
Threshold range	1–99	0–99
Trigger threshold	High threshold	Midpoint between high and low threshold.
Max interval tasks of 0	No hourly migration will be done on this host. Note: The option of NOINTERVALMIGRATION provides the same capability for this category of data.	No hourly migration will be done on this host. Note: Be aware that if you set the SETSYS MAXINTERVALTASKS parameter to zero, you are preventing hourly migration for all storage groups and non-SMS volumes on this host.
Max interval tasks x	x tasks available to do hourly migration of volumes meeting trigger threshold.	x tasks available to do hourly migrations of volumes meeting trigger threshold.
AMHOST=blank	Hourly migration can be performed on any host permitting interval migration and having interval tasks.	Hourly migration can be performed on any host having interval tasks.
AMHOST=system or system group name Note: If you want to allow hourly processing on more than one host but not on all hosts of a sysplex, then you can specify the combination of a sysplex restricting name and also specify SETSYS MAXINTERVALTASKS(0) on the hosts where you do not want to permit hourly migration.	Primary space management and hourly migration limited to this system or group.	Primary space management and hourly migration limited to this system or group.

Automatic recall

If a data set that has migrated is needed for processing, DFSMSHsm recalls the data set from the migration volume on which it resides to an SMS-managed volume. The storage management subsystem selects the SMS-managed volume to which the data set is returned. Migrated extended format VSAM data sets can be recalled only as extended format VSAM data sets. Migrated extended format sequential access method (SAM) data sets can be recalled as extended format or nonextended format data sets.

If a data set is assigned to a management class with no defined transition, then the allocation will be performed using the standard ACS routines.

If a data set is assigned to a management class with a defined transition, then the allocation will be done as follows:

- If the data set has not yet transitioned nor missed a transition, then the allocation will be performed using the standard ACS routines.
- If the data set had transitioned before being migrated, then the allocation will be performed using the class transition ACS routines to ensure that the data set is allocated back to the appropriate level.
- If the data set missed one or more transitions during the period in which it was migrated, then the allocation will be performed using the class transition ACS routines. If multiple transitions were missed, then the class transition ACS routines will be invoked for each transition that was missed (up to 5 times), to ensure that the data set is allocated on the appropriate level.

Note: To avoid circularity, DFSMSHsm will limit the number of repeated ACS routine invocations to five.

Automatic dynamic volume expansion space management

Certain DASD controllers have the ability to dynamically change the size of a logical volume. After the size of a volume is dynamically changed, the ICKDSF REFORMAT command must be run to rebuild the VTOC and INDEX on the volume. Until this is done, DFSMSHsm will not be aware of the new volume size. After the REFORMAT has been processed, DFSMSHsm automatically updates its volume records so that future processing will use the new volume size.

Note: DFSMSHsm only gets notified of volume changes when the VTOC REFORMAT is done on the same sysplex. If done on another sysplex, DFSMSHsm is not notified and will continue to use the original size until DFSMSHsm is restarted.

Command space management

DFSMSHsm provides you with the ability, by issuing individual commands, to perform space management functions. For associated command space management tasks, see [“Command space management tasks”](#) on page 88.

Tasks for automatic space management

The tasks for controlling automatic space management of SMS-managed storage require that you add DFSMSHsm commands to the ARCCMDxx member and that you specify attributes in the SMS storage groups and management classes. It is assumed that you have already defined the storage groups and management classes. The descriptions within this document are of attributes that you need to add to the previously defined storage groups and management classes. Similarly, the descriptions of DFSMSHsm commands relate to commands to be added to the ARCCMDxx member of SYS1.PARMLIB.

Specifying the storage to be managed and the space requirements

You must configure what storage DFSMSHsm manages and how much space to free on the volumes. This configuration specification comes from the storage group that the volume belongs to.

You can use the Auto Migrate attribute to specify whether DFSMSHsm performs primary space management and event-driven migration on the volumes that are assigned to the storage group. The Auto Migrate attribute specifies the type of space management that DFSMSHsm uses on the volumes that

are assigned to the storage group. Table 3 on page 33 lists the acceptable values of the Auto Migrate attribute and the space management methods that are performed based on those values.

<i>Table 3. Storage group Auto Migrate attribute values</i>			
Auto Migrate (AM) Value	Primary Space Management Performed?	Interval Migration Performed?	On-demand Migration Performed?
Y (yes)	Yes	Yes ¹	Yes ²
I (interval)	Yes	Yes	No
P (primary)	Yes	No	No
N (no)³	No	No	No

Notes:

1. Depends on whether the setting SETSYS INTERVALMIGRATION or SETSYS NOINTERVALMIGRATION is specified and whether the setting SETSYS ONDEMANDMIGRATION(Y) or SETSYS ONDEMANDMIGRATION(N) is specified.
2. Depends on whether the setting SETSYS ONDEMANDMIGRATION(Y) or SETSYS ONDEMANDMIGRATION(N) is specified.
3. Used for volumes that are managed by other products, or when you do not want space management, such as for application database or system volumes that must remain DASD-resident for other reasons.

A storage group defined with AM=Y indicates that you want the storage group space managed by primary space management, and, on occasion, by event-driven migration. If the setting SETSYS INTERVALMIGRATION is specified, volumes that belong to the storage group are eligible for interval migration. If the setting SETSYS ONDEMANDMIGRATION(Y) is specified, volumes that belong to the storage group are eligible for on-demand migration and are not eligible for interval migration, even if SETSYS INTERVALMIGRATION is specified. Specific interval migration or on-demand migration requirements must be satisfied for a volume to be eligible.

A storage group defined with AM=I indicates that you want the storage group space managed by interval migration. Primary space management can also be performed, but plays a minor role overall. Processing of a storage group defined with AM=I by interval migration is not affected by the SETSYS INTERVALMIGRATION or SETSYS NOINTERVALMIGRATION settings that apply to storage groups that are defined with AM=Y. Defining a storage group with AM=I is ideal for use with tape mount management (TMM), which redirects tape data sets to the storage group, and then lets DFSMSHsm move them to tape hourly, as needed.

AM=I storage groups also have a different trigger for interval migration to be performed. That is, when a volume exceeds the midpoint between low and high threshold, interval migration is performed. For example, if low threshold is 20 percent and the high threshold is 70 percent, then volumes that exceed 45 percent are processed.

A storage group defined with AM=P indicates that you want the storage group managed by primary space management only. This is useful when event-driven migration is determined to consume too much resource or is disruptive in other ways. Processing of a storage group defined with AM=P by primary space management is not affected by the SETSYS INTERVALMIGRATION or SETSYS ONDEMANDMIGRATION settings that apply to storage groups defined with AM=Y.

A storage group defined with AM=N indicates that you do not want automatic space management to be performed. Data sets and volumes must be migrated using commands instead. Defining a storage group with AM=N is ideal when the volumes are managed by other products or when you do not want space management performed (such as for application database or system volumes that must remain DASD-resident for other reasons).

Example

In Figure 5 on page 34, primary space management is performed on storage groups SG1 and SG2 (although event-driven migration is also performed on these storage groups). Interval migration is performed on storage group SG3. Storage group SG4 is not space managed, but has thresholds to control new data set allocation.

```
NAME ===> SG1
MIGRATE SYSTEM/SYSPLEX NAME ===>
AUTO MIGRATE ===> Y
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ===> 85 LOW ===> 70

NAME ===> SG2
MIGRATE SYSTEM/SYSPLEX NAME ===>
AUTO MIGRATE ===> Y
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ===> 95 LOW ===> 80

NAME ===> SG3
MIGRATE SYSTEM/SYSPLEX NAME ===>
AUTO MIGRATE ===> I
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ===> 20 LOW ===> 0

NAME ===> SG4
MIGRATE SYSTEM/SYSPLEX NAME ===>
AUTO MIGRATE ===> N
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ===> 95 LOW ===> 80
```

Figure 5. Storage Group Attributes for Space Management

The **Migration and Allocation Thresholds** fields configure two DFSMSHsm controls:

- How much free space is available on the volumes when space management is finished (low)
- How full the volumes must be to cause space management to be performed (high)

“Pass 1 of automatic primary space management” on page 65 (non-data-movement function in primary space management) and extent reduction are performed for all volumes in storage groups defined with AM=Y and AM=I (SG1, SG2, and SG3). “Pass 2 of automatic primary space management” on page 70 (migration) is performed only for those volumes that are above the low threshold after pass 1 and extent reduction have completed.

SMS tries to stay within these thresholds by assessing the primary space allocation of each data set before assigning it to a given DASD volume. When a volume reaches or exceeds the high threshold, SMS issues an ENF72 signal, regardless of the Auto Migrate setting. This event code is the trigger for on-demand migration.

DFSMSHsm uses the high threshold attribute to determine the need for event-driven migration. The low threshold attribute specifies the maximum level of occupied space after space management is performed.

Space management is considered complete when both thresholds have been met. This applies only to interval migration of non-SMS-managed volumes and event-driven migration of SMS-managed volumes when Auto Migrate is not I. Interval migration of SMS-managed volumes that belong to a storage group defined with AM=I is performed if the volume is at or above the midpoint between the low and high threshold.

Note: A low threshold of 0 is permitted only with AM=I.

Related reading

- For more information about primary space management, see [“Automatic primary space management” on page 23](#).
- For more information about event-driven migration, see [“Event-driven migration” on page 24](#).
- For more information about storage groups and their functions in SMS, see [z/OS DFSMSdfp Storage Administration](#).

Specifying the DFSMSHsm hosts that process each storage group

When DFSMSHsm processes an SMS-managed volume for space management, it records the time at which the volume was processed for cloud or non-cloud migrations. If another DFSMSHsm host attempts to process the same volume for space management to cloud or non-cloud migrations, it checks the time at which the volume was last processed. If the volume was processed within the last 14 hours, DFSMSHsm does not process the volume again for that type of target. A volume may be processed by one host for non-cloud migrations, and later processed by another host for cloud migrations.

Because of the time-recording process for SMS-managed volumes, DFSMSHsm does not require you to control which DFSMSHsm host processes which volumes, and you should not do such manual load balancing. However, should you find it necessary to limit processing of certain volumes to one z/OS image, you can do so with the MIGRATE SYSTEM/SYSPLEX NAME attribute in the storage group.

The system name is the system-wide identifier for the z/OS image, not the identifier assigned with the HOSTID parameter of the DFSMSHsm startup procedure.

If only z/OS to process storage group SG1 for space management, you could specify:

```
Storage group 1 (SG1)
MIGRATE SYSTEM/SYSPLEX NAME ==> PRC1
```

If you find that a single DFSMSHsm system is insufficient to complete processing of a storage group, the SETSYS STORAGEGROUPAFFINITY command can be used to allow a particular DFSMSHsm host to process storage groups assigned to other systems.

To allow DFSMSHsm on z/OS image 2 to process the storage group in the example above, you would specify the following:

```
STORAGEGROUPAFFINITY to DFSMSHsm on
z/OS image 2
F HSMIMG2,SETSYS STORAGEGROUPAFFINITY(PRC1)
```

Specifying how groups of data sets are managed

During space management, DFSMSHsm processes individual SMS-managed data sets to comply with the attributes specified in the management classes with which the data sets are associated. Management class attribute groups let you specify:

- Expiration attributes
 - EXPIRE AFTER DAYS NON-USAGE
 - EXPIRE AFTER DATE/DAYS
 - RETENTION LIMIT
- Partial release attribute
- Migration attributes
 - COMMAND OR AUTO MIGRATE
 - PRIMARY DAYS NON-USAGE
 - SIZE LESS THAN OR EQUAL TO ACTION
 - SIZE GREATER THAN ACTION
 - LEVEL 1 DAYS NON-USAGE
 - LEVEL 2 DAYS NON-USAGE
 - CLOUD NAME
- Class transition attributes
 - TIME SINCE CREATION | TIME SINCE LAST USE | PERIODIC
 - TRANSITION COPY TECHNIQUE
 - SERIALIZATION ERROR EXIT
- GDG management attributes

GDG ELEMENTS ON PRIMARY
ROLLED-OFF GDS ACTION

Specifying expiration attributes

The expiration attributes, EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS, govern how long a data set associated with the management class is kept if no expiration date is listed in the data set VTOC entry or if the VTOC entry contains an INVALID EXPIRATION DATE. The expiration date is in the data set VTOC entry if the RETENTION LIMIT attribute does not specify a value of zero to disallow the use of explicit expiration dates and the expiration date is explicitly specified by the user or data class.

DFSMSHsm considers explicit expiration dates to be INVALID when:

- the day portion of the date is not between 1 and 366, nor is 999.

Note: Expiration dates of 99.365, 99.366, and 99.999 continue to be honored as "never expire" when explicitly specified.

- the current definition of the assigned SMS Management Class specifies a RETENTION LIMIT value of zero to disallow the use of explicit expiration dates prior to a Management Class change.

EXPIRE AFTER DAYS NON-USAGE specifies how many days must elapse from the last access of the data set until the data set becomes eligible for expiration. EXPIRE AFTER DATE/DAYS specifies an absolute date or period after the data set has been created for a data set to become eligible for expiration. The default for both attributes is NOLIMIT.

Both attributes can be used together. When both attributes have a value other than NOLIMIT, both attributes must be satisfied before the data set is expired. When both attributes are NOLIMIT, the data set does not expire. When one of the attributes is NOLIMIT, that attribute is not used to expire the data set; the expiration of the data set is based on the attribute that is not NOLIMIT.

It is possible to have a management class request both automatic backup processing—by using the AUTO BACKUP setting of Y— and expiration processing—by using the EXPIRE AFTER.... as numeric values. In this case, DFSMSHsm determines if a backup copy has been made before expiring the data set. The data set will be deleted when any of the following conditions are true:

- The data set change flag is off in the VTOC entry *and* the last backup date in the catalog is nonzero.

Note: DFSMSHsm does not verify the existence of the backup version.

- The data set change flag is off in the VTOC entry *and* the last backup date in the catalog is zero *and* the management class does not request automatic backup for this data set.
- The data set change flag is on in the VTOC entry *and* the management class does not request automatic backup for this data set.

Rule: ADMIN OR USER COMMAND BACKUP must not be NONE and AUTO BACKUP must be Y to request automatic backup.

Specifying partial release

The PARTIAL RELEASE attribute specifies whether DFSMSHsm space management processing releases unused allocated space in physical sequential, partitioned, and extended format VSAM data sets. The default for PARTIAL RELEASE is NO. The other options are YES and CONDITIONAL. CONDITIONAL means to release the space only if there is a nonzero secondary allocation quantity.

Note:

1. The CONDITIONAL option applies only to physical sequential and partitioned data sets. If specified for an extended format VSAM data set, it will be processed as if YES were specified.
2. The PARTIAL RELEASE attribute does not apply to normal migration and recall processing.

Related reading

For more information about migration and recall space allocation, see [“Additional space management during migration and recall when DFSMSHsm is the data mover” on page 87.](#)

Specifying migration attributes

The migration attributes allow you to specify:

- Whether migration is permitted for the data sets.
- When a data set is eligible to migrate from level 0.
- When a data set is eligible to migrate from level 1.
- To what level a data set is eligible to migrate.
- When a data set is eligible to migrate to cloud storage.
- The cloud storage name.
- The migration tier to move a data set to when it is less than a fixed size.
- The migration tier to move a data set to when it is greater than a fixed size.

The COMMAND OR AUTO MIGRATE attribute specifies whether to migrate the data sets automatically, by command, or not at all. If you specify BOTH, both automatic and command migration can occur. If you specify COMMAND, the data sets cannot migrate automatically, but a migrate command for individual data sets can cause the data sets to migrate, regardless of the value specified on the PRIMARY DAYS NON-USAGE attribute. If you specify NONE, the data sets cannot migrate. The default is BOTH. The COMMAND OR AUTO MIGRATE attribute does not affect Class Transition processing.

The PRIMARY DAYS NON-USAGE attribute specifies the minimum number of days a data set must remain unused on an SMS-managed volume before it becomes eligible to migrate. Eligibility to migrate does not imply that the data set will migrate the next time that space management occurs. If space management achieves the necessary free space without migrating all the eligible data sets on the volume, space management processing stops when the necessary free space is available. The default for this attribute is two days.

The SIZE LESS THAN OR EQUAL TO attribute specifies a size threshold. If a data set meets the PRIMARY DAYS NON-USAGE criteria, and the current size of the data set is less than or equal to the threshold, then the specified action is taken.

- The action can be NONE, indicating that the data set should not be migrated. This can be useful if you don't want to migrate data sets below a certain size, even if they have not been referenced.
- The action can be ML1, indicating that the data set should be migrated to the ML1 tier.
- The action can be ML2, indicating that the data set should be migrated directly to the ML2 tier regardless of the values for LEVEL 1 DAYS NON-USAGE.
- The action can be MIG, indicating that the data set should be migrated to ML1, or ML2 according to the value of LEVEL 1 DAYS NON-USAGE.
- The action can be CLOUD, indicating that the data set should be migrated to the cloud storage named in the CLOUD NAME field.
- The action can be TRANS, indicating that the data set should have class transition processing that is performed for it.

The LEVEL 1 DAYS NON-USAGE attribute serves two purposes:

- It specifies the minimum number of days that a data set must be unused before it becomes eligible for migration from level 1 to level 2. This number is the total days not used; it is not days that the data set has actually resided on an ML1 volume. All data sets that are eligible migrate at each level 1 to level 2 migration.
- It indicates what level to migrate to from either level 0 or level 1. A value of 1 to 9999 for this attribute specifies that the data set may migrate first from level 0 to level 1, then from level 1 to the specified ML2 device, either DASD or tape. A value of 0 for this attribute specifies that the data set migrates

directly from level 0 to ML2 tape. (If the data set has already migrated to ML1 before the attribute was set to 0, it migrates to the specified ML2 device, either DASD or tape.) A value of NOLIMIT specifies that the data set migrates to ML1 and is not to migrate to ML2.

If the value of LEVEL 1 DAYS NON-USAGE has been met and the data set still resides on level 0, it will migrate directly to ML2.

The default for this attribute is 60.

The LEVEL 2 DAYS NON-USAGE attribute allows migrations to be directed to cloud storage. It has two values, 0 and NOLIMIT. When NOLIMIT is specified (or defaulted to because blank), the data set will migrate to the ML2 tier based on the value of LEVEL 1 DAYS NON-USAGE. When 0 is specified, the data set will be migrated to the cloud specified in the CLOUDNAME field as long as the PRIMARY DAYS NON-USAGE value is met and the data set still resides on Level 0. The LEVEL 1 DAYS NON-USAGE value does not apply in this case. The default value for LEVEL 2 DAYS NON-USAGE is NOLIMIT.

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

The data set must reside on a volume that is connected to cloud storage. Additionally, the SMS Cloud construct named or CDA provider file must exist and the password for the cloud must be defined to DFSMSHsm.

Specifying class transition attributes

The class transition attributes allow you to specify:

- When a data set is eligible to transition
- Which copy technique to use to move the data set
- What action to take when there is a serialization error

There are three defined class transition attributes that can be used to specify if and when data sets should transition:

- TIME SINCE CREATION indicates the YEARS, MONTHS and/or DAYS that must elapse relative to the date that a data set was created. If left blank (the default value), this indicates that no transitions should occur based on these criteria.
- TIME SINCE LAST USE indicates the YEARS, MONTHS and/or DAYS that must elapse relative to the date that a data set was last referenced or used. If left blank (the default value), this indicates that no transitions should occur based on these criteria.
- PERIODIC indicates a MONTHLY, QUARTERLY or YEARLY cycle for which a data set should transition. If left blank (the default value), this indicates that no transitions should occur based on these criteria.

TRANSITION COPY TECHNIQUE specifies whether a point in time copy technique should be used. The available parameters are:

- STANDARD (STD)
- FAST REPLICATION PREFERRED (FRP)
- FAST REPLICATION REQUIRED (FRR)
- PRESERVE MIRROR PREFERRED (PMP)
- PRESERVE MIRROR REQUIRED (PMR)
- FC XRCPRIMARY (FCX)

These parameters function as follows:

- STANDARD (STD) specifies that only standard I/O should be used to perform the data movement. This is the default.
- FAST REPLICATION PREFERRED (FRP) specifies that fast replication should be used when possible. If fast replication cannot be used, then it is acceptable to use standard I/O to perform the data movement.
- FAST REPLICATION REQUIRED (FRR) specifies that fast replication is required. If fast replication cannot be used, then the transition fails.

- PRESERVE MIRROR PREFERRED (PMP) specifies that a Metro Mirror primary volume is allowed to become a FlashCopy target, and it would be preferable that the Metro Mirror pair does not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, then the FlashCopy operation is still to be attempted. If the intended FlashCopy target volume is *not* a Metro Mirror primary volume, then the rules for FAST REPLICATION PREFERRED (FRP) are followed.
- PRESERVE MIRROR REQUIRED (PMR) specifies that a Metro Mirror or Global Mirror primary volume is allowed to become FlashCopy target volumes. If the FlashCopy target volume is a Metro Mirror primary device, a Preserve Mirror option is required when performing FlashCopy operation. If the target volume is *not* a Metro Mirror or Global Mirror primary volume, then the rules for FAST REPLICATION REQUIRED (FRR) are followed.
- FC XRCPRIMARY (FCX) specifies that an XRC primary volume is allowed to become a FlashCopy target. When the FCX option is set, the PMR option is also specified to DFSMSdss.

SERIALIZATION ERROR EXIT specifies an action to take when the data set cannot be exclusively serialized for the data movement. The available parameters are:

- NONE
- DB2®
- CICS®
- ZFS
- EXIT

These parameters function as follows:

- NONE specifies that the transition should fail with no additional action. This is the default.
- Db2 specifies that Db2 should be invoked to close the data set. If there are no Db2 transactions in progress and the data set is successfully closed and unallocated, then the data set will be exclusively serialized. If exclusive access is obtained, then the data set will be transitioned. After the data set has moved, Db2 is reinvoked to allocate and open the data set. If the serialization cannot be obtained or there are active Db2 transactions then the transition will fail.
- CICS specifies that CICS should be invoked to make the data set unavailable for use by CICS and to close all files open to the data set. If these steps are successful, then the data set will be exclusively serialized and then transitioned. After the data set has moved, CICS is reinvoked to enable the CICS files to use the data set and make available the data set to be used by CICS. If the serialization cannot be obtained a second time, then the transition will fail.
- ZFS specifies that zFS should be invoked to unmount the data set. If the data set is successfully unmounted, then the data set will be exclusively serialized. The unmount will fail if the file system is currently accessing the data set. If exclusive access is obtained, then the data set will be transitioned. After the data set has moved, zFS is reinvoked to mount the data set.
- EXIT specifies that a user exit should be invoked to unserialize the data set. The exit will be invoked twice: initially to unserialize the data set, and a second time after the transition in order to reserialize the data set. The transition will be performed if the data set can be exclusively serialized after the user exit has been initially invoked.

Note:

1. When specifying a value other than NONE, ensure that only the appropriate types of data sets are assigned to the management class. For example, if Db2 is specified, then only Db2 objects should be assigned to this management class.
2. See [z/OS DFSMSdss Storage Administration](#) for more detailed information on the behavior of this option.
3. When a value of Db2 is specified, DFSMSShsm runs with DFSMSdss loaded in the DFSMSShsm address when transitioning/moving the data set.
4. When moving Db2 objects, DFSMSShsm must be authorized to the Db2 STOP and START commands.

Specifying GDG management attributes

The generation data group (GDG) management attributes let you specify migration eligibility for generation data sets based on relative generation number and what to do with rolled-off generation data sets (GDSs).

In establishing the order for migrating data sets from a DFSMSHsm-managed volume, DFSMSHsm can give high priority to generation data sets (members of generation data groups). That is, DFSMSHsm can choose to migrate generation data sets before it migrates non-generation data sets. Generation data sets that are chosen in preference to non-generation data sets are said to receive early migration. You may want to give high priority to generation data sets if old generations are used infrequently. To specify that old generations have high priority for migration and to specify how many generations have standard priority, you can specify the # GDG ELEMENTS ON PRIMARY attribute. When you specify this attribute, all non-rolled-off generations of each GDG that exceed the number you specify are given priority for early migration. If you specify 0 for this attribute, all generations of the GDGs are given priority for early migration.

The ROLLED-OFF GDS ACTION attribute specifies what to do with rolled-off GDSs. The values for this attribute are MIGRATE and EXPIRE. If you specify MIGRATE, DFSMSHsm assigns the highest migration priority to rolled-off GDSs. If you specify EXPIRE, DFSMSHsm deletes any rolled-off GDS that has met the expiration date specified in its data set VTOC entry or that has no expiration date specified in the data set VTOC entry. If a NOSCRATCH GDS rolls off while on a level 0 volume and is migrated, the GDS is expired during migration cleanup.

The GDG management attributes have no defaults. If you do not specify values for these attributes, the other migration and expiration attributes in the management class govern migration and expiration of generation data sets. If a GDS does not meet any of the criteria specified by the GDG management attributes, the other migration and expiration attributes in the management class govern migration and expiration of the data set.

Using the default management class

SMS allows you to define a management class and to specify that this management class is the default management class. Thus, any SMS data set that is not assigned to a management class by the automatic class selection routine uses the attributes specified in the default management class.

If you do not specify a default management class, DFSMSHsm generates its own default by using the default values for all of the management class attributes. Thus, in the absence of a specified default management class, DFSMSHsm uses the following values for any SMS-managed data set for which a management class is not assigned.

Management Class Attribute	Default Value
EXPIRE AFTER DAYS NON-USAGE	NOLIMIT
EXPIRE AFTER DATE/DAYS	NOLIMIT
PARTIAL RELEASE	N
PRIMARY DAYS NON-USAGE	2
SIZE LESS THAN OR EQUAL TO	No default
SIZE GREATER THAN	No default
LEVEL 1 DAYS NON-USAGE	60
LEVEL 2 DAYS NON-USAGE	NOLIMIT
CLOUD NAME	No default
# GDG ELEMENTS ON PRIMARY	No default
ROLLED-OFF GDS ACTION	No default

Management Class Attribute	Default Value
BACKUP FREQUENCY	1
NUMBER OF BACKUP VERS (Data Set Exists)	2
NUMBER OF BACKUP VERS (Data Set Deleted)	1
RETAIN DAYS ONLY BACKUP VER (Data Set Deleted)	60
RETAIN DAYS EXTRA BACKUP VERS	30
ADMIN OR USER COMMAND BACKUP	BOTH
AUTO BACKUP	Y
BACKUP COPY TECHNIQUE	S
CLASS TRANSITION	BLANK
TRANSITION COPY TECHNIQUE	STD
SERIALIZATION ERROR EXIT	NONE

Note: DFSMSHsm creates a management class called HSMDEFMC.

Management class examples

The following examples show how to use the various management classes.

Standard management class attributes

For data sets that use the standard management class (STANDMC), the following actions occur:

- The data sets expire if they are not used in 120 days and 360 days have passed since they were created.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.
- Data sets that are not used for 10 days are migrated to ML1.
- Data sets that are not used for 60 days are migrated to ML2.

These are the management class attributes for STANDMC:

```
NAME ==>  STANDMC
          EXPIRE AFTER DAYS NON-USAGE ==> 120
          EXPIRE AFTER DATE/DAYS       ==> 360
          PARTIAL RELEASE               ==> Y
          PRIMARY DAYS NON-USAGE        ==> 10
          LEVEL 1 DAYS NON-USAGE        ==> 60
          COMMAND OR AUTO MIGRATE       ==> BOTH
          SIZE LESS THAN OR EQUAL TO
          SIZEGT OPTIONS                ==> NONE
          SIZE GREATER THAN OR EQUAL TO
          SIZELTE OPTIONS               ==> NONE
          LEVEL 2 DAYS NON-USAGE        ==> NOLIMIT
          CLOUD                        ==>
```

GDS management class attributes

Data sets in generation data groups use the GDS management class (GDSMC) for the following results:

- Data sets expire if they are not used in seven days and 90 days have passed since they were created.
- Unused allocated space is not released from any data sets.
- Data sets that are not used in five days migrate to ML1.
- Data sets never migrate to migration level 2.

Space management of SMS-managed storage

- Non-rolled-off data sets that exceed the last two generations receive priority for command and event-driven migration.
- Rolled-off generation data sets are expired.

These are the management class attributes for GDSMC:

```
NAME ==> GDSMC
      EXPIRE AFTER DAYS NON-USAGE ==> 7
      EXPIRE AFTER DATE/DAYS      ==> 90
      PARTIAL RELEASE              ==> N
      PRIMARY DAYS NON-USAGE      ==> 5
      LEVEL 1 DAYS NON-USAGE      ==> NOLIMIT
      COMMAND OR AUTO MIGRATE     ==> BOTH
      # GDG ELEMENTS ON PRIMARY   ==> 2
      ROLLED-OFF GDS ACTION       ==> EXPIRE
SIZE LESS THAN OR EQUAL TO
      SIZEGT OPTIONS              ==> NONE
      SIZE GREATER THAN OR EQUAL TO
      SIZELTE OPTIONS            ==> NONE
      LEVEL 2 DAYS NON-USAGE     ==> NOLIMIT
      CLOUD                      ==>
```

Large management class attributes

In the example system, large data sets are not to migrate to ML1 DASD. Instead, those data sets should migrate directly to ML2 tape. Data sets that use this management class (LARGEMC) receive the following processing:

- The data sets expire if they are not used in 90 days, and 180 days have passed since they were created.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.
- Data sets that are not used for 15 days migrate to ML2 tape.

These are the management class attributes for LARGEMC:

```
NAME ==> LARGEMC
      EXPIRE AFTER DAYS NON-USAGE ==> 90
      EXPIRE AFTER DATE/DAYS      ==> 180
      PARTIAL RELEASE              ==> Y
      PRIMARY DAYS NON-USAGE      ==> 15
      LEVEL 1 DAYS NON-USAGE      ==> 0
      COMMAND OR AUTO MIGRATE     ==> BOTH
SIZE LESS THAN OR EQUAL TO
      SIZEGT OPTIONS              ==> NONE
      SIZE GREATER THAN OR EQUAL TO
      SIZELTE OPTIONS            ==> NONE
      LEVEL 2 DAYS NON-USAGE     ==> NOLIMIT
      CLOUD                      ==>
```

No-migration management class attributes

For data sets that have value as long as they are used but lose value quickly if they are not used, the no-migration (NOMIGMC) management class has been defined. Data sets that use this management class receive the following processing:

- The data sets never migrate.
- The data sets expire if they are not used in 20 days.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.

These are the attributes for NOMIGMC:


```

NAME ====> NOMIGMC
          EXPIRE AFTER DAYS NON-USAGE ====> 20
          EXPIRE AFTER DATE/DAYS       ====> NOLIMIT
          PARTIAL RELEASE                ====> Y
          PRIMARY DAYS NON-USAGE         ====>
          LEVEL 1 DAYS NON-USAGE         ====> NOLIMIT
          COMMAND OR AUTO MIGRATE        ====> NONE
SIZE LESS THAN OR EQUAL TO
          SIZEGT OPTIONS                  ====> NONE
          SIZE GREATER THAN OR EQUAL TO  ====>
          SIZELTE OPTIONS                 ====> NONE
          LEVEL 2 DAYS NON-USAGE         ====> NOLIMIT
          CLOUD                          ====>

```

The EXPIRE AFTER DAYS NON-USAGE attribute has been set to 20 days and the EXPIRE AFTER DATE/DAYS attribute has been set to NOLIMIT. The data sets will expire after 20 days of nonusage. If both attributes are set to NOLIMIT, the data sets will never expire.

Transition data management class attributes

For data sets that need to remain online for an extended period of time, but need to be stored on a less expensive storage media after a certain time period, the standard transition (STDTRANS) management class can be defined. Data sets that use this management class receive the following processing:

- The data sets transition after 15 days of inactivity.
- Data sets do not transition if serialization cannot be obtained.
- Standard data movement is used.
- The data sets migrate to ML2 after 366 days of inactivity.
- The data sets expire after 7 years.

These are the attributes for STDTRANS:

```

NAME ====> STDTRANS
          TIME SINCE LAST USE            ====> 15
          SERIALIZATION ERROR EXIT       ====> NONE
          TRANSITION COPY TECHNIQUE      ====> STD
          PRIMARY DAYS NON-USAGE         ====> 366
          LEVEL 1 DAYS NON-USAGE         ====> 0
          COMMAND OR AUTO MIGRATE        ====> BOTH
          EXPIRE AFTER DATE/DAYS         ====> 2556
SIZE LESS THAN OR EQUAL TO
          SIZEGT OPTIONS                  ====> NONE
          SIZE GREATER THAN OR EQUAL TO  ====>
          SIZELTE OPTIONS                 ====> NONE
          LEVEL 2 DAYS NON-USAGE         ====> NOLIMIT
          CLOUD                          ====>

```

Transition Db2 data management class attributes

For Db2 data sets that need to remain online, but need to be stored on a less expensive storage media after a certain time period, the Db2 transition (DB2TRANS) management class can be defined. DB2 data sets that use this management class receive the following processing:

- The data sets never migrate.
- The data sets transition after 90 days from creation.
- The data sets are expected to be open during primary space management processing, so the Db2 exit will be invoked.
- To minimize the duration of the transition, fast replication will be preferred.
- The data sets expire after 7 years.

These are the attributes for DB2TRANS:


```

NAME ==> DB2TRANS
        TIME SINCE CREATION          ==> 90
        SERIALIZATION ERROR EXIT     ==> DB2
        TRANSITION COPY TECHNIQUE    ==> FRP
        PRIMARY DAYS NON-USAGE       ==> BLANK
        LEVEL 1 DAYS NON-USAGE       ==> NOLIMIT
        COMMAND OR AUTO MIGRATE      ==> NONE
        EXPIRE AFTER DATE/DAYS       ==> 2556
SIZE LESS THAN OR EQUAL TO
        SIZEGT OPTIONS               ==> NONE
        SIZE GREATER THAN OR EQUAL TO
        SIZELTE OPTIONS              ==> NONE
        LEVEL 2 DAYS NON-USAGE       ==> NOLIMIT
        CLOUD                        ==>

```

Enhanced management class 1 attributes

Small data sets should not be migrated, but as they grow in size, the location can be ML1 or ML2. Data sets that use this management class receive the following processing:

- The data sets expire if they are not used in 120 days and 360 days have passed since they were created.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.
- Data sets that are not used for 10 days are eligible to migrate. If the size of the data set is less or equal to 20 tracks, it is not migrated. If the size is between 20 and 4000 tracks, it is migrated to ML1. If the size is larger than 4000 tracks, it is migrated directly to ML2.
- Data sets that are not used for 60 days are eligible for migration. If the size is less than or equal to 20 tracks, it is not migrated. If the size is greater than 20 tracks, it is migrated to ML2 if it is on level 0 DASD or ML1.

These are the attributes of ENHMC1:

```

EXPIRE AFTER DAYS NON-USAGE ==> 120
EXPIRE AFTER DATE/DAYS     ==> 360
PARTIAL RELEASE            ==> Y
PRIMARY DAYS NON-USAGE     ==> 10
SIZE LESS THAN OR EQUAL    ==> 20
        SIZELTE ACTION      ==> NONE
        SIZE GREATER THAN    ==> 4000
        SIZEGT ACTION        ==> ML2
LEVEL 1 DAYS NON-USAGE     ==> 60
LEVEL 2 DAYS NON-USAGE     ==> NOLIMIT
CLOUD NAME                 ==>
COMMAND OR AUTO MIGRATE    ==> BOTH

```

Enhanced management class 2 attributes

Small data sets in this management class are never migrated. Medium size data sets go to ML1 or ML2. Large data sets are migrated to the cloud. Data sets in this management class receive the following processing:

- The data sets expire if they are not used in 120 days and 360 days have passed since they were created.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.
- Data sets that are not used for 10 days are eligible to migrate. If the size of the data set is less or equal to 20 tracks, it is not migrated. If the size is between 20 and 4000 tracks, it is migrated to ML1. If the size is larger than 6000 tracks, it is migrated to the cloud storage defined in the LOCALCLOUD SMS cloud construct.
- Data sets that are not used for 60 days are eligible for migration. If the size is less than or equal to 20 tracks, it is not migrated. If the size is between 20 and 6000 tracks, it is migrated to ML2 if it is on level 0 DASD or ML1.

These are the attributes of ENHMC2:

```

EXPIRE AFTER DAYS NON-USAGE ===> 120
EXPIRE AFTER DATE/DAYS      ===> 360
PARTIAL RELEASE              ===> Y
PRIMARY DAYS NON-USAGE      ===> 10
SIZE LESS THAN OR EQUAL     ===> 20
  SIZELTE ACTION             ===> NONE
SIZE GREATER THAN           ===> 6000
  SIZEGT ACTION              ===> CLOUD
LEVEL 1 DAYS NON-USAGE      ===> 60
LEVEL 2 DAYS NON-USAGE      ===> 0
CLOUD NAME                   ===> LOCALCLOUD
COMMAND OR AUTO MIGRATE     ===> BOTH

```

MOST2CLOUD management class attributes

Small data sets are not migrated. All other data sets migrate to cloud storage. Data sets that use this management class receive the following processing:

- The data sets expire if they are not used in 120 days and 360 days have passed since they were created.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.
- Data sets that are not used for 10 days are eligible to migrate. If the size of the data set is less or equal to 150 tracks, it is not migrated. If the size is greater than 150 tracks, it is migrated to the cloud storage defined in the LOCALCLOUD SMS construct.

These are the attributes of MOST2CLOUD

```

EXPIRE AFTER DAYS NON-USAGE ===> 120
EXPIRE AFTER DATE/DAYS      ===> 360
PARTIAL RELEASE              ===> Y
PRIMARY DAYS NON-USAGE      ===> 10
SIZE LESS THAN OR EQUAL     ===> 150
  SIZELTE ACTION             ===> NONE
SIZE GREATER THAN           ===>
  SIZEGT ACTION              ===>
LEVEL 1 DAYS NON-USAGE      ===> 0
LEVEL 2 DAYS NON-USAGE      ===> 0
CLOUD NAME                   ===> LOCALCLOUD
COMMAND OR AUTO MIGRATE     ===> BOTH

```

BIG2 Cloud management class

Very big data sets migrate to cloud storage. Other sizes migrate to ML1 or ML2 normally. Data sets with this management class receive the following processing:

- The data sets expire if they are not used in 120 days and 360 days have passed since they were created.
- Unused allocated space is released from physical sequential, partitioned, and extended format VSAM data sets.
- Data sets that are not used for 10 days are eligible to migrate. If the size of the data set is less or equal to 7500 tracks, it is migrated to ML1. If the size of the data set is greater than 7500 tracks, it is migrated to the cloud storage described by the LOCALCLOUD SMS construct.
- Data sets that are not used for 20 days are eligible for migration. If the size is less than 7500 tracks, it is migrated to ML2 if it is on level 0 DASD or ML1.

These are the attributes of BIG2CLOUD


```
EXPIRE AFTER DAYS NON-USAGE ===> 120
EXPIRE AFTER DATE/DAYS      ===> 360
PARTIAL RELEASE              ===> Y
PRIMARY DAYS NON-USAGE      ===> 10
SIZE LESS THAN OR EQUAL     ===>
SIZELT ACTION                ===>
SIZE GREATER THAN           ===> 7500
SIZEGT ACTION                ===> CLOUD
LEVEL 1 DAYS NON-USAGE      ===> 20
LEVEL 2 DAYS NON-USAGE      ===> 0
CLOUD NAME                   ===> LOCALCLOUD
COMMAND OR AUTO MIGRATE     ===> BOTH
```

Specifying when automatic primary space management can process

The PRIMARYSPMGMTCYCLE parameter of the DEFINE command defines the cycle of automatic primary space management.

Example: To define an automatic primary space management cycle, use the following command:

```
DEFINE PRIMARYSPMGMTCYCLE(YNNNYNN) -
CYCLESTARTDATE(1998/09/12)
```

Although this example shows automatic primary space management running two days a week (Saturday and Wednesday), you might want to run it more often.

Specifying when automatic primary space management starts

The PRIMARYSPMGMTSTART parameter of the SETSYS command controls the processing window of automatic primary space management.

Example:

To define a processing window of 2:00 AM to 4:00 AM, use the following command:

```
SETSYS PRIMARYSPMGMTSTART(0200 0400)
```

Allowing primary space management to start at midnight

Many primary space management windows start after midnight to allow data sets to be processed as soon as their non-referenced day count reaches the threshold in the management class. The MIGRATEPSMEARLY parameter on the SETSYS command can be used to allow DFSMSHsm process a data set early. If the primary space management window begins after 6 PM, one will be added to the data set age up until the current time is midnight. Afterward, the current age is used in the calculation.

If the cutoff time of 6 PM needs to be changed, a patch (MIGCB.+98 X'hhmm') can be used to modify the cutoff time. For more information, see [Earlier checking for primary space management \(PSM\) in z/OS DFSMSHsm Implementation and Customization Guide](#).

Note: Backup of the control data sets can interfere with starting automatic primary space management.

Related reading

For more information about backing up control data sets, see [“Phase 1: Backing up the control data sets” on page 199](#).

Specifying when automatic secondary space management can process

The SECONDARYSPMGMTCYCLE parameter of the DEFINE command defines the cycle of automatic secondary space management.

Example: To define the days for secondary space management to run and the start date, use the following command:


```
DEFINE SECONDARYSPMGMTCYCLE(YNNNNYNN) -  
CYCLESTARTDATE(1998/09/12)
```

Specifying when automatic secondary space management starts

The SECONDARYSPMGMTSTART parameter of the SETSYS command controls the processing window of automatic secondary space management.

You can run automatic secondary space management any time of the day except when automatic primary space management is running (if SDSPs are used).

Example: To define a processing window of 12:30 a.m. to 2:00 a.m., use the following command:

```
SETSYS SECONDARYSPMGMTSTART(0030 0200)
```

Although secondary space management ending time is 2:00 a.m., any data set deletion management work elements (MWEs) that secondary space management built up to 2:00 a.m. would still continue to be processed. If a large number of data sets needed deletion, MWE processing could continue for a significant time after secondary space management stops building data set deletion MWEs.

Note:

1. You must specify SECONDARYSPMGMTSTART if you want it to run; it will not run by default.
2. Data set deletions generated by SSM are processed after any user recalls or user generated deletions.
3. Backup of the CDSs can interfere with the starting of automatic secondary space management. See [“Phase 1: Backing up the control data sets” on page 199](#) for discussion of this consideration.

Specifying migration of data sets to overflow volumes

You can ADDVOL migration volumes (ML1) to be OVERFLOW volumes and they will be considered for backup or migration of data sets when either one of the following are true:

- The data you are migrating or backing up is larger than a given size, as specified on the SETSYS ML1OVERFLOW(DATASETSIZE(*dssize*)) command
- DFSMSHsm cannot allocate enough space on a NOOVERFLOW volume by selecting either the least used volume or the volume with the most free space.

Related reading

- See [“OVERFLOW and NOOVERFLOW: Specifying the usage of a level 1 volume for data set migration and backup versions” on page 707 in Chapter 26, “ADDVOL command: Adding or changing DFSMSHsm volumes,” on page 697.](#)
- See [“ML1OVERFLOW: Manage ML1 OVERFLOW volumes” on page 1160 in Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.](#)

Specifying how to migrate data sets to migration level 2

Parameters of the SETSYS command provide the specification for how to migrate data sets to ML2.

If you are using the duplex tape option for your ML2 tapes, see [“Duplex tape option” on page 436](#) for information specific to the duplexing environment.

See the following topics:

- [“Using the SETSYS TAPEMIGRATION parameter” on page 47](#)
- [“Specifying when to migrate from ML1 to ML2” on page 48](#)

Using the SETSYS TAPEMIGRATION parameter

You can specify that DFSMSHsm use only tape volumes for ML2 volumes.

Example: You can add the following SETSYS TAPEMIGRATION command to the ARCCMDxx member in each DFSMSHsm host:

```
SETSYS TAPEMIGRATION(ML2TAPE(TAPE(ACL)) RECONNECT(ALL))
```

The TAPEMIGRATION parameter of the SETSYS command specifies how migration to tape occurs. The ML2TAPE subparameter specifies that migration occurs from ML1 to ML2 tape. The TAPE(ACL) option of the ML2TAPE subparameter limits the kind of tape unit that is selected to the kind of unit that is specified. This example specifies that only units that are identified by the esoteric name of ACL can be used as output tape units for ML2 volumes.

The RECONNECT parameter of the SETSYS TAPEMIGRATION command controls whether DFSMSHsm attempts to reconnect a data set that is recalled from a single ML2 tape to that migration copy when the data set is next migrated.

The ALL keyword of the fast subsequent migration RECONNECT parameter specifies that you want all eligible data sets to reconnect to their ML2 tapes. You want this to occur even if the data sets would otherwise be migrated to ML1.

Note: Only data sets that are directed to ML1 by the FORCML1 parameter (on the ARCHMIG macro) are excluded from reconnection.

Alternative subparameters that have meaning to SMS-managed data sets are NONE and NONE(ROUTETOTAPE).

The NONE subparameter specifies that ML2 volumes are DASD. NONE(ROUTETOTAPE(*unittype*)) specifies that although ML2 volumes are DASD, any management class with a value of 0 in the LEVEL 1 DAYS NON-USAGE attribute causes data sets to migrate directly to tape of the kind specified by *unittype*. The *unittype* subparameter restricts the kind of unit for the ML2 tape unit.

If you specify TAPEMIGRATION(NONE) or TAPEMIGRATION(NONE(ROUTETOTAPE(*unittype*))), you can define key ranges to be stored on different DASD ML2 volumes. The MIGRATIONLEVEL2 parameter of the DEFINE command specifies the key ranges for DASD ML2 volumes.

Guideline: If you do not intend to use tape, do not specify TAPEMIGRATION. If you do, an offline control data set is required.

With SMS, both DASD and tape can be used together by specifying the NONE or (NONE(ROUTETOTAPE)) subparameters of the TAPEMIGRATION parameter and specifying LEVEL 1 DAYS NON-USAGE = 0 in some management classes. Under those specifications, level 1 to level 2 migration is to DASD but migration from DFSMSHsm-managed volumes is to tape for data sets in the appropriate management classes.

When migrating SMS volumes and both PRIMARY DAYS NON-USAGE and LEVEL 1 DAYS NON-USAGE have been satisfied in an ML2 tape environment, the SMS data sets are moved directly to ML2 tape. In an ML2 DASD environment, the target is ML1 DASD unless LEVEL 1 DAYS NON-USAGE is 0.

Related reading

- For more information about enabling fast subsequent migration, see [Enabling fast subsequent migration](#).
- For more information, see [Chapter 40, “DEFINE command: Defining control structures for use by DFSMSHsm,” on page 809](#).

Specifying when to migrate from ML1 to ML2

If you back up or migrate data sets to ML1 OVERFLOW volumes (see “Specifying migration of data sets to overflow volumes” on page 47), you can specify the percentage of occupied space that must be in the ML1 OVERFLOW volume pool before the migration of data sets to ML2 volumes occurs during automatic secondary space management.

As shown in the example below, the default setting for the ML1 OVERFLOW volume pool threshold is 80%. This requires the occupied space of the entire overflow volume pool reach 80% full before any data is moved off of OVERFLOW volumes to ML2 volumes.

```
SETSYS ML1OVERFLOW(THRESHOLD(80))
```

In this case, the percentage of occupied space for each ML1 OVERFLOW volume will be calculated and then DFSMSHsm will compare the average value with the specified threshold (80%). If the average of the ML1 OVERFLOW volume pool has an occupancy that is equal to or greater than the threshold, then secondary space management will be started and move data from ML1 to ML2. See [“ML1OVERFLOW: Manage ML1 OVERFLOW volumes”](#) on page 1160.

Specifying how to do extent reduction

The MAXEXTENTS parameter of the SETSYS command specifies how many extents a non-VSAM single-volume data set can have before it is reduced to fewer extents.

Example: You can add the following command to the ARCCMDxx member in each DFSMSHsm host:

```
SETSYS MAXEXTENTS(11)
```

Note:

1. The daily statistics report output shows the number of extent reductions done each day. You can use this information to tune the SETSYS MAXEXTENTS parameter.
2. DFSMSHsm does not perform extent reduction for extended format data sets.
3. Using a low MAXEXTENTS value might cause extent reduction to occur more frequently than you want. If you specify a MAXEXTENTS value, try using a value of 5 or more.

Specifying how to scratch expired data sets

During automatic primary space management, the example scratches expired data sets that have valid explicit expiration dates in the data set's VTOC entry. The EXPIREDDATASETS parameter of the SETSYS command specifies the SCRATCH subparameter indicating for DFSMSHsm to scratch any data set that has passed the expiration date. See the explanation of VALID EXPIRATION DATES under [“Specifying expiration attributes”](#) on page 36.

Example: You can add the following command to the ARCCMDxx member in each DFSMSHsm host:

```
SETSYS EXPIREDDATASETS(SCRATCH)
```

The expiration attributes in the management class control the expiration of non-VSAM data sets that do not have an expiration date in their data set VTOC entry or VSAM data sets that do not have an expiration date in the base cluster catalog entry. The management class does not control data sets with an expiration date in the data set VTOC or catalog entry. These data sets require the EXPIREDDATASETS(SCRATCH) parameter of the SETSYS command.

Note: For multivolume non-VSAM data sets, DFSMSHsm looks at the VTOC of the first volume.

An alternative subparameter for the EXPIREDDATASETS parameter is NOSCRATCH. NOSCRATCH is the default value and prevents DFSMSHsm from scratching data sets that have passed the expiration date indicated in the data set's VTOC entry during space management processing.

Specifying the maximum automatic volume space management tasks and subtasks

DFSMSHsm allows up to 15 automatic volume space management (primary space management, interval migration, and on-demand migration) tasks to run concurrently in each DFSMSHsm host during space management. Use the MAXMIGRATIONTASKS parameter of the SETSYS command to control the

maximum number of tasks that you want to run concurrently. If you do not specify this parameter on any SETSYS command, the default value for the maximum number of concurrent automatic volume space management tasks is 2.

Example: You can add the following command to the ARCCMDxx parmlib member of SYS1.PARMLIB of each DFSMSHsm hosts to handle automatic volume space management:

```
SETSYS MAXMIGRATIONTASKS(5)
```

If migration is to tape, each concurrent primary space management task that migrates a data set directly to tape requires a tape unit. If you are using the duplex tape option, two tape units per task are required. Tape units are allocated when a task encounters the first data set that is being migrated to tape, and the task does not release the tape units until it has completed processing all volumes. If migration is to SDSP data sets, each concurrent primary space management task being performed needs an SDSP data set. An SDSP data set is allocated when a task encounters the first data set that is being migrated to an SDSP. The task does not release the SDSP until the entire volume has been processed unless it remains unused for 30 seconds or the functions RECALL or ABACKUP need the SDSP data set.

Note: The actual number of migration tasks can be equal to the number of tasks specified on the MAXMIGRATIONTASKS or the MAXINTERVALTASKS and up to twice the specified number of tasks because of DFSMSHsm's preprocessing enhancement. This is reflected in a number of ARC0161I messages issued in response to the QUERY ACTIVE or QUERY ACTIVE(TCB) commands. MIGRATING, MIGRATION PREPROCESSING, FREEVOL PROCESSING, and DBA/DBU PROCESSING differentiate the type of space management DFSMSHsm is performing and these are displayed in the ARC0161I message.

Migration tasks perform all space management tasks, including class transitions. Class transitions have a higher priority than migrations and will be processed first during the data movement phase. If a data set is eligible for both a class transition and migration, then it will be migrated.

DFSMSHsm can also run multiple subtasks concurrently under each migration task for primary space management, on-demand migration, and interval migration on level 0 volumes that migrate data sets to ML1 or ML2 volumes. This can improve throughput by reducing the elapsed time for any migration function involving multiple data sets. Use the MIGRATIONSUBTASKS parameter of the SETSYS command to enable this function.

If you wish to specify additional subtasks under each migration task, you can do this dynamically with the ADDITIONALSUBTASKS(*nn*) subparameter. Up to 10 additional subtasks are allowed. Specifying a value of zero will remove all previously-defined additional subtasks.

Example: You can add the following command to the ARCCMDxx parmlib member of SYS1.PARMLIB of each DFSMSHsm host to enable migration subtasks, and also specify two additional subtasks for each migration task:

```
SETSYS MIGRATIONSUBTASKS(YES ADDITIONALSUBTASKS(2))
```

Note that specifying an ADDITIONALSUBTASKS(*nn*) value greater than 0 should be done with caution, as the additional concurrent subtasks may increase the amount of below the line storage used by DFSMSHsm.

Note, also, that when MIGRATIONSUBTASKS(YES) is specified, concurrent cell pool activity may increase in cell pool 5. It is therefore recommended to review the value specified by the CELLS parameter in the DFHSM startup procedure for CELL POOL 5. See "Specifying the size of cell pools" and "Adjusting the size of cell pools" in the [z/OS DFSMSHsm Implementation and Customization Guide](#) for more information.

Specifying the maximum automatic secondary space management tasks

DFSMSHsm allows up to 30 secondary space management tasks to run concurrently within a single DFSMSHsm host during automatic secondary space management. Use the MAXSSMTASKS parameter of the SETSYS command to control the maximum number of automatic secondary space management tasks that you want to run concurrently. Up to 15 tasks are available for cleanup functions and up to 15

tasks are available for tape movement functions. Secondary space management cleanup consists of the following functions:

- Expiration of migrated data sets.
- Deletion of MCDS records that are no longer needed.
- Deletion of SDSP records that are no longer needed.

Tape movement functions consists of moving ML1 data to ML2 tape, which is also referred to as *ML1-to-ML2 data movement*.

The MAXSSMTASKS(EMPTYCONTAINERDELETION(n)) parameter on the SETSYS command allows management of the processing of empty containers. A value of 0 or 1 can be specified.

- DFSMSHsm does not examine the known clouds for empty containers if 0 is specified.
- If 1 is specified, DFSMSHsm looks for empty containers to delete. Only empty containers created using the current HSMplex name will be found. This function will not find empty containers that were created using an HSMplex name other than the current HSMplex name.

This SETSYS MAXSSMTASKS parameter is dynamic. If you change the number of tape movement tasks or cleanup tasks while secondary space management is active, the new number of tasks are immediately effective.

If you do not specify this parameter on any SETSYS command, the default value for the maximum number of concurrent cleanup tasks is 2 and the default value for the maximum number of concurrent tape movement tasks is 1. To aid in the processing of the deletions initiated by secondary space management, the default for the number of recall tasks is 15. You can use the SETSYS MAXRECALLTASKS command to change this default.

Example: You can add the following command to the ARCCMDxx member in each DFSMSHsm host to handle automatic secondary space management:

```
SETSYS EMPTYCONTAINERDELETION(1) MAXSSMTASKS(CLEANUP(2) TAPEMOVEMENT(1))
```

Note: Each task that you specify for tape movement requires a tape unit. If you are using the duplex tape option, two tape units per task are required.

The MAXSSMTASKS(EMPTYCONTAINERDELETION(n)) parameter on the SETSYS command allows management of the processing of empty containers. A value of 0 or 1 can be specified.

- DFSMSHsm does not examine the known clouds for empty containers if 0 is specified.
- If 1 is specified, DFSMSHsm looks for empty containers to delete.

Specifying the maximum interval migration or on-demand migration tasks

DFSMSHsm allows up to 15 interval migration or on-demand migration tasks to run concurrently in each DFSMSHsm host during automatic interval or on-demand space management. Use the MAXINTERVALTASKS parameter of the SETSYS command to control the maximum number of interval migration or on-demand migration tasks that you want to run concurrently. If you do not specify MAXINTERVALTASKS on any SETSYS command, it defaults to the value of MAXMIGRATIONTASKS.

Note: MAXEVENTDRIVENMIGTASKS or the MAXEDMT alias can be used instead of the MAXINTERVALTASKS parameter.

If migration is to tape, a tape unit is required for each concurrent interval migration or on-demand migration task that migrates a data set directly to tape. If using the duplex tape option, two tape units per task are required. Tape units are allocated when a task encounters the first data set being migrated to tape, and the task does not release the tape units until it has completed processing all volumes. If migration is to SDSP data sets, an SDSP data set is needed for each concurrent interval migration or on-demand migration task being performed. An SDSP data set is allocated when a task encounters the first data set that is being migrated to an SDSP. The task does not release the SDSP until the entire volume has been processed unless it remains unused for 30 seconds or the functions RECALL or ABACKUP need the SDSP data set.

Note: The actual number of migration tasks can be equal to the number of tasks specified on the MAXMIGRATIONTASKS or the MAXINTERVALTASKS and up to twice the specified number of tasks because of DFSMSHsm's preprocessing enhancement. This is reflected in a number of ARC0161I messages issued in response to the QUERY ACTIVE or QUERY ACTIVE(TCB) commands. MIGRATING, MIGRATION PREPROCESSING, FREEVOL PROCESSING, and DBA/DBU PROCESSING differentiate the type of space management DFSMSHsm is performing and these are displayed in the ARC0161I message.

Migration tasks perform all space management tasks, including class transitions. Class transitions have a higher priority than migrations and will be processed first during the data movement phase. If a data set is eligible for both a class transition and migration, then it will be migrated.

Specifying an on-demand migration number of concurrently queued volumes notification limit

Notification of the number of volumes that are concurrently queued for on-demand migration is issued to the operator's console (highlighted message ARC1901E). The SETSYS ODMNOTIFICATIONLIMIT(*limit*) command is used to specify a custom notification limit. The *limit* parameter is a decimal number 1 to 99999. The default notification limit is 100 volumes.

Example: You can use the following command to change the notification limit to 50 volumes.

```
SETSYS ODMNOTIFICATIONLIMIT(50)
```

When the number of volumes eligible for on-demand migration falls below the notification limit, message ARC1901E is removed from the operator's console.

Specifying recall tasks

DFSMSHsm can perform multitasking operations on a maximum of 15 tasks that recall data sets from DASD or tape and that delete migrated data sets. These are called *recall tasks*.

The MAXRECALLTASKS parameter of the SETSYS command controls the number of recall tasks that can be processed at the same time. The TAPEMAXRECALLTASKS parameter specifies how many of these tasks can be recalling data sets on ML2 tape volumes. The number of tape recall tasks must be less than or equal to the maximum number of recall tasks. If the number of active tape recalls is less than or equal to the number specified by the TAPEMAXRECALLTASKS parameter, DFSMSHsm performs multitask DASD recall operations up to the number specified by the MAXRECALLTASKS parameter.

Example: In the following example, if you specified 15 recall tasks and 6 tape recall tasks and if only 3 tape recall tasks were running, DFSMSHsm would start as many as 12 recall tasks from DASD. You can add this command to the ARCCMDxx member in each DFSMSHsm host.

```
SETSYS MAXRECALLTASKS(15) TAPEMAXRECALLTASKS(6)
```

Specifying small data set packing

The SMALLDATASETPACKING parameter of the SETSYS command controls the sending of data sets to the SDSP data sets.

In specifying small data set packing, you specify the actual physical size of the data sets to be considered as candidates for migration to an SDSP data set. You can specify sizes as large as 800 KB. A candidate data set is eligible for migration to an SDSP data set if its estimated size, after compaction, is less than or equal to 400 KB or to the value you specify, whichever is smaller. The compaction size is estimated using an internal default of 50%. If the data set is not compacted, it is a candidate for migration to an SDSP if its actual size is less than or equal to 400 KB or to the value you specify, whichever is smaller.

Small data set packing causes data sets that are smaller than the size specified to be written as records in special VSAM data sets that you have allocated on the ML1 volumes. Because the minimum allocation unit is a track, writing small data sets as records in a larger data set results in saving space on the ML1 volumes.

The number of SDSP data sets defined must be at least equal to the maximum number of concurrent volume migration tasks that could be executing in your complex.

Example: You can add the following command, the command to be added to the ARCCMDxx member in each DFSMSHsm host:

```
SETSYS SMALLDATASETPACKING(KB(110))
```

Recommendation: Configure recall and ABARS use more SDSP data sets in case some data sets become full during migration.

Related reading

For more information about small data set packing, see [“Managing SDSP data sets” on page 555](#).

Specifying the first qualifier for names of migrated data sets

When DFSMSHsm migrates a data set, it leaves the original data set name in the catalog. For use in its own records, DFSMSHsm generates its own name for the data set. The form of the data set name that DFSMSHsm generates is:

```
prefix.HMIG.Tssmmhh.user1.user2.Xydd
```

If patch-enabled, the following dataset name will be generated:

```
prefix.HMIG.Tcccchh.user1.user2.Xydd
```

For more information, refer to [Changing DFSMSHsm backup and migration generated data set names to reduce contention for similar names and eliminating a possible performance degradation in z/OS DFSMSHsm Implementation and Customization Guide](#).

The MIGRATEPREFIX parameter of the SETSYS command specifies the *prefix* (high-level qualifier) of the generated name. If you do not specify a migrate prefix, DFSMSHsm uses the UID that you specified in the startup procedure.

The migrate prefix is used in the ACS filters that prevent DFSMSHsm-owned data sets from being allocated on SMS-managed volumes. In addition, the use of the prefix allows you to use RACF generic profile protection for all your migration copies.

Example: You can add the following command to the ARCCMDxx member in each DFSMSHsm host:

```
SETSYS MIGRATEPREFIX(HSM)
```

Related reading

- For more information about data set names, see [“Names of migrated data sets” on page 73](#).
- For more information about the migrate prefix, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Specifying if class transitions should be performed during event-driven migration

While most class transitions occur during primary space management, class transitions may also be performed during interval migration and on-demand migration. Class transitions should *not* be performed during the event-driven migration functions if the data is assigned a management class for which a serialization exit is invoked to close a data set for a transition, but it would not be appropriate to take that action during the time frame that the event-driven migration functions process.

Example: In the following example, class transitions will not be performed during event-driven migration. You can add the following command to the ARCCMDxx member in each DFSMSHsm host:

```
SETSYS CLASSTRANSITIONS(EVENTDRIVENMIGRATION(N))
```

Configuring migration cleanup

During automatic secondary space management, DFSMSHsm cleans up the migration control data set (MCDS) and the migration volumes by:

- Deleting expired migrated data sets
- Cleaning up SDSP data sets
- Deleting certain data sets that were not scratched during recall, deletion, or GDS roll-off
- Deleting old daily statistics and volume statistics records
- Deleting old MCDS records

You can control the age of the MCDS records and statistics records to be deleted, the number of cleanup tasks that run concurrently, and the days on which migration cleanup is done.

Data sets that are eligible for ML2 reconnection with the fast subsequent migration function may have their MCD records retained for a longer period of time than for nonreconnectable data sets. This increases the likelihood of a reconnection when the data set next migrates. However, keeping the MCD records longer (while increasing the likelihood of reconnection) also causes growth in the size of the MCDS.

The MIGRATIONCLEANUPDAYS parameter of the SETSYS command follows the form:

```
SETSYS MIGRATIONCLEANUPDAYS(recalldays statdays reconnectdays clouddays)
```

Replace *recalldays* with a decimal number between 1 and 999 that specifies the number of days that DFSMSHsm keeps MCDS data set records for recalled data sets that are not compacted, or are compacted and meet or exceed the current value of SETSYS COMPACTPERCENT. For data sets that are compacted but do not meet the current value of SETSYS COMPACTPERCENT, the MCDS data set records are retained for 90 days.

Replace *statdays* with a decimal number from 1 to 999 that specifies the number of days DFSMSHsm keeps the daily and volume statistics records. The REPORT command uses these records, which the IDCAMS DCOLLECT function also collects.

Replace *reconnectdays* with the number of days that you want to add to the predicted remigration date to control retention of the MCD record for reconnectable data sets. The predicted remigration date is based on the number of days that the data set was unreferenced before the date of its last migration. MCD records for reconnectable data sets are kept until they meet both the *recalldays* and the *reconnectdays* criteria.

Replace *clouddays* with the number of days that you want to keep old cloud migration copies of a data set around once a new modified version of the same data set has been remigrated. A value of 0 will indicate to DFSMSHsm that no old cloud migration copies will be kept.

You can specify different ages for MCDS records and statistics records.

Example: You can specify the following command to the ARCCMDxx member in each DFSMSHsm host:

```
SETSYS MIGRATIONCLEANUPDAYS(25 10 3)
```

In this example, the length of time to keep MCDS records is a period of 25 days. This is because 20 days is the longest period of residence on a specified DFSMSHsm-managed volume. Therefore, any data set that is recalled has a chance to migrate again before its MCDS record is scratched. Daily statistics and volume statistics records are kept for a period of 10 days. MCDS records for data sets that are marked as eligible for reconnection are kept 3 days beyond their predicted remigration date. The *reconnectdays* value can be adjusted up or down, if the data set is expected to reside on level 0 for a shorter or longer period before its next migration as compared to its most recent migration.

Note: The `MIGRATIONCLEANUPDAYS(recalldays)` parameter of the `SETSYS` command is one of the criteria that are used for deleting the MCD record of a recalled data set. However, if the data set was compacted during migration and is not eligible for compaction again, the MCD record is kept for at least 90 days. The 90 day retention occurs regardless of the value specified by *recalldays*. Also the MCD record specifies a count of the number of times that a data set by this name has migrated. This historic information is lost when the MCD record is deleted. If you plan to use this field, use this information to influence the value that you use for *recalldays*.

Related reading

For more information about fast subsequent migration, see [“Reconnecting data sets using fast subsequent migration”](#) on page 94.

Identifying migration level 2 volumes to DFSMSHsm

If you use DASD ML2 volumes or tape volumes that are set aside only for DFSMSHsm use, you must identify the volumes to DFSMSHsm. The `ADDVOL` command does this. An example command that could be issued in at least one DFSMSHsm host if you were using DASD ML2 volumes is:

```
ADDVOL ML2001 UNIT(3390) MIGRATION(MIGRATIONLEVEL2)
```

Specifying duplexing for migration level 2 tapes

The duplex tape option creates two identical tapes in parallel, with one tape designated as the original and one the alternate. To create duplex migration level 2 tapes, specify

```
SETSYS DUPLEX(MIGRATION(Y))
```

If duplexing is not desired for your migration level 2 tapes, specify

```
SETSYS DUPLEX(MIGRATION(N))
```

The DFSMSHsm default is no duplexing.

Specifying the allowed target tiers

The `MIGRATIONAUTOCLOUD` parameter on the `SETSYS` command can tailor the migration tiers that a DFSMSHsm host can target. Migrating data sets to cloud storage can be time-consuming and might possibly cause all planned migrations to not complete during the existing primary space management window. You might want to have some DFSMSHsm hosts only perform migrations to ML1 and ML2, and other hosts only target cloud storage for the migrations.

If `MIGRATIONAUTOCLOUD(CLOUDONLY)` is in effect during the automatic migration (Primary Space Management, Interval Management, On Demand Management), only data sets that meet the criteria to migrate to cloud storage are moved. All other eligible data sets that would be migrated to ML1 or ML2 or transitioned would not be migrated by this host. `MIGRATIONAUTOCLOUD(NOCLOUD)` is the default if a value is not specified.

If `MIGRATIONAUTOCLOUD(ALL)` is in effect, eligible data sets are migrated to the tier indicated by the management class.

If `MIGRATIONAUTOCLOUD(NOCLOUD)` is in effect, only data sets that would target ML1 or ML2, or transition are processed. Any eligible data sets that would target cloud storage are not migrated by this host during the automatic migration. `MIGRATIONAUTOCLOUD(NOCLOUD)` is the default if a value is not specified.

Integrity age for space management

When DFSMSHsm does space management on a volume, it maintains data set integrity by processing only data sets that meet a certain *integrity age*. This integrity age specifies the minimum time that must elapse since the data set was last used before the data set can be processed by space management. The integrity age is affected by the DFSMSHsm host environment and by the method of serialization.

Note: A multiple DFSMSHsm-host environment is determined by DFSMSHsm at startup time by examining whether the index component of the migration control data set resides on a DASD volume that has been SYSGEN'd as SHARED or SHAREDUP. If this is the case, DFSMSHsm performs what is called *multiple host serialization*.

Integrity age is established as follows:

- If serialization is DFHSM DATASET SERIALIZATION and the processing environment is a single DFSMSHsm-host environment, the integrity age is one day.
- If serialization is DFHSM DATASET SERIALIZATION and the processing environment is a multiple DFSMSHsm-host environment, the integrity age is two days.
- If serialization is USER DATASET SERIALIZATION, the integrity age is zero days.
- For temporary SMS-managed data sets in a JES3 environment, the integrity age is two days.

Automatic space management operation examples

It has been previously pointed out how the commands to control DFSMSHsm are structured. In the text that follows, examples are used to describe what DFSMSHsm does in the process of automatic space management. Automatic primary and secondary space management, event-driven migration, and automatic recall are shown.

Automatic space management

Figure 6 on page 57 shows the start of automatic space management. The data set names above the names of each of the groups of DASD devices and below the groups of tape devices are the names of data sets that are stored on the devices. Only the names of data sets that DFSMSHsm processes during this example are shown.

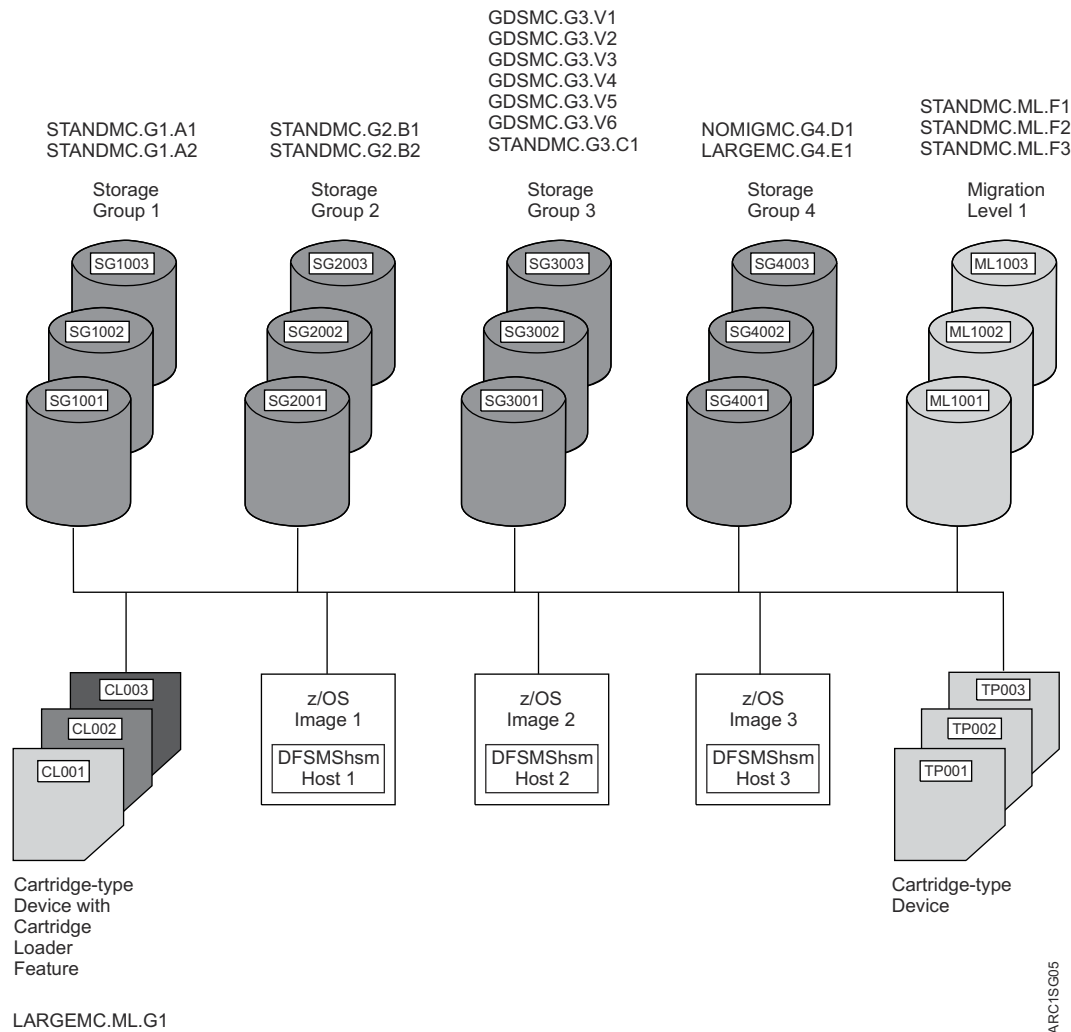


Figure 6. Example System of SMS-managed volumes before start of automatic space management

Note: The names of the management classes have been used as high-level qualifiers for the data set names. This was done to make it easier for you to associate the data sets with the management classes. This is merely a teaching aid. You are not required to use the management class name as any part of the name of the data set.

Table 4 on page 57 shows the characteristics of each of the data sets used in Figure 6 on page 57. The names of the data sets are not precisely as the system might generate them. They have been shortened and modified for convenience in using them for examples.

Table 4. Data set characteristics for automatic space management example				
Data set name	Days since created	Last usage (days)	Size (KB)	GDG Status
GDSMC.G3.V6	1	1	217	Not rolled off
GDSMC.G3.V5	2	2	215	Not rolled off
GDSMC.G3.V4	8	5	210	Not rolled off
GDSMC.G3.V3	10	5	207	Not rolled off
GDSMC.G3.V2	11	5	205	Not rolled off
GDSMC.G3.V1	15	5	200	Rolled off
LARGEMC.G4.E1	50	16	1200	None

Table 4. Data set characteristics for automatic space management example (continued)

Data set name	Days since created	Last usage (days)	Size (KB)	GDG Status
NOMIGMC.G4.D1	25	21	20	None
LARGEMC.ML.G1	183	120	1100	None
STANDMC.G1.A1	361	180	200	None
STANDMC.G1.A2	60	7	200	None
STANDMC.G1.A3	60	7	200	None
STANDMC.G2.B1	54	12	600	None
STANDMC.G2.B2	30	14	100	None
STANDMC.G3.C1	110	61	400	None
STANDMC.ML.F1	181	120	190	None
STANDMC.ML.F2	380	121	3	None
STANDMC.ML.F3	120	62	240	None

Note: 1 KB = 1024 bytes.

Integrating cloud storage into a DFSMSHsm automatic space management environment

Because cloud migration takes longer as opposed to migrating data sets to ML1, ML2, or CLOUD but uses less CPU time, you might want to set up a separate DFSMSHsm host that will perform migrations to cloud storage. When you set up the separate host, keep the following facts in mind:

- To tailor the processing that is done by the DFSMSHsm hosts, use the MIGRATIONAUTOCLOUD, STORAGEGROUPAFFINITY, and MIGRATEPSMEARLY parameters on the SETSYS command.
- HSM1 and HSM2 are hosts that should process migrations to ML1 and ML2.
- HSMC is a host that should process migrations to cloud storage.
- SYS1 is the system that HSM1 runs on. Storage group SG1 is defined to be processed by SYS1. Similarly, SYS2 is the system HSM2 runs on and storage group SG2 is defined to be processed by the SYS2.

Steps to take

To assign the cloud processing to HSMC, use the following specifications.

1. Specify MIGRATIONAUTOCLOUD(CLOUDONLY) to HSMC.
2. Specify MIGRATIONAUTOCLOUD(NOCLOUD) to HSM1 and HSM2.
 - HSM1 and HSM2 will skip any data sets it finds in SG1 and SG2, respectively, that are eligible to migrate to cloud storage as they perform primary space management.
 - HSMC will skip any data sets it finds that are not directed to cloud storage.
3. Specify STORAGEGROUPAFFINITY(SYS1, SYS2) to HSMC.
 - SG1 and SG2 will be processed even though HSMC runs on SYSC. (SG1 and SG2 are storage groups that are associated with both SYS1 and SYS2 in the MIGRATE SYSTEM/SYSPLEX NAME field of the storage group definition.)
4. Specify the EARLYMIGRATION(YES) parameter to HSMC, along with PRIMARYSPMGMTSTART(1800 0400).
 - The HSMC host will start its primary space management processing at 6 PM.

- HSMC will check the data sets for cloud migration eligibility and will migrate data sets at 6 PM that are eligible now or will be eligible after midnight. Thus, data sets that are eligible for migration after midnight can be migrated before midnight.

Starting automatic secondary space management

The REQUEST parameter of the SETSYS command can affect whether automatic secondary space management runs. When REQUEST is specified, DFSMSHsm requests the operator's permission before starting automatic secondary space management. If the operator replies NO, automatic secondary space management does not run until the next Y day in the cycle. If the operator replies YES, automatic secondary space management runs immediately after the reply, regardless of the amount of time that elapsed before the reply.

At the beginning of automatic secondary space management, DFSMSHsm requests a list of all defined management classes from SMS. If SMS does not return the list, DFSMSHsm considers SMS inactive and issues a message that automatic secondary space management cannot be performed for SMS-managed data sets.

DFSMSHsm also checks the last time automatic secondary space management ran, if the ending time was reached, and if all processing was complete. If the ending time was reached but processing was not complete, then the next time processing is scheduled to start, DFSMSHsm will resume processing at approximately the point it stopped, if the planned start time has not been changed.

Automatic secondary space management saves the resume key for every 50,000 MCD records read. The value 50,000 is stored in the MGCRCDS# field and can be patched to any other desired value.

The order of running migration cleanup and level 1 to level 2 migration depends on the point of resumption. The level 1 to level 2 migration could have run before migration cleanup if the automatic secondary space management stopped at level 1 to level 2 migration during the last run and did not complete. The threshold of level 1 migration volumes will not be checked again before the running of level migration if the automatic secondary space management functions resume processing at a point other than the beginning.

Using secondary host promotion, you can enable another host to take over secondary space management from a failed host. Secondary space management is typically run from only one host, so the ability to run the function from another host, should the need arise, can be quite useful.

Related reading

- For more information about the exceptions to obtaining a new list of management classes, see [“DFSMSHsm retrieval of management classes and storage groups” on page 154](#).
- For more information about secondary host promotion, see the sysplex topic in [z/OS DFSMSHsm Implementation and Customization Guide](#).

Migration cleanup

Before automatic secondary space management starts migration cleanup, it determines if there is any migration cleanup task running in another DFSMSHsm host. If there is, the migration cleanup will not be performed on this DFSMSHsm host. This avoids any unexpected results that might be created by running two migration cleanup tasks against the same MCDS at the same time.

When migration cleanup starts, it checks to see if any internal TAPECOPY processing is needed, and if so, schedules the internal TAPECOPY MWEs. Internal TAPECOPY processing could be needed because of duplexing errors or ABARS recovery of ML2 volumes.

Migration cleanup deletes the following:

- Expired migrated data sets. Migration cleanup deletes the data sets if:
 - For both SMS-managed and non-SMS-managed data sets, the data set has passed the expiration date indicated in the data set's VTOC entry and the EXPIREDDATASETS(SCRATCH) parameter is specified in the SETSYS command.

- For SMS-managed data sets, the data set does not have an expiration date but has passed the expiration attributes specified in the management class to which it is associated.
- For an SMS-managed NOSCRATCH GDS, the data set rolled off while on a level 0 volume and migrated, with rolled off GDG=EXPIRE in the management class and no expiration date in the VTOC.

See [“Specifying expiration attributes” on page 36](#) and [“Specifying how to scratch expired data sets” on page 49](#) for details on SMS-managed data sets.

- Migration copies of the data sets recalled from SDSP data sets. Because multiple recall tasks can access the small data set packing data sets concurrently, recall does not erase the migration copy.

Even though SDSP data sets used during migration cleanup are periodically released when not in use, and SDSP data sets needed by recall or aggregate backup are released when needed, it is recommended that you do not run secondary space management when automatic primary space management is running in order to avoid contention for SDSP data sets.

Every 30 seconds, migration cleanup checks to determine if recall or aggregate backup needs an SDSP data set. The value 30 is stored in the MCVT#CHK field and can be patched to any other desired value.

- MCDS data set (MCD) records for nonreconnectable data sets that are older than the number of days that are specified by the *recalldays* value of the SETSYS MIGRATIONCLEANUPDAYS command.
- MCDS data set (MCD) records for reconnectable data sets that have passed their predicted remigration date by the number of days that are specified by the *reconnectdays* value of the SETSYS MIGRATIONCLEANUPDAYS command. This occurs only if the *recalldays* criterion is also met.
- Daily statistics records and volume statistics records that are older than the number of days specified by the *statdays* value from the SETSYS MIGRATIONCLEANUPDAYS command.
- Migration copies not scratched during recall, deletion, or GDG roll-off of non-SMS-managed migrated data sets. These copies exist if an error occurred during recall or deletion, or if a generation data set had not passed its expiration date when the roll-off occurred.

Migration cleanup example

[Figure 7 on page 61](#) shows an example of SMS-managed volumes after migration cleanup has occurred for the conditions shown in [Figure 6 on page 57](#). Data set STANDMC.ML.F2 is deleted from the ML1 volumes because it has passed the EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS values from management class STANDMC. Data set LARGEMC.ML.G1 is deleted from the ML2 tapes because it has passed the EXPIRE AFTER DAYS NON-USAGE and the EXPIRE AFTER DATE/DAYS values from the management class LARGEMC.

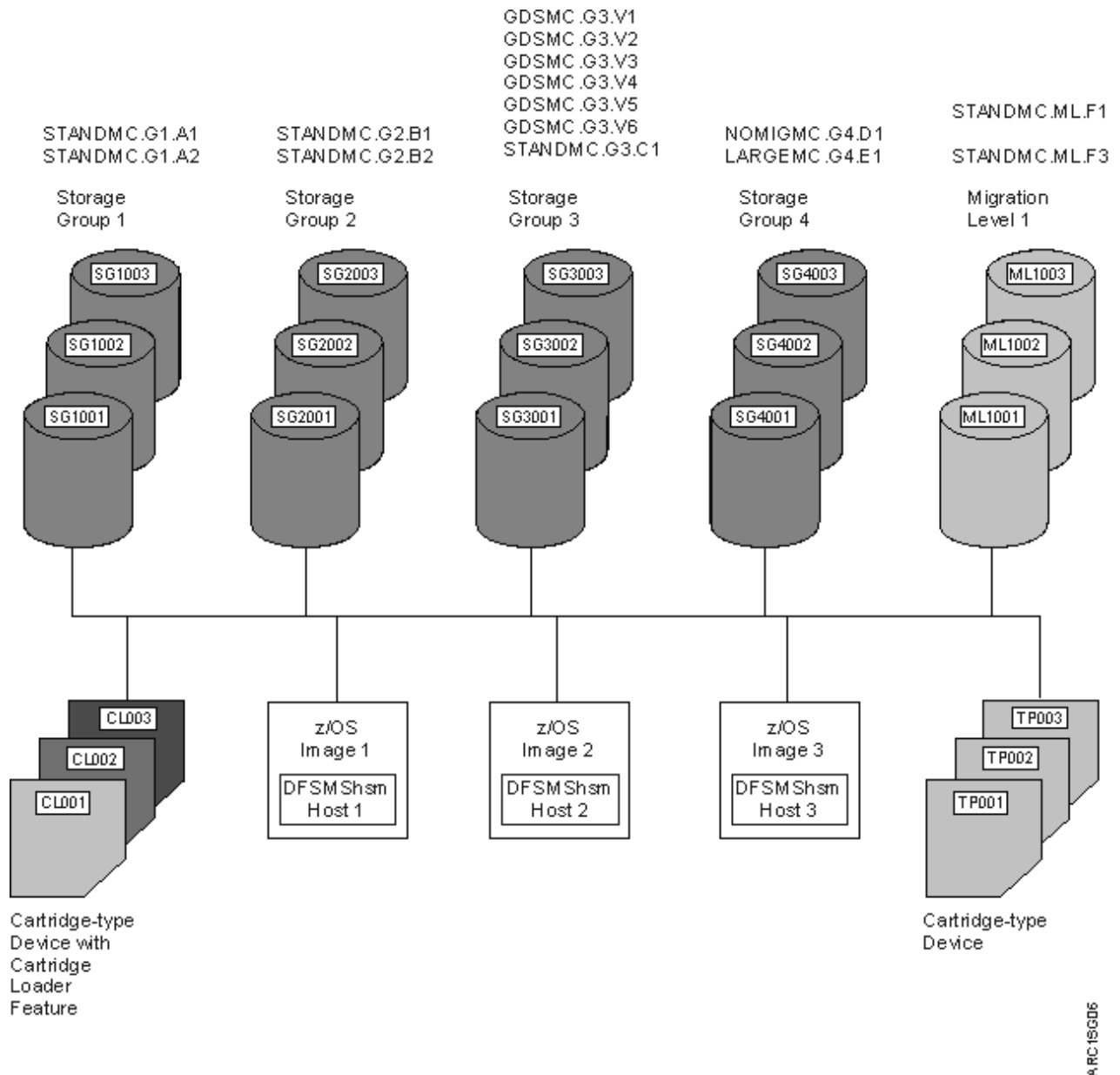


Figure 7. Example System of SMS-Managed Volumes after Migration Cleanup

Level 1 to level 2 migration

Before automatic secondary space management starts level 1 to level 2 migration, it performs the following checks:

- Checks to see there is any command level 1 to level 2 running. If yes, the level 1 to level 2 migration of automatic secondary space management will not start. There is no reason to run two level 1 to level 2 migrations at the same time.
- Performs a space check on the all ML1 volumes. If any NOOVERFLOW ML1 volume has an occupancy that is equal to or greater than its high threshold, then all eligible data sets migrate from the NOOVERFLOW ML1 volume to ML2 volumes. For OVERFLOW volumes, the percentage of occupied space for each ML1 OVERFLOW volume will be calculated and then DFSMSHsm will compare the average value with the specified threshold. Secondary space management is started for all eligible ML1 OVERFLOW data sets if the overflow volume pool threshold is met or exceeded. If you choose to use OVERFLOW volumes, you can specify the average overflow volume pool capacity threshold at which you want automatic secondary space management to migrate ML1 OVERFLOW volumes to ML2 volumes using the SETSYS ML1OVERFLOW command:


```
SETSYS ML1OVERFLOW(THRESHOLD(threshold))
```

Eligible SMS-managed data sets are those data sets that have passed the value you specified with the LEVEL 1 DAYS NON-USAGE attribute in their management class. Eligible non-SMS-managed data sets are those that have passed the value you specified with the MIGRATIONLEVEL1DAYS parameter of the SETSYS command.

ML2 volumes can be either DASD or tape. The TAPEMIGRATION parameter of the SETSYS command specifies what type of ML2 volume is used. The SETSYS command for DFSMSHsm host 2 specifies ML2 migration to tape.

Related reading

For more information about data set eligibility from Level 1 to Level 2, see [“Specifying when data sets become eligible for level 1 to level 2 migration”](#) on page 101. For information about the SETSYS ML1OVERFLOW command, see [“ML1OVERFLOW: Manage ML1 OVERFLOW volumes”](#) on page 1160.

Migration from SDSP data sets

Although SDSP data sets used during SSM migration cleanup are periodically released when not in use, it is still recommended that you do not run secondary space management when automatic primary space management is running in order to avoid possible contention for SDSP data sets.

Every 30 seconds, level 1 to level 2 migration checks to determine if recall or aggregate backup needs an SDSP data set. The value 30 is stored in the MCVT#CHK field and can be patched to any other desired value.

Migration to a tape migration level 2 volume

At the beginning of a level 1 to level 2 migration task, DFSMSHsm selects the partial tape with the largest percentage that is written. This might be one of the following tapes:

- The tape last used by this task
- A tape ML2 volume from the list of not-associated, partially empty tape volumes
- An empty tape (if available), but only if no partially empty tape is available

DFSMSHsm selects tapes differently if you are using the duplex tape option for migration tapes.

DFSMSHsm does not migrate a data set to a tape ML2 volume if the data set needs to be backed up.

When a tape unit and a tape volume are allocated, they remain allocated and mounted until the migration is completed or the volume becomes full or is needed by another function. When level 1 to level 2 migration is completed, the tape volume is deallocated and demounted.

Related reading

- For more information about the duplex tape option, see [“Duplex tape option”](#) on page 436.
- For more information about contention for migration volumes, see [“Tape migration level 2 volume contention”](#) on page 136.

Migration to a DASD level 2 volume

For each data set that migrates from an ML1 volume to DASD ML2, DFSMSHsm uses the set of initial characters of the data set name and the key ranges that you defined to choose the DASD ML2 volume. The *key range* is the total set of data set names whose initial characters are between a low key and a high key in collating sequence.

If DFSMSHsm fails the migration of data sets from an ML1 volume to an ML2 volume because the ML2 volume being used with that key range does not have enough space, DFSMSHsm does the following:

1. Disassociates the full volume from the key range.

2. Tries to assign an available DASD ML2 volume to the key range in place of the full volume. An available DASD ML2 volume is a volume that meets all of the following criteria:
 - It has been added to DFSMSHsm as an ML2 volume. The ADDVOL *valid* MIGRATION(MIGRATIONLEVEL2) command adds ML2 volumes to DFSMSHsm.
 - It has not been associated with a key range. The DEFINE MIGRATIONLEVEL2(KEYS(*key* . . .) VOLUMES(*valid* . . .)) command associates ML2 volumes with key ranges.
 - DFSMSHsm has never chosen it as a replacement for an ML2 volume.
 - It does not have the DRAIN attribute in its ADDVOL command.
3. Retries the migration of the data set one time.

If there is no ML2 volume to choose as a replacement volume, DFSMSHsm stops level 1 to level 2 migration for the key range defined for the full ML2 volume. DFSMSHsm continues migrating data sets for all other key ranges.

Related reading

- For more information about how to define key ranges, see [“MIGRATIONLEVEL2: Specifying DASD level 2 volumes and their associated key range structure” on page 831.](#)
- For more information about what to do to make new ML2 volumes available when all ML2 volumes are full, see [“Making migration level 2 key range volumes available” on page 551.](#)

Output media error handling

During level 1 to level 2 migration of automatic secondary space management processing, if an error condition occurs on any output media (tape or DASD), a message is sent to the operator's console to inform the operator of the error condition, and the task is stopped. DFSMSHsm does not remove this message from the console; therefore, the operator must enter a console command to remove the message.

If you are using the duplex tape option for your migration tapes, and the error condition occurs on the alternate volume, the function is not stopped.

When the error condition that caused the level 1 to level 2 migration to end is corrected, the operator can restart level 1 to level 2 migration if the current time is within the automatic secondary space management window. You can use one of the following commands to restart level 1 to level 2 migration:

- RELEASE MIGRATION(AUTO)
- RELEASE MIGRATION
- ADDVOL (volume of the device type in error) and RELEASE MIGRATION or RELEASE MIGRATION(AUTO)

Related reading

For more information about the duplex tape option, see [“Duplex tape option” on page 436.](#)

Installation exits called during level 1 to level 2 migration

Two installation exits can be called during level 1 to level 2 migration: the tape data set exit and the second-level migration data set exit.

The tape data set exit (ARCTDEXT) receives control (if it is enabled) when an output tape data set is opened if the tape security option is expiration date (EXPIRATION) or expiration date including password-protected data sets (EXPIRATIONINCLUDE). You can use the tape data set exit to change the expiration date to be recorded in the tape label to a date other than 99365.

The second-level migration data set exit (ARCMEXT) receives control when a migrated data set is selected for migration. You can use this exit to further control whether data sets migrate from ML1 volumes to other migration volumes. For example, you can use this exit to hold small data sets in SDSP data sets on ML1 volumes instead of allowing them to migrate to ML2.

Related reading

For more information about DFSMSHsm exits, see [z/OS DFSMS Installation Exits](#).

Example of level 1 to level 2 migration

Figure 8 on page 64 shows SMS—managed volumes after level 1 to level 2 migration. In this figure, at least one of the ML1 volumes had reached or exceeded its threshold of occupancy when DFSMSHsm performed its space check. Data set STANDMC.MLF3 migrates from the ML1 volumes to the tape ML2 volumes (a cartridge-type device) because it has passed the LEVEL 1 DAYS NON-USAGE value from management class STANDMC.

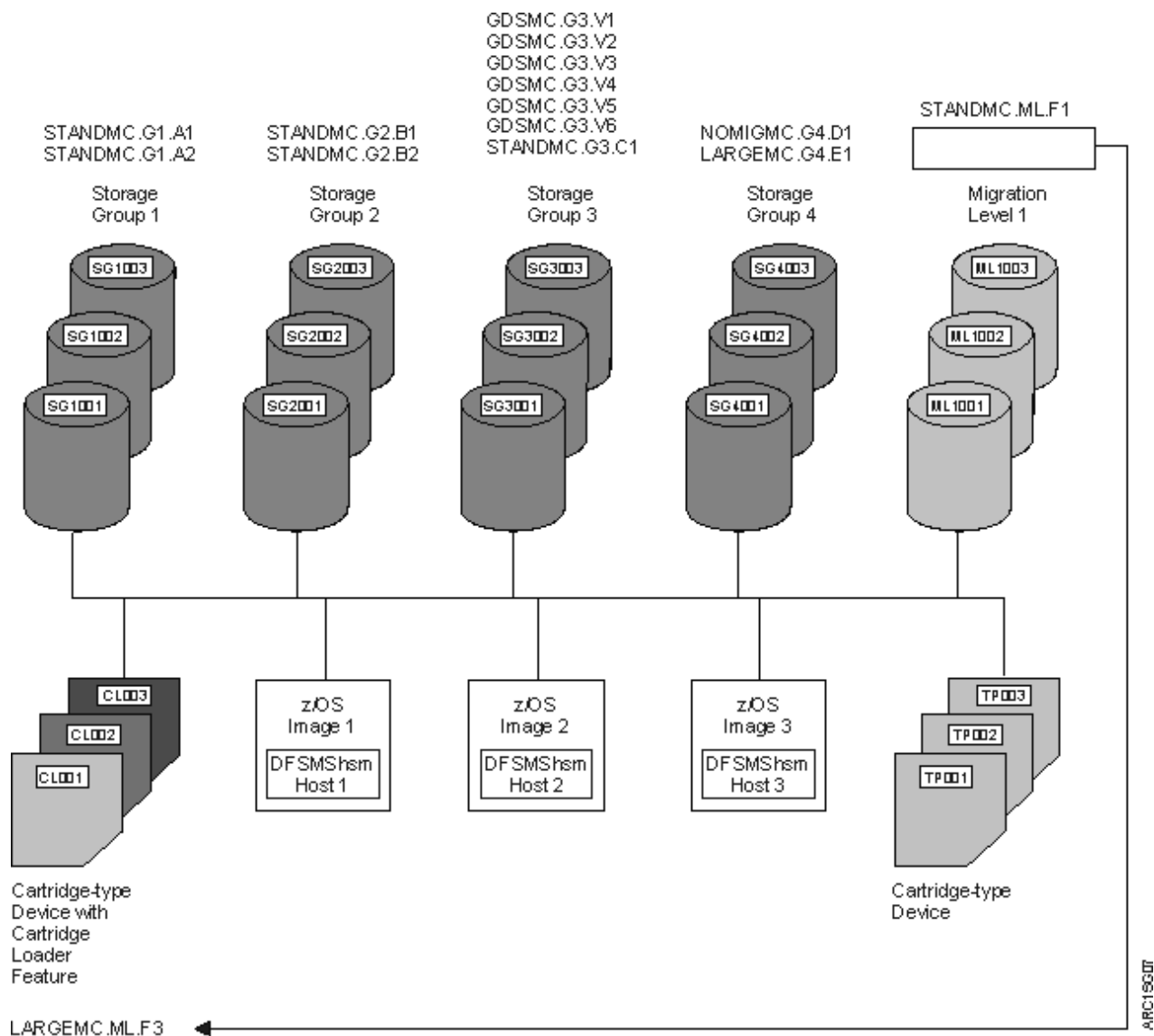


Figure 8. Example System of SMS-Managed Volumes after Level 1 to Level 2 Migration

Starting automatic primary space management

At the beginning of automatic primary space management in each of the DFSMSHsm hosts, if SMS is active, DFSMSHsm requests a list of all defined management classes and storage groups from SMS if the SMS configuration has changed since the last time DFSMSHsm requested this information. If you have not provided any management class definitions, DFSMSHsm builds a default management class definition. If SMS does not return any storage group definitions, DFSMSHsm considers SMS to be inactive.

If SMS is active, DFSMSHsm processes any SMS volumes that the z/OS image is allowed to process and that have the AM=Y, AM=I, or AM=P attribute, and any non-SMS-managed primary volumes. If SMS is not installed and active, DFSMSHsm processes only non-SMS-managed primary volumes. For purposes of the example, it is assumed that SMS is installed and active.

Related reading

For more information about the exceptions to obtaining a new list of management classes, see [“DFSMSHsm retrieval of management classes and storage groups” on page 154.](#)

Automatic primary space management

At the beginning of automatic primary space management, DFSMSHsm determines which volumes are candidates for processing. Volumes in storage groups that have a z/OS image specified by the MIGRATE SYSTEM/SYSPLEX NAME have a z/OS image affinity.

DFSMSHsm uses the following order of processing as a method of controlling input path contention. This order is subject to prioritizing for SMS volumes restricted to processing by one z/OS image.

1. Affinity SMS volumes
2. Retry of in-use affinity SMS volumes
3. Non-SMS and non-affinity SMS volumes

After DFSMSHsm has attempted to process all the candidate volumes eligible for automatic primary space management, it retries the selection of any of these volumes that were found to be in use by another DFSMSHsm function.

If multiple DFSMSHsm hosts perform automatic primary space management, only the first host to process a given volume performs the function.

Specifying a storage group priority for space management functions

The storage group processing priority of the volumes assigned to an SMS storage group may be specified within the storage group definition. The default value is 50. DFSMSHsm will process all of the SMS volumes in the order of the processing priority, with 100 being the highest and 1 being the lowest. This processing priority can help ensure that storage groups that need to be processed first during interval migration, on-demand migration, and primary space management are done so in order.

Note: This processing priority can help ensure that storage groups that will be receiving data sets as the result of class transitions are processed first, in order to free up the necessary space for the expected new allocations.

Overview of two-pass automatic primary space management

DFSMSHsm-managed volume space management is performed in two passes, as follows:

1. **Pass 1** deletes the appropriate data sets, releases unused space, performs fast subsequent migration, and determines which data sets on the volume are candidates for migration, class transitions, or extent reduction. Performs class transitions with management class changes only.
2. **Pass 2** processes the candidates for extent reduction, transitions data sets, and migrates data sets until the low threshold of occupancy is reached.

Pass 1 of automatic primary space management

During pass 1, DFSMSHsm examines every data set on a volume, independent of the occupancy of the volume, without moving data sets. Unless the data set is an unmatched data set, DFSMSHsm compares the integrity age with the unused age to make sure that it is allowed to process the data set. The unused age is the difference between the current date and the date-last-referenced.

Rule: The unused age must be greater than or equal to the integrity age.

Pass 1 of DFSMSHsm-managed volume space management performs the following functions:

- Issues a message for unmatched data sets
- Deletes temporary data sets
- Deletes expired data sets
- Determines data sets that are eligible for migration
- Excludes from migration to tape those data sets that are eligible to be backed up
- Determines data sets that are eligible for extent reduction
- Releases unused space for eligible data sets
- Performs fast subsequent migration for eligible data sets
- Determines data sets that are eligible for class transitions
- Performs class transitions with management class changes only

For each of the data sets, DFSMSHsm uses the attributes specified in the management class with which the data set is associated. If no management class is associated with the data set, DFSMSHsm uses the default management class.

Related reading

- For more information about the integrity age, see [“Integrity age for space management” on page 56](#).
- For more information about unmatched data sets, see [“Unmatched data sets” on page 66](#).
- For more information about default management classes, see [“Using the default management class” on page 40](#).

Unmatched data sets

If the catalog entry, the data set VTOC entry, or both are missing, the data set is an unmatched data set. For an unmatched data set, the space management and backup exit (ARCSAEXT) is invoked, if the exit is enabled. If the exit indicates that processing continues, or if the exit is not enabled, DFSMSHsm issues a message, does no further processing on the unmatched data set, and begins processing the next data set. If the exit indicates that processing is not to continue, DFSMSHsm begins processing the next data set without issuing any message.

Deleting temporary data sets

If for some reason (such as a system outage) SMS-managed temporary data sets are not deleted at end of job, DFSMSHsm attempts to delete them during automatic primary space management. A flag in the data set VTOC entry indicates that a data set is temporary.

If no cross-system serialization exists, or if cross-system serialization is not performed for temporary data sets, an in-use temporary data set can be deleted. In a JES3 environment, the data set must be two days old before it can be deleted.

Deleting expired data sets

If the EXPIREDDATASETS(SCRATCH) parameter is specified in the SETSYS command, DFSMSHsm deletes any non-VSAM data set whose VTOC entry shows that the data set has passed its expiration date and any VSAM data set whose catalog entry shows that the data set has passed its expiration date. The management class expiration attributes control the expiration of non-VSAM data sets that do not have a valid expiration date in their data set VTOC and VSAM data sets that do not have a valid expiration date in their base cluster catalog entry.

Note: For multivolume non-VSAM data sets, DFSMSHsm looks at the VTOC of the first volume.

If concurrent copy was used to back up the data set and a system failure or cancel of DFSMSHsm occurred after the concurrent copy session was established, then the data-set-changed indicator can be off even

though the data set was not successfully backed up. This can result in the data set prematurely meeting the expiration criteria.

DFSMSHsm provides a patch byte that enables users to override the requirement that an SMS-managed data set have a backup copy before it is expired. For more information about this patch, see [Tuning DFSMSHsm in z/OS DFSMSHsm Implementation and Customization Guide](#).

DFSMSHsm deletes rolled-off generation data sets that meet either of the following conditions, regardless of the EXPIREDDATASETS parameter of the SETSYS command.

- It is associated with a management class that has the attribute ROLLED-OFF GDS ACTION=MIGRATE and both of the following apply:
 - It does not have an expiration date in the data set VTOC entry.
 - It has passed the date and time limit specified by the EXPIRE AFTER DAYS NON-USAGE and EXPIRE AFTER DATE/DAYS attributes in the management class.
- It is associated with a management class with the attribute ROLLED-OFF GDS ACTION = EXPIRE and it has no expiration date in the data set VTOC entry.

Note: DFSMSHsm deletes active generation data sets in the same way it deletes non-GDS data sets.

Determining eligible data sets for class transitions and migration

DFSMSHsm checks the COMMAND OR AUTO MIGRATE attribute in the management class to which the data set is associated. If the value of the attribute is BOTH, DFSMSHsm determines whether the data set is eligible for a class transition or migration.

A data set is eligible for a class transition if the data set has met or exceeded any of the three class transition criteria specified within the assigned management class (TIME SINCE CREATION, TIME SINCE LAST USE, or PERIODIC). If a data set transitions to a new management class, then the data set will be eligible for another transition when the volume is next space managed, based on the criteria within the newly assigned management class. If a different management class is not assigned during a transition, then a data set will not transition again if either TIME SINCE CREATION or TIME SINCE LAST USE are the criteria that is used. If a PERIODIC value is specified, then a data set will transition as often as needed.

Four conditions determine whether a data set is eligible for migration:

- The MIGRATE value of the ROLLED-OFF GDS ACTION attribute
- Whether the data set is a generation that exceeds the value specified by the # GDG ELEMENTS ON PRIMARY attribute in the management class
- Whether the data set is an SMS-managed MVS or IMS GSAM checkpointed data set
- The number of days since the data set was last used

Rolled-off generation data sets that belong to a management class with the attribute ROLLED-OFF GDS ACTION=MIGRATE are always identified as migration candidates with the highest migration priority.

Non-rolled-off generation data sets that exceed the value specified with the # GDG ELEMENTS ON PRIMARY attribute for their management class are always identified as candidates with priority for early migration. The priority of such data sets is just below that for rolled-off generation data sets.

For both command and event-driven migration, DFSMSHsm determines whether the data set being migrated is a checkpointed data set. DFSMSHsm migration, including extent reduction, is delayed for a fixed number of days for SMS-managed MVS and IMS GSAM checkpointed data sets. The default delay is five days. If the data set is checkpointed, DFSMSHsm determines whether the data set is eligible for migration. A checkpointed data set is eligible for migration when the date-last-referenced, plus the number of days the data set is to be treated as unmovable, are less than or equal to the current date. For data set command migration, error messages are issued for ineligible data sets.

DFSMSHsm provides a patch byte that enables users to modify the number of days that must have elapsed since the date-last-referenced for a checkpointed data set to be eligible for migration. For more information about this patch, see [Tuning DFSMSHsm in z/OS DFSMSHsm Implementation and Customization Guide](#).

Checkpointed data sets must meet a minimum age value before being considered for migration. The default is five days and can be changed using a patch. Once the minimum age value is met, the checkpointed data sets can be identified as migration candidates if the time since they were last used meets or exceeds the value specified by the PRIMARY DAYS NON-USAGE attribute for their management class.

All other data sets, including generation data sets that were not given special migration priority, are identified as migration candidates if the time since they were last used meets or exceeds the value specified by the PRIMARY DAYS NON-USAGE attribute for their management class.

If the COMMAND OR AUTO MIGRATE attribute in a data set's management class is set to BOTH, with the PRIMARY DAYS NON-USAGE attribute set to 9999, and the # GDG ELEMENTS ON PRIMARY attribute set to 1, then relative zero-generation data sets can remain on a user's primary disk for up to 30 years. (The # GDG ELEMENTS ON PRIMARY attribute overrides the PRIMARY DAYS NON-USAGE attribute.) When a generation moves from zero to -1, it becomes eligible to migrate immediately, independent of the PRIMARY DAYS NON-USAGE value.

When a data set is identified as a migration candidate, it is also identified for the target migration volume device type: DASD or tape. The device type chosen depends on the value in the LEVEL 1 DAYS NON-USAGE attribute of its management class. If the value of the attribute is 0, the data set is identified for migration to ML2 tape. If the value of the attribute is 1–9999 or NOLIMIT, the data set is identified for migration to ML1 DASD.

Related reading

For more information about DFSMSHsm support for checkpointed data sets, see the topic about user data sets in [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Determining eligible data sets for backup

After DFSMSHsm determines that a data set is eligible for a class transition or migration, it checks whether the data set needs to be backed up.

For class transitions, a data set must be backed up if the TRANSITION COPY TECHNIQUE specifies that fast replication should be used or preserve mirror is preferred. If the data set has not been backed up and fast replication or preserve mirror are preferred, then standard data movement will be performed. If the data set has not been backed up and fast replication is required, then the transition will fail. A backup copy is not required when FC XRC Primary is set.

A backup copy is required whenever a fast replication technique other than preserve mirror required or FC XRC primary is used because DFSMSHsm processing for the data set completes before the data set is physically copied. If the TRANSITION TECHNIQUE preserve mirror required is specified to allow FlashCopy to Global Mirror Primaries, a valid backup copy must exist before the data set residing on a Global Mirror primary volume is transitioned. While unlikely, it is possible for the physical copy to fail. Requiring a backup copy of the data set ensures the availability of the data set should the physical copy fail. Preserve mirror and Remote Pair FlashCopy (RPF) for XRC ensures that the data set is physically copied at both the local and remote site, which protects against errors.

For migration, a data set must be backed up if all of the following criteria are met:

- The data-set-changed indicator in the data set VTOC entry indicates that the data set has been changed.
- The management class contains the attribute AUTO BACKUP=Y.
- The management class ADMIN OR USER COMMAND BACKUP attribute contains either of the values ADMIN or BOTH.
- The storage group has the attribute AUTO BACKUP=Y.

DFSMSHsm identifies any migration-eligible data set that must be backed up. If the data set that must be backed up is identified for migration to tape, DFSMSHsm removes the identification as a migration candidate.

If concurrent copy was used to backup the data set and a system failure or cancel of DFSMSHsm occurred after the concurrent copy session was established, then the data-set-changed indicator can be off even though the data set was not successfully backed up. If this happens, migration will not identify the data set as needing a backup copy.

Determining eligible data sets for extent reduction

After determining if the data set is eligible for backup, DFSMSHsm determines if the data set is eligible for extent reduction. DFSMSHsm identifies a non-VSAM single-volume data set as eligible for extent reduction if all of the following criteria are met:

- The data set has as many extents or more than the value specified by the MAXEXTENTS parameter of the SETSYS command.
- The data set also meets data integrity age.
- If the data set is an SMS-managed MVS or IMS GSAM checkpointed data set, it meets the requirements for date-last-referenced plus the number of days the data set is to be treated as unmovable.
- The value of the COMMAND OR AUTO MIGRATE attribute is BOTH.

Note:

1. DFSMSHsm does not perform extent reduction on any extended format sequential data set.
2. Extent reduction will not occur on SMS-managed data sets with COMMAND OR AUTO MIGRATE=NONE specified, or on non-SMS-managed data sets with SETMIG NOMIGRATION specified.

Releasing unused space

For all physical sequential, partitioned, and extended format VSAM KSDS single-volume data sets that belong to a management class with the attribute PARTIAL RELEASE=Y (or C for VSAM), DFSMSHsm invokes the PARTREL function of DADSM if either of the following conditions is met:

- The data set is not a candidate for extent reduction.
- At least one track will be released by PARTREL.

DFSMSHsm issues a message indicating the result of the partial release and the number of tracks freed for each data set if the activity logging message level is FULL or REDUCED.

Performing fast subsequent migration

Fast subsequent migration allows unchanged data sets that are recalled from a single ML2 tape to be reconnected to the ML2 tape from which they were most recently recalled. For a data set to be eligible for reconnection, it must meet standard migration eligibility requirements as well as fast subsequent migration eligibility requirements.

Related reading

- For more information about fast subsequent migration, see [“Reconnecting data sets using fast subsequent migration”](#) on page 94.
- For a list of reconnection eligibility requirements, see [Fast subsequent migration \(reconnection\) eligibility requirements](#).

Performing class transition with management class changes only

Change the data set management class if the storage group and storage class are not changed.

Pass 1 processing example

Figure 9 on page 70 shows the results of pass 1 processing. Data set STANDMC.G1.A1 was deleted because it exceeded the EXPIRE AFTER DATE/DAYS and EXPIRE AFTER NON-USAGE values of the STANDMC management class. Data set GDSMC.G3.V1 was deleted because it is a rolled-off GDS with

the EXPIRE value in the ROLLED-OFF GDS ACTION attribute in GDSMC management class. Data set NOMIGMC.G4.D1 was deleted because it exceeded the EXPIRE AFTER DAYS NON-USAGE attribute of the NOMIGMC management class. Data set STANDMC.G1.A3, which has not changed from Figure 8 on page 64, was recalled from ML2. It is reconnected during Pass 1 to its original ML2 tape.

Note: The data sets in the NOMIGMC management class never migrate because the COMMAND OR AUTO MIGRATE attribute has the value of NONE, but they are processed for expiration and for PARTIAL RELEASE.

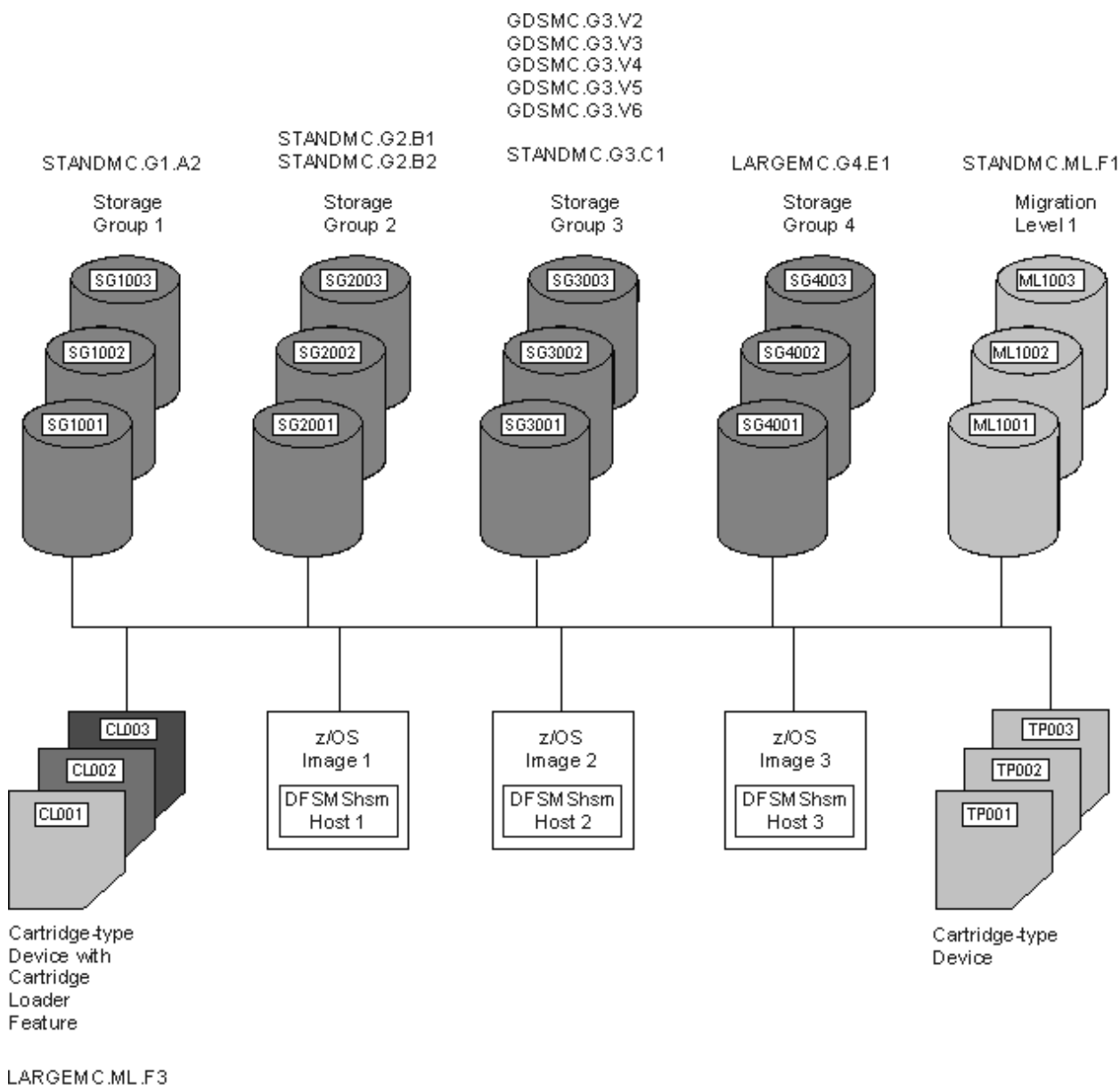


Figure 9. Example System of SMS-Managed Volumes after DFSMSHsm-Managed Volume Management Pass 1

Pass 2 of automatic primary space management

During pass 2 of automatic primary space management, DFSMSHsm transitions or migrates data sets from the volumes until the space occupied on the volumes is equal to or less than the space specified by the low-threshold attributes contained in the storage groups to which the volumes belong. DFSMSHsm migrates only those data sets that have been identified for class transitions, migration, or extent reduction by pass 1 of DFSMSHsm-managed volume space management. The following steps are performed in pass 2:

1. Transition data sets that are candidates both for a class transition and extent reduction.
2. Migrate data sets that are candidates both for migration and for extent reduction.
3. Migrate data sets that are candidates only for extent reduction. Schedule a recall for each data set.
4. Transition data sets that are candidates for a class transition in FIFO until the low threshold of occupancy is reached.
5. Migrate data sets in priority order until the low threshold of occupancy is reached.

Note: If a data set is eligible for both a class transition and migration, then the data set will migrate.

Transition data sets that are candidates for a class transition

Data sets that are eligible for a class transition are processed first. If a data set is eligible for both a class transition and migration, then the data set will be migrated.

When a data set meets the management class criteria for a class transition, then DFSMSHsm invokes the ACS routines with the ACS environment SPMGCLTR to determine if there is a newly assigned management class, storage class or storage group. If that data set is assigned a new management class and the storage class and storage group remain unchanged, then DFSMSHsm will ALTER that data set to reassign the data set to the newly assigned management class. No data movement will occur. The newly assigned management class will be used to determine how the data set is managed during future space management processing.

If a data set is assigned a new storage class or storage group, then the data set will be moved. DFSMSHsm will invoke DFSMSdss to move the data set with the logical data set copy and delete operation. This process will invoke SMS to use the newly assigned storage class and/or storage group for the allocation.

Data sets are transitioned until the occupied space is equal to or less than the low threshold specified by the storage group or until no more class transition candidates exist on the volume. If the low threshold has been met, DFSMSHsm ends automatic primary space management processing for that volume. If the low threshold has not been met, then primary space management processing continues for migration and extent reduction candidates.

DFSMSHsm does not keep any records for transitioned data sets within its control data sets. Function Statistics Records (FSR) are used to track transition processing. Storage administrators may use the various reporting techniques available for FSR records to analyze DFSMSHsm's class transition processing. A sample report is provided for the DFSMS Report Generator. (See [“Running new reports with report generator” on page 575.](#)) The DFSMSHsm REPORT function also provides basic reporting information.

By default, failing messages are not issued for data sets that fail a transition because exclusive serialization could not be obtained. It is recommended that these data sets be analyzed by post processing the FSR records with failing return code RC68 and reason code REAS410. If desired, the patch offset which enables these messages is:

```
PATCH .MGCB.+EF BITS(...1...)
```

Migrate candidates for both migration and extent reduction

Any data set that is identified as a candidate for extent reduction always migrates. However, a data set that is identified as a candidate for both migration and extent reduction is not recalled.

Migrate candidates for extent reduction

DFSMSHsm migrates the data sets that are candidates for extent reduction and schedules recalls for those same data sets. During the migration, DFSMSHsm moves only the valid data from the data sets. Therefore, when the data sets are recalled into fewer extents, they may occupy less space.

Note, also, the following rules:

- If the data set is eligible for a class transition with data movement (DSS Copy/Delete) and extent reduction, then only the class transition will be performed.

- If the data set is eligible for a class transition without data movement (i.e. only the management class is changing) and extent reduction, then extent reduction will be performed after the class transition (again, with only the management class changing, without data movement).
- If the data set is eligible for a class transition, migration and extent reduction, then only the migration will be performed.
- If the data set is eligible for extent reduction only, then the data set will be processed after all data sets eligible for extent reduction and class transition, and after all the data sets eligible for extent reduction and migration have been processed.

Place candidate data sets in priority order for migration

After performing class transitions and migrating data sets for extent reduction, DFSMSHsm places the remaining candidates for migration in order by priority. DFSMSHsm gives the highest priority for migration to rolled-off generation data sets. It gives second priority to non-rolled-off generation data sets that have been identified for early migration. These are the non-rolled-off data sets that exceed the number specified by the # GDG ELEMENTS ON PRIMARY attribute.

After the two highest priority groups, priority is assigned to all other data sets on the basis of size and age. The age, for migration priority purposes, is an eligibility age determined from the last-used date and the value of the PRIMARY DAYS NON-USAGE parameter. Small data sets must be eligible for a longer time to attain the same or higher priority as larger data sets.

Migrate data sets

After putting the candidate data sets in priority order, DFSMSHsm compares the occupied space on the volume to the low threshold for the volume. If the occupied space is more than the low threshold, DFSMSHsm migrates the highest priority data set. Data set size is a determining factor for the order in which data sets are migrated. Managing an entire extended address volume (EAV) with one threshold is not sufficient because average data set size in the cylinder-managed space will likely exceed the average data set size in track-managed space. This will result in data sets allocated in the cylinder-managed space being migrated at a higher rate than data sets allocated in the track-managed space, potentially causing no space in the track-managed space being freed. To prevent this, a unique threshold for the track-managed space (track-managed threshold) is used to manage the free space separate from the free space on the entire volume (volume-level threshold).

Data sets that fail reconnection remain eligible for normal migration. Data sets that fail reconnection because there is no ability to serialize either the data set or the target ML2 volume are assigned the lowest migration priority.

Data sets are migrated until the occupied space is equal to or less than the low threshold specified by the storage group or until no more migration candidates exist on the volume. At that point, DFSMSHsm ends automatic primary space management processing for that volume.

As DFSMSHsm selects each data set for migration, it transfers control to the space management exit (ARCMDEXT) if the exit is installed and enabled.

Table 5 on page 72 shows the rules governing migration versus class transition, based on eligibility and the return code from the migration exit.

<i>Table 5. Rules for migration versus class transition, based on eligibility and return code from migration exit</i>			
Eligibility Status	Migration Exit RC=0 (Allow Migration)	Migration Exit RC=8 (Disable Migration)	Migration Exit RC=52 (Disable Migration and Class Transition)
Eligible for migration only	Migration	Data set processing fails with RC=45, reason 92	Data set processing fails with RC=45, reason 92
Eligible for migration and class transition	Migration	Class transition	Data set processing fails with RC=45, reason 92

Table 5. Rules for migration versus class transition, based on eligibility and return code from migration exit (continued)

Eligibility Status	Migration Exit RC=0 (Allow Migration)	Migration Exit RC=8 (Disable Migration)	Migration Exit RC=52 (Disable Migration and Class Transition)
Eligible for class transition only	Class transition	Class transition	Data set processing fails with RC=45, reason 92

If any data set is directed to tape during DFSMSHsm-managed volume space management, a tape volume is allocated and mounted. If you are using the duplex tape option for migration, two tape volumes are allocated and mounted. When a tape volume is allocated for DFSMSHsm-managed volume migration, it remains allocated until it becomes full or is needed for some other DFSMSHsm function. When the DFSMSHsm host finishes DFSMSHsm-managed volume migration, DFSMSHsm deallocates and demounts the tape volume.

Related reading

- For more information about ARCMDEXT, see *z/OS DFSMS Installation Exits*.
- For more information about having tapes available during DFSMSHsm-managed volume space management, see [“Automatic space management operation examples”](#) on page 56.

Output media error handling

Class transitions may only be directed to DASD, via the SMS ACS routines.

You can use both tape and DASD as migration devices concurrently. During such an operation, if migration to either tape or DASD is ended because of an error condition, a message is issued, but migration continues to the device type that is not affected by the error condition. A highlighted message is sent to the operator’s console to inform the operator of the error condition. DFSMSHsm does not remove this message from the console; the operator must issue a console command to remove it.

After correcting the error condition that caused the migration to tape or DASD to end, the operator can allow DFSMSHsm to resume migration to the device that was in error by issuing one of the following commands:

- RELEASE MIGRATION(AUTO)
- RELEASE MIGRATION
- ADDVOL (for a volume of the device type in error)

If error conditions make both tape and DASD unavailable as migration targets, migration processing ends, a message is issued to the operator, and migration is held.

Names of migrated data sets

When DFSMSHsm migrates a data set, it changes the catalog entry for that data set. The data set retains its name, so the users of the data set know how to address it, but the location is changed to MIGRAT. However, DFSMSHsm gives the migration copy of the data set a name of the form:

```
prefix.HMIG.Tssmmhh.user1.user2.Xydd
```

If patch-enabled, the following dataset name will be generated:

```
prefix.HMIG.Tcccchh.user1.user2.Xydd
```

where:

- *prefix* is the prefix you specify with the SETSYS MIGRATEPREFIX command.

- HMIG and T are constants.

Note: If two or more data sets with the same first two high-level qualifiers are migrated during the same second on the same day, DFSMSHsm assigns a letter other than T to one of these data sets.

- *ssmmhh* is the time in seconds, minutes, and hours.

If the format is *Tcccchh*, then *cccchh* is the time in hundredths of seconds from the beginning of the hour converted to four alphabetic characters, and *hh* is hours.

The four characters *cccc* may be converted to time as follows:

1. Create a variable for each character of the string *cccc*, using *p1* through *p4*, from left to right. Then assign a value 0 thru 24 for each character A thru Y in the sequence. (Note that 25 characters are used, for A through Y, but not Z.)
2. Find the total hundredths of seconds from the beginning of the hour using this formula:

```
hunds_seconds_begin_hour = p1+(p2*25)+(p3*625)+(p4*625*25)
```

3. Find the minutes, seconds, and hundredths using these formulae (% = integer division):

```
minutes = hunds_seconds_begin_hour%(60*100)
seconds = (hunds_seconds_begin_hour-minutes*60*100)%100
hundredths = hunds_seconds_begin_hour-minutes*60*100-seconds*100
```

- *cccchh* is the time in hundredths of seconds from the beginning of the hour converted to four alphabetic characters (*cccc*) and hours (*hh*).

The four characters *cccc* may be converted to time as follows:

1. Create a variable for each character of the string *cccc*. Assume these variables are *p1* thru *p4*, from left to right. Assign a value 0 thru 24 for each character A thru Y, in sequence (25 characters are used A - Y, but not Z).
2. Find the total seconds from the beginning of the hour:

```
hunds_seconds_begin_hour = p1+(p2*25)+(p3*625)+(p4*625*25)
```

3. Find minutes, seconds, and hundredths (% = integer division):

```
minutes = hunds_seconds_begin_hour%(60*100)
seconds = (hunds_seconds_begin_hour-minutes*60*100)%100
hundredths = hunds_seconds_begin_hour-minutes*60*100-seconds*100
```

For more information, refer to [Changing DFSMSHsm backup and migration generated data set names to reduce contention for similar names and eliminating a possible performance degradation in z/OS DFSMSHsm Implementation and Customization Guide](#).

- *user1* and *user2* are the first two qualifiers of the data set name.
- *Xydd* is the year and day. X is replaced with a letter that represents the decade as follows:

G = 1970	B = 2020
H = 1980	C = 2030
I = 1990	D = 2040
J = 2000	E = 2050
A = 2010	F = 2060

If, for example, I9012 replaces *Xydd*, then I9012 represents January 12, 1999.

If you need to access a single-file format tape, you must specify a valid data set name for the single file that comprises up to 255 tape volumes. The following migrated data set name is valid for each set of cartridge-type tape volumes in single-file format:

```
prefix.HMIGTAPE.DATASET
```


Only the last 17 characters (.HMIGTAPE.DATASET) are recorded in the tape label. To access the single-file format data set, you must supply any valid *prefix* along with the .HMIGTAPE.DATASET qualifiers.

The following tape data set name for each copy of a cartridge-type tape volume is created by the TAPECOPY command or the duplex tape option:

```
prefix.COPY.HMIGTAPE.DATASET
```

Compaction or zEDC compression during migration

If you specify the compaction during migration option, DFSMSHsm compacts each data set as it migrates. The first time the data set migrates, DFSMSHsm always compacts it. If a migrated data set is recalled and is again a candidate for migration, DFSMSHsm checks the compaction history in the MCDS for that data set. If the compaction from the earlier migration did not result in saving at least the percentage of space that you specified with the COMPACTPERCENT parameter of the SETSYS command, DFSMSHsm does not perform compaction again.

Note: If the data set is a SAM or VSAM compressed data set, DFSMSHsm suspends compaction during migration and records in the MCDS that the data set is striped, compressed, or both.

If the data set is to migrate to an SDSP, the compaction size is always estimated using an internal default of 50%. Previous compaction history is not considered.

Alternately, if you specify zEDC compression instead of compaction (via the ZCOMPRESS parameter of the SETSYS command), zEDC compression will be done instead of compaction. If zEDC Services are *not* available, DFSMSHsm will use the SETSYS COMPACT settings to determine what type of software compaction is used for migration. See [“ZCOMPRESS: Specifying when compression with zEDC should be done” on page 1207](#) for more information.

If the data set is being migrated to a cloud object store defined in the SMS network connection construct that supports transparent cloud tiering compression, then the data set can be compressed by TCT compression if it is not already compressed, encrypted, or compressed and encrypted. If the migrated data set is TCT compressed, then the percentage of space saved for the data set is stored in the MCDS and FSR records. DFSMS selects a TCT compression capable primary volume when the TCT compressed migrated data set is recalled.

If the data set is being migrated to a DIRECT cloud provider defined with a CDA provider file and SETSYS ZCOMPRESS(ALL | CLOUDMIGRATE(YES)) has been specified, DFSMSHsm will request CDA compression to be used during migration.

Space management of VSAM data sets

DFSMSHsm migration functions that exist for non-VSAM data sets also exist for VSAM data sets that are listed in the integrated catalog facility catalog.

The migration of a VSAM data set is performed only on a complete VSAM sphere. The base cluster and all alternate index clusters defined on the base cluster are migrated as part of the data set and are not specified separately. Paths are included automatically in the data set migration command.

When DFSMSHsm migrates a VSAM data set, it tracks a maximum of eight component names for that data set. See [Figure 10 on page 76](#), which shows component name limitations when DFSMSHsm is recalling a migrated VSAM data set.

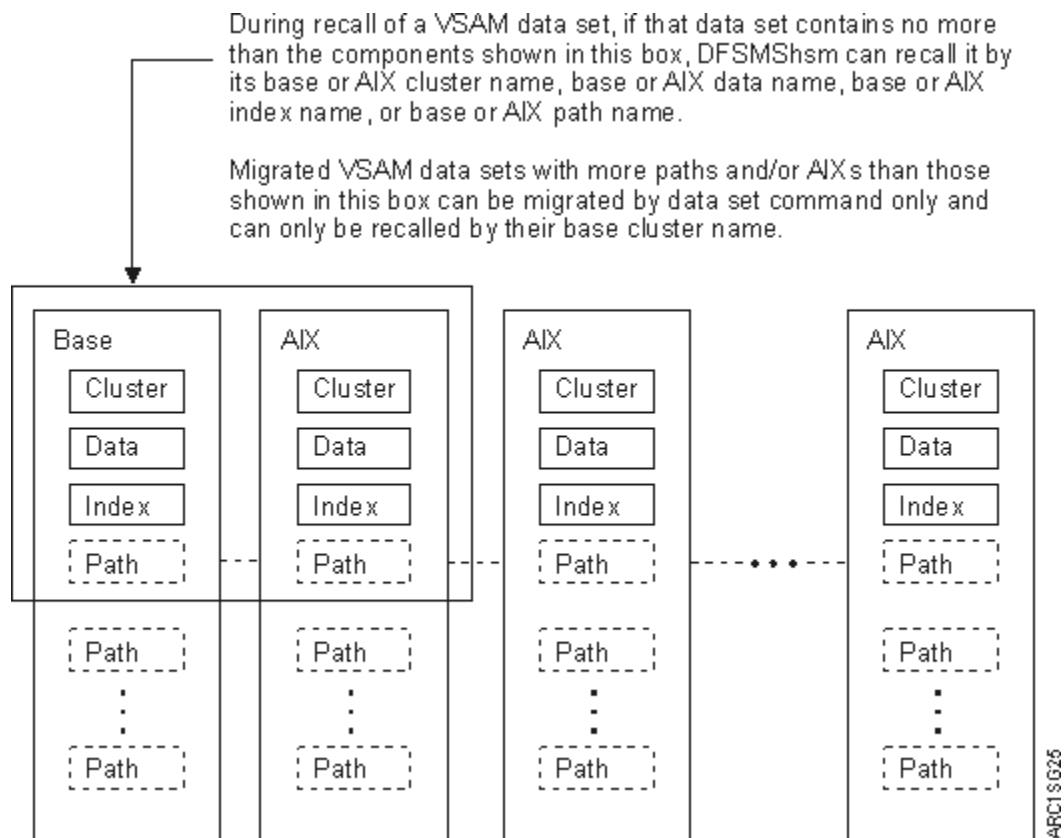


Figure 10. Component Name Limitations when Recalling Migrated VSAM Data Sets

DFSMSHsm can migrate multiple-volume SMS-managed VSAM data sets when it processes the volume containing the base data component. DFSMSHsm does not migrate a sphere that has a component that needs to be verified.

Data sets cataloged in the integrated catalog facility catalog with more than one alternate index (AIX), more than one path on the base, or more than one path defined on the AIX are not selected for migration during volume processing. These data sets can be migrated by data set command.

DFSMSHsm does not migrate a VSAM data set marked for forward recovery, or one with retained locks.

Size eligibility for data sets to migrate to an SDSP

Data sets that meet certain size requirements can be migrated to SDSP data sets. Data set size is determined as follows:

- **Physical sequential data sets**—by the high-used track, which is stored in the data set VTOC entry
- **Physical sequential data sets with zero secondary allocation**—by allocated size
- **Partitioned data sets (not PDSE)**—by the high-used track, which can include unused space within the data set

Partitioned data sets can only migrate to SDSPs when the following two conditions are met: (1) the migration request is for a volume (not for a data set) and (2) an AX cell exists in the catalog for the PDS. In these cases, eligible PDSs are migrated to SDSPs using DFSMSdss as the data mover.

- **Direct access data sets**—by allocated size
- **VSAM data sets**—by allocated size address (RBA)
- **PDSE data sets**—by the high-used page, which is obtained from DFSMSdftp
- **HFS data sets**—by the high-used page, which is obtained from DFSMSdftp
- **Unmounted zFS data sets**—by allocated size address (RBA)

A candidate data set is eligible for migration to an SDSP data set if its estimated size is less than or equal to the number of KB or 3380 tracks that you specify with the `SMALLDATASETPACKING` parameter of the `SETSYS` command. (See [“Specifying small data set packing”](#) on page 52 for an explanation of small data set packing.) A data set’s source size in KB in DFSMSHsm is calculated as the number of bytes per track for the device type times the number of tracks used by the data set. For each device type in DFSMSHsm the number of bytes per track is as follows:

Device Type	Bytes per Track
3380	47,476
3390	56,664

Note: If you specify the size eligibility in tracks, DFSMSHsm converts the tracks value to KB with an assumed track size of a 3380 device type.

SDSP selection

When a user data set meets the criteria to be migrated to an SDSP data set, DFSMSHsm selects the SDSP data set with the most free space. When an SDSP is allocated and opened, it remains open until the DFSMSHsm-managed volume has been completely processed, the SDSP goes unused for 30 seconds, or if the SDSP is needed for recall or aggregate backup.

When a data set fails to migrate because it will not fit into the selected SDSP, DFSMSHsm selects a different SDSP when it migrates the next data set to an SDSP. You should consider two error conditions:

- If no SDSP is available to a DFSMSHsm host (they have all become full), the data set migrates, but not into an SDSP data set. To avoid this type of migration, define more SDSP data sets than the maximum number of concurrent migration tasks. Because each ML1 volume has no more than one SDSP data set, you must have more ML1 volumes available than the maximum number of migration tasks.
- Do not run automatic secondary space management while you are running automatic primary space management.
- If all SDSP data sets are busy, the migration of the data set fails.

Out-of-space condition during migration

For each data set that migrates to an ML1 volume and not into an SDSP data set, DFSMSHsm chooses the ML1 volume with the lowest activity or, as a secondary consideration, the most available space. An out-of-space condition on the ML1 volumes never causes immediate migration from an ML1 volume to an ML2 volume. An out-of-space condition occurs when DFSMSHsm:

- Performs migration on a volume and no space exists on any ML1 volume
- Tries to allocate the migration copy and an out-of-space failure occurs

To prevent a large amount of unproductive processing when there is little free space on any ML1, the migration of data sets—caused by an out-of-space failure—retries to the ML1 volume having the most available free space. (The activity to the volume is not considered.) If a second out-of-space condition occurs while targeting the ML1 volume that has the most free space, DFSMSHsm does the following:

- If the failing data set is small (less than five tracks), and if this is the first time during this processing that a small data set did not fit on an ML1 volume, the task that encountered the out-of-space condition determines the free space on all ML1 volumes, and the migration of the data set is retried. If the task encounters the out-of-space condition again, the free space determination is not done. In this case, the volume migration is terminated and, if ML1 volumes are the only target devices for the volume migration, migration is held. If tape is being used, that part continues.
- If the data set is not small (greater than or equal to five tracks), DFSMSHsm fails the migration for the data set but continues migration with the next data set on the source volume.

The previous processing does not occur if the out-of-space condition is caused by a full VTOC for a volume. In this case, DFSMSHsm marks the ML1 volume full and tries another.

Catalog considerations for migration

DFSMSHsm migrates only cataloged data sets. When a data set migrates, the volume serial number of the data set is replaced with MIGRAT in the catalog. DFSMSHsm uses the volume serial number of MIGRAT to specify that a data set has migrated. Therefore, an installation must not have any volume with a volume serial number of MIGRAT. If it has, data access failures could result.

DFSMSHsm does not recognize JOBCAT and STEPCAT DD statements, so any data sets that require JOBCAT or STEPCAT DD statements appear as uncataloged to DFSMSHsm. Because DFSMSHsm does not recognize JOBCAT or STEPCAT DD statements, DFSMSHsm uses only catalogs available through the standard catalog search. Multilevel alias support is part of the standard catalog search.

Maintenance and use of the data set date-last-referenced field for space management

DFSMSHsm initializes the date-last-referenced field of supported data sets in the data set VTOC entry when the volume is first processed for volume space management. When a data set is opened after that, the open routine writes the current date in the date-last-referenced field. DFSMSHsm refers to the date-last-referenced field and the PRIMARY DAYS NON-USAGE and LEVEL 1 DAYS NON-USAGE attributes to determine if a data set is eligible for space management. Since the date-last-referenced is set when a data set is opened, a data set that is open for several days can be eligible for migration shortly after it is closed. Allow extra days in the management class parameters for this situation.

If the date-last-referenced field is zero during volume space management, DFSMSHsm places the current date into the field. Zeros in the date-last-referenced field mean that volume space management has never processed the data set.

For the majority of ways that DFSMSHsm recalls can be initiated, the recall is considered a reference to the data set, so the date-last-referenced field of the data set is set to the current date. As a result, you can schedule work by recalling data sets that will be required during a later work session. DFSMSHsm normally does not automatically migrate these recalled data sets before you can use them.

Certain DFSMSHsm activities produce recalls for which the date-last-referenced field is not set to the current date. The date-last-referenced field is not set to the current date during recalls for:

- Certain indirect references causing a recall, such as the MODEL parameter of the IDCAMS DEFINE command, or a catalog access done by CICS® using the ALTER command
- Renaming data sets
- Extent reduction
- Conversion of data sets from one level 0 volume to another level 0 volume

Instead, the date-last-referenced field (DS1REFD) is set to the date-last-referenced value when the data set was originally migrated.

The date-last-referenced field is not updated when a data set is moved as a result of a class transition.

Writing to a cartridge-type, single-file-format tape during migration

During volume migration, DFSMSHsm opens the output data set when it processes the first eligible user data set. DFSMSHsm writes the migration copy and then updates the control data set records. After current processing has been completed, the tape is already positioned to receive the next migrated data set. If a failure occurs, DFSMSHsm marks the cartridge-type tape volume full and selects another tape volume. Migration processing continues.

When the volume migration task has no more volumes to process, the DFSMSHsm tape data set is closed and the output tapes are deallocated.

If an I/O error occurs on the source data but does not prevent DFSMSHsm from using the cartridge-type tape volume, DFSMSHsm adds a data set entry to the TTOC for the failing data set and immediately invalidates the entry.

When end-of-volume is reached while DFSMSHsm is writing a 16KB block on a cartridge-type, single-file-format tape volume, DFSMSHsm does normal end-of-volume processing.

When individual user data sets migrate to a cartridge-type tape volume, DFSMSHsm closes the DFSMSHsm tape data set only after all individual data set requests on the migration queue have been processed.

End of level 0 volume processing

At the end of space management of each level 0 volume, DFSMSHsm determines the amount of free space remaining on the level 0 volume just processed and the volume's fragmentation. At this time DFSMSHsm also passes control to the space management volume exit (ARCMVEXT) if it is turned on.

At the end of processing a level 0 volume, DFSMSHsm writes messages ARC0519I and ARC0523I to the migration activity log. The messages record:

- The number of tracks freed by partial release of over-allocated space
- The number of data sets migrated and deleted
- The number of tracks freed by migration
- The time

Related reading

For more information about ARCMVEXT, see [z/OS DFSMS Installation Exits](#).

Pass 2 processing example

[Figure 11 on page 80](#) shows the results of pass 2 processing. It is assumed that the low thresholds for the volumes have not been reached by pass 1 processing. Compare [Figure 9 on page 70](#) and [Figure 11 on page 80](#).

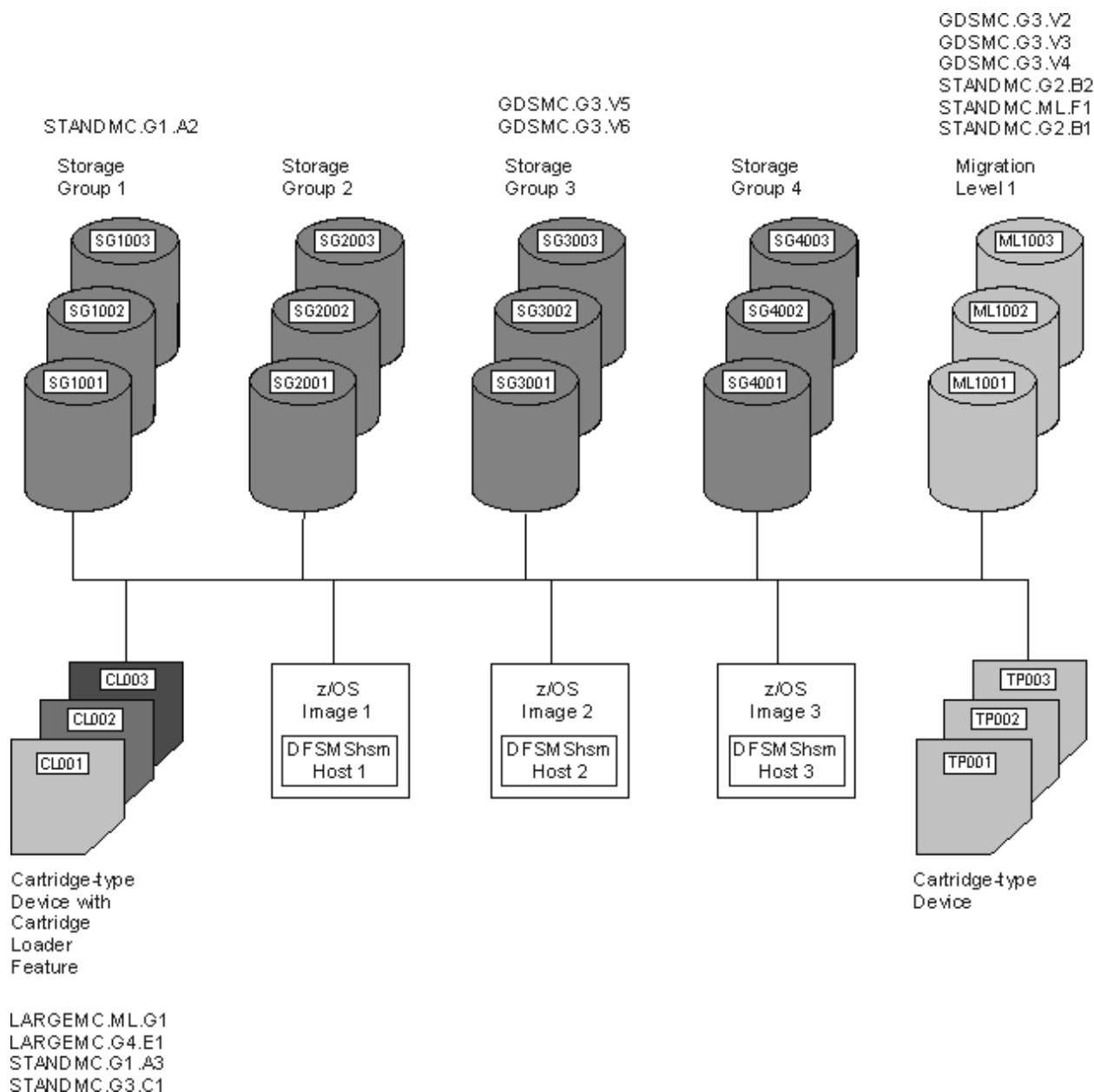


Figure 11. Example System of SMS-Managed Volumes after DFSMSHsm-Managed Volume Management Pass 2

First look at storage group 1. Because data set STANDMC.G1.A2 has not passed the value for PRIMARY DAYS NON-USAGE (see Table 4 on page 57), it does not migrate.

Now look at storage group 2. Because both STANDMC.G2.B1 and STANDMC.G2.B2 exceed the PRIMARY DAYS NON-USAGE value, both migrate.

Now look at storage group 3. Data sets GDSMC.G3.V3 and GDSMC.G3.V2 migrate first because they exceed the value in the # GDG ELEMENTS ON PRIMARY specification for their management class. Because both are candidates for early migration, either one can be the first one migrated. Next, STANDMC.G3.C1 migrates because its size gives it a priority over GDSMC.G3.V4. Notice that because the unused age of STANDMC.G3.C1 is greater than the value that is specified for LEVEL 1 DAYS NON-USAGE, the data set migrates directly to level 2. Last, GDSMC.G3.V4 migrates because it has exceeded the value for PRIMARY DAYS NON-USAGE.

Finally, look at storage group 4. Data set LARGEMC.G4.E1 migrates because it has passed the PRIMARY DAYS NON-USAGE value. Because the LEVEL 1 DAYS NON-USAGE value for its management class is 0, LARGEMC.G4.E1 migrates to ML2 tape.

You can also compare [Figure 11 on page 80](#) with [Figure 6 on page 57](#) to see how many data sets have been removed or migrated from the DFSMSHsm-managed volumes.

Event-driven migration example scenario

In the following event-driven migration example scenario, there are three DFSMSHsm hosts which manage three storage groups. A portion of the ARCCMDxx parmlib member of SYS1.PARMLIB is shown for each host. The basic storage group definitions are shown for each storage group.

DFSMSHsm host 1:

```

:
/* Start secondary space management at 12:30 AM. */
SETSYS SECONDARYSPMGMTSTART(0030 0200)
DEFINE SECONDARYSPMGMTCYCLE(YNNNYNN -
    CYCLESTARTDATE(2011/09/12))
:
/* Do not perform event-driven migration. */
SETSYS NOINTERVALMIGRATION
SETSYS ONDEMANDMIGRATION(N)
:

```

DFSMSHsm host 2:

```

:
/* Start primary space management at 2 AM. */
SETSYS PRIMARYSPMGMTSTART(0200 0400)
DEFINE PRIMARYSPMGMTCYCLE(Y -
    CYCLESTARTDATE(2011/09/12))
:
/* Perform event-driven migration (on-demand). */
SETSYS NOINTERVALMIGRATION /* Ignored when SETSYS ODM(Y) is specified. */
SETSYS ONDEMANDMIGRATION(Y)
:

```

DFSMSHsm host 3:

```

:
/* Perform event-driven migration (on-demand). */
SETSYS NOINTERVALMIGRATION /* Ignored when SETSYS ODM(Y) is specified. */
SETSYS ONDEMANDMIGRATION(Y)
:

```

The storage groups are defined:

```

NAME ==> SG1
MIGRATE SYSTEM/SYSPLEX NAME ==>
AUTO MIGRATE ==> Y
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ==> 85 LOW ==> 70
:
NAME ==> SG2
MIGRATE SYSTEM/SYSPLEX NAME ==>
AUTO MIGRATE ==> Y
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ==> 95 LOW ==> 80
:
NAME ==> SG3
MIGRATE SYSTEM/SYSPLEX NAME ==>
AUTO MIGRATE ==> N
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ==> 20 LOW ==> 0

```

Immediately after SMS issues an ENF72 event code signaling a volume in storage group SG1 or SG2 has exceeded the high threshold (85% for SG1 and 95% for SG2), DFSMSHsm host 2 and DFSMSHsm host 3 perform on-demand migration (immediate space management) on eligible volumes that belong to SG1 and SG2. DFSMSHsm host 2 and DFSMSHsm host 3 perform on-demand migration on eligible volumes in these storage groups because the SETSYS ONDEMANDMIGRATION(Y) setting is specified and the storage groups are defined with AM=Y (Auto Migrate, yes). On-demand migration is performed on SG1 and SG2 as needed throughout the day on both hosts, except when primary space management is running between 2:00 AM and 4:00 AM on DFSMSHsm host 2. During this time, DFSMSHsm host 3 is available to perform

on-demand migration and on-demand migration is paused on DFSMSHsm host 2. After primary space management completes, DFSMSHsm host 2 is available to perform on-demand migration again.

DFSMSHsm host 1 is reserved for handling high priority DFSMSHsm requests. The only space management task DFSMSHsm host 1 performs is secondary space management between 12:30 AM and 2:00 AM. DFSMSHsm host 1 does not perform interval migration because the SETSYS NOINTERVALMIGRATION setting is specified. Additionally, DFSMSHsm host 1 does not perform on-demand migration because the SETSYS ONDEMANDMIGRATION(N) setting is specified.

None of the DFSMSHsm hosts perform event-driven migration on volumes that belong to storage group SG3 because it is defined with AM=N (Auto Migrate, no).

All eligible volumes are processed by event-driven migration until the low threshold is reached or until there are no eligible data sets to migrate.

Related reading

- For more information about primary space management, secondary space management, or event-driven migration, see [“Automatic primary space management” on page 23](#), [“Automatic secondary space management” on page 24](#), or [“Event-driven migration” on page 24](#).
- For more information about interval migration or on-demand migration, see [“Using interval migration” on page 25](#) or [“Using on-demand migration” on page 27](#).
- For more information about the SETSYS commands used, see Chapter 65, [“SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#).
- For more information about managing storage and space requirements, see [“Specifying the storage to be managed and the space requirements” on page 32](#).
- For more information about storage groups and management classes, see [z/OS DFSMSdfp Storage Administration](#).

Class transitions for Db2 objects example scenario

As Db2 objects age, it is desirable to keep the objects online in order to maintain immediate access, but it is also desirable to minimize the cost of maintaining them online. To achieve this, a class transition environment is configured such that the Db2 objects are allocated and maintained on standard DASD for 90 days. After this time, they are transitioned to less expensive SATA DASD. Transitioning the objects to SATA DASD enables the objects to be kept online at a lower cost.

If any object in general remains unreferenced for 95 days, it will become eligible for migration to ML2 tape. Because the Db2 objects are expected to always be allocated and open after 90 days, the SERIALIZATION EXIT will therefore indicate that Db2 should be invoked to close and reopen the objects around the time of this transition. To minimize the period in which these objects are closed to Db2, fast replication will be the preferred data movement technique. The storage group containing the SATA DASD will be given a higher processing priority to help ensure that any objects eligible for migration are migrated off of the SATA DASD to maximize the available space for the objects that will transition to the SATA DASD from the standard DASD.

The storage groups are defined:

```
NAME ===> STANDARD
MIGRATE SYSTEM/SYSPLEX NAME ===>
AUTO MIGRATE ===> P
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ===> 85 LOW ===> 20

:
PROCESSING PRIORITY ===> 50

NAME ===> SATA
MIGRATE SYSTEM/SYSPLEX NAME ===>
AUTO MIGRATE ===> P
MIGRATION AND ALLOCATION THRESHOLDS: HIGH ===> 95 LOW ===> 1

PROCESSING PRIORITY ===> 75
```


The management class is defined:

```
NAME ==> DB2TRANS
      TIME SINCE CREATION      ==> 90
      SERIALIZATION ERROR EXIT ==> DB2
      TRANSITION COPY TECHNIQUE ==> FRP
      PRIMARY DAYS NON-USAGE    ==> 95
      LEVEL 1 DAYS NON-USAGE    ==> 0
      COMMAND OR AUTO MIGRATE   ==> AUTO
      EXPIRE AFTER DATE/DAYS    ==> NOLIMIT
```

During primary space management, the volumes within storage group SATA will be processed first. Any objects that have been unreferenced for more than 95 days will be migrated to tape. After all of the volumes within storage group SATA have been processed, the volumes assigned to storage group STANDARD will be processed. Any object that is 90 days or older will be transitioned to a volume assigned to the SATA storage group. Fast replication will be the preferred data movement technique.

Automatic recall

Recall occurs automatically when a data set is referred to by a TSO user, JCL, or a program. The reference to the data set may cause a request that DFSMSHsm recall the data set either to a specific volume or to any volume. Unlike non-SMS-managed data, no special considerations apply for SMS-managed data sets between JES2 and JES3 systems. When DFSMSHsm receives the request to recall the data set, it passes the following information to the automatic class selection (ACS) routine of DFSMSdfp:

- The name of the data set.
- The name of the storage class associated with the data set when the data set migrated. If the data set was not SMS-managed when it migrated, no storage class is passed.
- The name of the management class associated with the data set when the data set migrated. If the data set used the default management class or was not SMS-managed when it migrated, no management class is passed.
- The name of the data class associated with the data set when the data set migrated. If the data set had no data class or was not SMS-managed when it migrated, no data class is passed.
- The volume serial number and unit type of the volume where the data set originally resided or of the volume specified by the request.

Note: If your storage class ACS routine checks for a valid DASD unit name, be sure to include any unit names that may have existed in your installation at one time and are no longer in use, because the data set could have resided on such a unit when it was migrated.

- If the data set was never transitioned and did not miss a transition while it was migrated, an environment indicator of RECALL.
- If the data set was transitioned before the migration and the Last Successful Class Transition Date (LSCTD) is set, or if one or more class transitions were missed while the data set was migrated, then an environment indicator of SPMGCLTR.

Passing the SPMGCLTR indicator ensures that the data set is recalled to the same storage class and storage group as when it transitioned. If the data set missed multiple transitions while it was migrated, then the ACS routines will be invoked for each transition that was missed.

Note: DFSMSHsm will check to see if a class transition was missed as many as five times. DFSMSHsm will limit the number of repeated ACS routine invocations to avoid circularity.

The Last Successful Class Transition Date (LSCTD) field is set and used by DFSMSHsm to manage class transitions. DFSMSHsm will use this field to determine if a data set has already been transitioned and should not be reprocessed, as shown in [Table 6 on page 84](#). The LSCTD can be displayed for a particular data set with the LISTCAT command, if it exists and the value is non-zero, and DCOLLECT.

<i>Table 6. Last Successful Class Transition Date (LSCTD) setting rules during class transitions and recalls</i>	
Action	LSCTD value
Management class is changed during class transition	The LSCTD is set to zero
Management class is not changed during class transition	The LSCTD is set to the current date
Management class is changed during recall when one or more class transition were missed while the data set was migrated	The LSCTD is set to zero
Management class is not changed during recall when one or more class transition were missed while the data set was migrated	The LSCTD is set to the current date
Management class is not changed during recall when the data set was transitioned before migration and the LSCTD was set	The LSCTD is set to the current date

- The RACF user ID and group ID for non-DFSMSHsm-authorized users.
- The data set size in KB.
- The data set expiration date.
- A generation data group indication.
- The data set organization.
- The record organization for VSAM data sets.

If SMS is installed and active, the ACS routine may return to DFSMSHsm with a storage class and, optionally, a management class. If the ACS routine returns a storage class, DFSMSdfp selects a volume. The volume selected is not necessarily the volume that was specified by the recall request. The catalog is updated to reflect the current status.

If SMS is installed and active but the ACS routine does not return a storage class, DFSMSHsm processes the data set as a non-SMS-managed data set. If the ACS routine does not return a storage class for a data set that was extended format when it was migrated, DFSMSHsm fails the recall.

If SMS is not installed and active and the data set was not SMS managed when it migrated, DFSMSHsm processes the data set as a non-SMS-managed data set.

If SMS is not installed and active and the data set was SMS managed when it migrated, DFSMSHsm fails the recall. If the data set migrated as an SMS-managed data set, DFSMSHsm assumes that the data set is still SMS managed and cannot be processed without SMS.

Table 7 on page 84 summarizes the recall results.

Table 7. Recall Destination Summary

SMS Active	Storage Class Returned	Data Set Migrated As	Data Set Recalled As
Yes	Yes	Either	SMS-managed
Yes	No	Either	Not SMS-managed
No	Not applicable	SMS-managed	Failed
No	Not applicable	Not SMS-managed	Not SMS-managed

Note: For a general discussion of recall requests with regard to the common recall queue, see [“Common recall queue”](#) on page 129.

Related reading

For more information about recalling non-SMS-managed data sets, see [“Recall” on page 119](#).

Automatic recall with DFSMSdss

When DFSMSdss is used as the data mover, DFSMSHsm determines if the data set is SMS managed as described under [“Automatic recall” on page 83](#). However, DFSMSHsm allows DFSMSdss to determine which volumes the data set is recalled to.

If a multivolume data set is no longer SMS managed, DFSMSHsm attempts to recall the data set. A list of up to five volumes is passed to DFSMSdss. If space permits, the data set is recalled to a single volume. Otherwise, it is recalled as a multivolume data set.

During recall processing with DFSMSdss as the data mover, if a data set was never transitioned and did not miss a transition while it was migrated, then the &ACSENVIR field contains RECALL for the storage class and management class ACS routines, and RECOVER for the storage group ACS routine. If the data set was transitioned before the migration or if one or more class transitions were missed while the data set was migrated, then the &ACSENVIR field contains SPMGCLTR for all ACS routines. &ACSENVIR is a read-only variable that indicates the environment in which the ACS routine is invoked.

Recalling extent-reduced data sets

When a data set migrates for extent reduction, DFSMSHsm immediately requests a recall for that data set to the same volume. Under these circumstances, DFSMSHsm recalls the data set to the specified volume with the same storage class, management class, and data class as were in effect when the data set migrated. DFSMSHsm does not invoke the ACS routine for these data sets.

Note: DFSMSHsm can be directed to use the ACS routines for volume selection by using the Return-Priority exit (ARCRPEXT). For more information about the ARCRPEXT exit, see [z/OS DFSMS Installation Exits](#).

Automatic recall during data set allocation

When the allocation process is trying to find a data set, DFSMSHsm intercepts the locate and tries to recall the data set if the data set is migrated.

Automatic recall for command deletion of VSAM AIX clusters

When a command is issued to delete an AIX cluster, the sphere may have been migrated. DFSMSHsm intercepts the deletion request, and if the sphere is migrated, DFSMSHsm recalls the data set.

Automatic recall for data set renaming

When a rename or alter request is issued for a migrated data set, DFSMSHsm first recalls the data set and then lets the rename or alter process continue, with the exception of an IDCAMS ALTER command issued to change the storage class or management class, or both, of a migrated data set. When an IDCAMS ALTER command is issued to change either the storage or management classes, or both, of a migrated data set, the change is made without recalling the data set. Both DFSMSHsm's and catalog's records are updated to reflect the change.

When data sets are recalled for a rename or alter request, DFSMSHsm does the following:

- Sets the last backup date to zero
- Sets the data-set-changed indicator in the data set VTOC entry to On
- Sets the date-last-referenced field to the date that the data set had when it migrated

The results of the preceding actions are as follows:

- The renamed data set is backed up the next time that backup runs.

- The renamed data set probably will migrate soon because the date-last-referenced field indicates a long period since the data set was opened.

Automatic recall during catalog updates

A data set, either VSAM or non-VSAM, could be on a migration volume when it is about to have its catalog entry altered. If the catalog routine is requested to UNCATALOG, RECATALOG, DELETE/NOSCRATCH, or ALTER the catalog entry of a data set on a volume with a volume serial of MIGRAT, DFSMSHsm first recalls the data set and then lets the catalog routine continue. In this case, DFSMSHsm handles the volume selection for the recall request in the same manner as it handles the allocation recall requests. It does not recall the data set to a specific volume.

An IDCAMS ALTER command issued to change the storage class or management class, or both, of a migrated data set is an exception. When an IDCAMS ALTER command is issued to change either the storage or management classes, or both, of a migrated data set, the change is made without recalling the data set. Both DFSMSHsm's and catalog's records are updated to reflect the change.

In a non-RACF environment, this function is bypassed, which permits any user to perform catalog maintenance without recalling data sets. In a RACF environment, this function is bypassed for UNCATALOG, RECATALOG, and DELETE/NOSCRATCH if the current RACF logon group is ARCCATGP, which permits only authorized users to perform these requests without recalling data sets.

Note: Automatic recall is bypassed if DFSMSHsm receives the DELETE command before any other command. In cases where another component performs an action that invokes DFSMSHsm before the DELETE command, an automatic recall occurs. For example, issuing a DELETE command on a migrated data set using some TSO options causes DFSMSHsm to recall that data set before processing the DELETE command.

The access methods services REPRO/MERGECA and CONVERT CATALOG commands do not cause a data set recall, because the catalog entries are not being altered.

A recall may be retried multiple times and lengthen processing time if the necessary resources are not available.

Related reading

For more information about performing RACF catalog maintenance, see [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Additional recall considerations

When DFSMSHsm recalls data sets from tape ML2 volumes, the tape remains allocated and mounted until DFSMSHsm has processed all recall requests requiring only that volume, or a TAPERECALLLIMITS parameter has been met. For tape devices before IBM 3480s, the additional requests are processed by file sequence order. For 3480 (single-file-format tape cartridge) and later tape devices, the additional requests are processed by priority. When the recall queue no longer has any recall requests requiring that mounted tape volume, DFSMSHsm sets a short timer and waits for another request. If a request needing that single tape arrives, it is started immediately. If the timer expires, DFSMSHsm deallocates and demounts the tape volume.

Note: In a common recall queue environment, a single tape mount processes all outstanding requests from all DFSMSHsm hosts that require the same tape.

To manage your space better and to reduce job-failure exposures, do not specify volume and unit information in JCL to refer to non-VSAM and VSAM data sets on DFSMSHsm-managed volumes. If a non-SMS-managed volume is specified in the JCL, DFSMSdfp and DFSMSHsm cannot cooperate in choosing the target volume for the recall. If the data set is non-SMS-managed, the specified non-SMS-managed volume is used as the target of the recall. If the data set is SMS-managed, the recall fails.

Relying on the standard catalog search allows DFSMSHsm, DFSMSdfp, and DFSMSdss if specified, to cooperate in choosing the volume and recalling the data set during allocation processing.

The SETSYS VOLCOUNT(NONE | ANY) command affects how the allocation of SMS-managed data sets are performed when DFSMSdss is the datamover. The environment established by the SETSYS VOLCOUNT command can be overridden on command recalls by specifying the DFDSSOPTION(VOLCOUNT(N(nn) | ANY)) option of the RECALL command.

Reblocking during recall

The purpose of reblocking is to make the most efficient use of available space on your DASD. Physical sequential data sets are moved with DFSMSdss. Specifying SETSYS CONVERSION(REBLOCKTOANY) will cause them to be reblocked during recall.

Nonkeyed, blocked, physical sequential data sets that were migrated with DFHSM 2.6.0 (or an earlier release) are also reblocked during recall when SETSYS CONVERSION(REBLOCKTOANY) is used. Partitioned data sets (DSORG=PO) are not reblocked during recall.

Marking data sets eligible for fast subsequent migration during recall

During recall of a data set, if the data set satisfies fast subsequent migration eligibility requirements, the data set is marked eligible for fast subsequent migration (that is, it is marked for reconnection to the ML2 tape from which it was recalled). If a data set is not marked eligible for fast subsequent migration, it will not be considered for reconnection during future migration processing.

Related reading

- For more information about fast subsequent migration, see [“Reconnecting data sets using fast subsequent migration”](#) on page 94.
- For a list of fast subsequent migration requirements, see [Fast subsequent migration \(reconnection\) eligibility requirements](#).

Additional space management during migration and recall when DFSMSHsm is the data mover

Note: Although DFSMSdss is the data mover for most data set types, the following topic refers to exception cases for which DFSMSHsm is the data mover, including most PDS data sets, as well as special cases of PS, VSAM, and other organizations.

When DFSMSHsm is used as the data mover during the migrating and recalling of PS, PDS, and VSAM data sets, it can automatically move a data set to a different device type. Thus, for these data sets, the requested space is always converted by DFSMSHsm to a space of approximately equivalent capacity when a device-type change occurs. Also, the original allocation type (that is, the cylinder, track, or average block) and the CONTIG option are maintained.

When DFSMSHsm recalls a migrated PS or PDS data set, it requests a primary allocation size approximately equal to the space used (all extents) by the original data set, unless the secondary allocation size is zero. When the secondary allocation size of the original data set is zero, DFSMSHsm requests a primary allocation size approximately equal to the space allocated (all extents) at the time of migration. For secondary allocation size, DFSMSHsm always keeps approximately that much of the original data set space. The option to release unused space is requested when a nonzero secondary allocation size is specified; release of unused space is not requested when secondary allocation is zero. For PDS data sets, the original number of directory blocks is retained.

For PDS data sets, the migrate and recall results in the freeing of embedded free space (compression), and the user information in the data-set-directory user field is retained.

When the DAOPTION is not used, the allocation for recall of migrated BDAM data sets differs from that of PS and PDS data sets. The primary allocation contains the same number of tracks (including all extents) allocated to the original data set at the time of migration. DFSMSHsm maintains the secondary allocation size in a similar manner; that is, the number of tracks is at least as large as that of the secondary allocation size of the original data set. When DFSMSHsm recalls a BDAM data set, it does not select a device that has a different track length than that of the device from which the data set was migrated.

However, you can direct the recall of a BDAM data set to a device that has a different track length. When DFSMSHsm moves BDAM data sets, it moves one input track of data to one output track of data. The data from an original track must fit on a track of the output device. If you direct the recall of a BDAM data set to a particular device, you are responsible for ensuring that the data is accessible in this form on that device. Release of unused space is never requested for BDAM data sets.

The recall of VSAM data sets includes translating space allocation when device types are changed and retaining the type of allocation (cylinder, track, or average block) changes. DFSMSHsm does not do any extent consolidation for VSAM data sets. The recall data set is processed in accordance with your original DEFINE request, which can cause the allocated space to grow if you request distributed free space.

Recommendation:: You should have a nonzero secondary allocation size if you request distributed free space.

The migration and recall of non-VSAM data sets normally results in a reduction of the number of extents for multiple-extent data sets, because DFSMSHsm's primary allocation size is large enough to contain the entire data set in one extent. If DFSMSHsm cannot find such an extent and the CONTIG option is not specified, up to five non-contiguous extents can be used. If DFSMSHsm cannot find such an extent and CONTIG is specified, the request fails.

Related reading

- For more information about data mover support for DFSMSHsm space management functions, see data mover support table in the topic about user data sets in [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about how DFSMSdss moves PS, PDS, and VSAM data sets, see [z/OS DFSMSdss Storage Administration](#).

Command space management tasks

In addition to its automatic volume space management functions for SMS-managed volumes, DFSMSHsm provides you with the ability, by issuing individual commands, to perform space management functions.

Deleting migrated data sets

To delete migrated SMS-managed data sets, you can use one of the following methods:

- DFSMSHsm DELETE command
- Access Method Services DELETE command
- TSO DELETE command
- IEHPROGM utility

Table 8 on page 89 shows what happens when you specify one of these commands or use the IEHPROGM utility to delete a migrated data set.

Table 8. Commands for Deleting Migrated Data Sets

Command or Utility	Result
DFSMSHsm DELETE, Access Method Services DELETE, TSO DELETE or IEHPROGM utility SCRATCH function for SMS-managed data set	<p>Scratches a data set on a DASD migration volume without recalling the data set or marks the data set not valid in the TTOC for any tape ML2 volumes that contain the data set. DFSMSHsm then uncatalogs the data set. DFSMSHsm deletes all control data set records related to the migration copy. If the data set is migrated to an SDSP, the MCD is updated to indicate that the migration copy needs to be scratched. If a discrete RACF data set profile exists, DFSMSHsm deletes it.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. If the optional parameter NON-VSAM is specified on the Access Method Services DELETE command, the migrated data set is recalled before deletion. 2. If DFSMSHsm is down when you specify the optional parameters GDG FORCE on the Access Method Services DELETE command, IDCAMS does not invoke DFSMSHsm. Therefore, you must then issue the DFSMSHsm DELETE command for each of the now-uncataloged migration copies.
IEHPROGM utility SCRATCH function for non-SMS-managed data set	<p>Scratches a migrated data set if:</p> <ul style="list-style-type: none"> • The specified volume is the same one the data set migrated from • The volume is not SMS managed • The data set is not SMS managed <p>DFSMSHsm scratches the migration copy of the data set and deletes the MCDS data set record. The data set remains cataloged.</p>

If the deleted data set has backup versions, DFSMSHsm keeps the backup versions.

If a data set has been deleted by one of the preceding commands and you want to recover a backup version of the data set, see [“Data set recovery and restore” on page 241](#).

Using the DELETE command

You can delete migrated data sets under the following conditions:

- For VSAM data sets, make the name specified in the command the name of the base cluster. Specifying the name of any other component causes the command to fail.
- The name specified for the data set (for either VSAM or non-VSAM data sets) must be the name in the catalog. You cannot specify the name that DFSMSHsm assigned to the migrated data set.

DFSMSHsm fails any DFSMSHsm DELETE command that specifies a member name of a partitioned data set.

Deleting data sets that have not expired

If the data set has an expiration date and the date has not been reached, DFSMSHsm deletes the data set only if the PURGE parameter of the DELETE command is specified. If a data set does not have an expiration date, the PURGE parameter does not apply, and DFSMSHsm deletes the data set.

Migrating an individual data set

You can cause individual SMS-managed data sets to migrate by using the MIGRATE or HMIGRATE command. (The HMIGRATE command is a non-DFSMSHsm-authorized command that is available to

users.) You can migrate a data set by command from either a level 0 volume or an ML1 volume. The migrating data set can move from a level 0 volume to either an ML1 volume or an ML2 volume.

When you issue a MIGRATE (or HMIGRATE) command, DFSMSHsm obtains the definition of the management class associated with the data set from SMS. If the management class definition cannot be obtained, DFSMSHsm issues an error message and does not migrate the data set. If the data set has no management class associated with it, DFSMSHsm uses the default management class.

Data set command migration overrides the management class attributes that control when an SMS-managed data set is eligible to migrate. However, for command data set migration to occur, the COMMAND OR AUTO MIGRATE attribute in the data set's management class must have the value COMMAND or BOTH.

DFSMSHsm may perform fast subsequent migration when it migrates an individual data set. If the fast subsequent migration fails, the data set migrates in the normal manner.

Command migration can occur for either migrated or non-migrated SMS-managed data sets. The data sets can migrate to either ML1 or ML2. Sample commands to migrate data sets are:

```
MIGRATE DATASETNAME(SAMPLE.DATASET)
MIGRATE DATASETNAME(SAMPLE.DATASET1) MIGRATIONLEVEL2
```

Consider the first example. If the data set is on an SMS-managed volume, the LEVEL 1 DAYS NON-USAGE attribute in its management class controls where the data set migrates to. If the value of the LEVEL 1 DAYS NON-USAGE attribute is not zero, the data set migrates to ML1. If the value of the LEVEL 1 DAYS NON-USAGE attribute is zero, the data set migrates to an ML2 tape volume. If the data set has already migrated to an ML1 volume, the data set does not migrate to an ML2 volume because the MIGRATIONLEVEL2 parameter is not specified.

Now consider the second example. If the data set is on an SMS-managed volume or an ML1 volume, it migrates to an ML2 volume no matter what value is specified in the LEVEL 1 DAYS NON-USAGE attribute.

If the target migration device type is tape and the data set needs to be backed up, the data set does not migrate and an error message is issued.

Two installation exits are invoked from the MIGRATE or HMIGRATE commands: the data set migration exit (ARCMDEXT) and the second-level migration data set exit (ARCMMEXT).

DFSMSHsm invokes ARCMDEXT whenever a data set command migration request causes a data set to migrate from a Level 0 volume to a DFSMSHsm Level 1 or Level 2 volume. In addition, an indicator in the ARCMDEXT parameter list indicates a data set that is migrating by a data set command (rather than by volume migration).

DFSMSHsm invokes ARCMMEXT whenever a data set command migration request causes a data set that is already on ML1 to migrate to ML2 (using the ML2 keyword on the data set migration command). In addition, an indicator in the ARCMMEXT parameter list indicates that this data set is migrating by a data set command (rather than by volume migration).

Related reading

For more information about fast subsequent migration, see [“Reconnecting data sets using fast subsequent migration”](#) on page 94.

Transitioning an individual data set

You can drive SMS-managed data sets through class transition processing by using the MIGRATE or HMIGRATE command. When you issue a MIGRATE (or HMIGRATE) command with the TRANSITION keyword, DFSMSHsm invokes the ACS routines with the SPMGCLTR environment variable to determine the new management class, storage class and/or storage group to which the data set should be assigned. This is performed even when the data set is not eligible for a transition based on the class transition criteria for the currently assigned management class. If neither management class, storage class nor storage group changes, then no processing occurs. If only management class changes, then the data set is assigned the management class without any data movement. If either the storage class or storage

group changes, then the data is moved. When data movement is performed, the management class transition copy technique and serialization error exit values are used.

For example, if a Db2 object is assigned a management class with a transition copy technique of PMREQ and serialization error exit value of Db2, and is assigned new storage group AGEDDATA, then a MIGRATE DSNAME(*db2.object*) TRANSITION command results in the object being closed by Db2 (if there are no active transitions), moved to a volume in storage group AGEDDATA using preserve mirror, and reopened by Db2 after the movement has completed.

Note: If a management class transition copy technique other than STANDARD, PRESERVE MIRROR REQUIRED, or FC XRCPRIMARY is specified, then a valid backup copy must exist before the data set is transitioned. This ensures that if an error occurs while physically moving the data through a storage controller function, then a backup is available to recover the data set.

If the transition technique PRESERVE MIRROR REQUIRED is specified to allow FlashCopy to Global Mirror Primaries, a valid backup copy must exist before the data set residing on a Global Mirror primary volume is transitioned. You can set one of the following patches to override this requirement:

```
PATCH .MGCB.+111 BITS(.....1.)
PATCH .MGCB.+111 BITS(.....1)
```

Migrating and transitioning an SMS-managed volume

You can cause data sets to migrate or transition from an SMS-managed volume by using the MIGRATE command. The MIGRATE VOLUME command can override some management class attributes that control when an SMS-managed data set is eligible to migrate or transition, depending on the form of the command. The default is that both migrations and transitions are processed. To process only migrations, specify the MIGRATIONONLY parameter. To process only transitions, specify the TRANSITIONONLY parameter. The MIGRATE VOLUME command migrates data sets for which the management class COMMAND OR AUTO MIGRATE attribute has the value of BOTH only. The MIGRATE VOLUME command transitions data sets that meet the management class class transition attribute.

When you issue a MIGRATE VOLUME command for an SMS-managed volume:

- Migration is from level 0 volumes to ML1 or ML2 volumes only.
- Class transitions are to the volume determined by the ACS routines.
- The volume is considered to be DFSMSshm-managed. DFSMSshm maintains control records for the volume until a DELVOL command is issued for that volume.
- A list of the management class definitions required for processing the SMS-managed data sets on the volume is obtained.

Table 9 on page 91 summarizes the various conditions that can be encountered when obtaining a list of management class definitions and the system action taken as a result.

Table 9. Conditions that can be encountered when obtaining a list of management class definitions during migration of an SMS-managed data set

If...	Then...
the list cannot be obtained.	the command fails and an error message is issued.
the management class definition for a data set cannot be found in the list.	the data set is not processed and an error message is issued.
the data set does not have a management class associated with it.	the default management class is used.

Fast subsequent migration can be performed when a data set is migrated as a result of a MIGRATE VOLUME command. If fast subsequent migration fails, the data set is migrated in the normal method.

Example commands used to migrate data sets SMS-managed volumes

The commands in [Figure 12 on page 92](#) are examples of commands that can be used to migrate data sets from an SMS-managed volume.

```
MIGRATE VOLUME(SG1001)
MIGRATE VOLUME(SG2003 MIGRATE(0))
MIGRATE VOLUME(SG1002) DAYS(0)
```

Figure 12. Example commands used to migrate data sets from SMS-managed volumes

Consider the first the command in the example: `MIGRATE VOLUME(SG1001)`. This command requests DFSMSHsm to process the volume in the same method as described in [“Automatic primary space management” on page 65](#). Both migration and class transitions are processed.

Next, consider the second command in the example: `MIGRATE VOLUME(SG2003 MIGRATE(0))`. This command overrides the PRIMARY DAYS NON-USAGE attribute of the management class so that all data sets on the volume that have the COMMAND OR AUTO MIGRATE attribute value of BOTH migrate. Any data set with a LEVEL 1 DAYS NON-USAGE attribute value of 1–9999 or NOLIMIT migrates to an ML1 volume. Any data set with a LEVEL 1 DAYS NON-USAGE attribute value of 0 migrates to an ML2 tape volume. If an attempt to migrate a data set fails, error message is issued and data set remain on the SMS-managed volume. If a nonzero value or no value is specified for the MIGRATE or DAYS parameter, the command fails and an error message issued.

For the third example, the DAYS(0) parameter behaves differently based on the keywords specified. Transition processing behaves differently than migration processing as follows:

- For migration processing, all data sets that are eligible to migrate are migrated off of the volume, regardless of whether or not they met their age criteria. Conversely, transition processing requires the transition criteria to be met. During transition processing, the low threshold does not stop processing of data sets on the volume. All data sets on the volume are processed for migration and transition eligibility, even after the low threshold has been met.
- If MIGRATIONONLY is specified, processing remains unchanged.
- If BOTH is specified, since all data sets eligible for migration are migrated and migration takes precedence over transitions, essentially all data sets eligible for migration are migrated. Only data sets that cannot migrate but are eligible to be transitioned are transitioned. These data sets must meet the transition eligibility criteria.
- If TRANSITIONONLY is specified, all data sets eligible to be transitioned are processed, regardless of the low threshold on the volume.

Related reading

- For more information about using the MIGRATE command, see [Chapter 55, “MIGRATE command: Requesting a space management function,” on page 955](#).
- For more information about fast subsequent migration, see [“Reconnecting data sets using fast subsequent migration” on page 94](#).

Migrating and transitioning a storage group

The STORAGEGROUP keyword enables migration and transition processing to be performed for each volume defined to a storage group. The `MIGRATE STORAGEGROUP(sgname)` command initiates primary space management processing for a storage group. For more information, refer to [“Automatic primary space management” on page 23](#)

Recalling SMS-managed data sets

You can recall individual SMS-managed data sets from migration volumes to level 0 SMS-managed volumes by using the RECALL and HRECALL commands. (The HRECALL command is a non-DFSMSHsm-authorized command that is available to users.)

When you issue a RECALL or HRECALL command for an SMS-managed data set, processing is the same as that for automatic recall (see [“Automatic recall”](#) on page 83).

When a data set is recalled to a non-SMS-managed volume, the volume to which it is recalled is selected as described in [“Recall”](#) on page 119.

If you want to return an SMS-managed data set to a non-SMS-managed volume, you can use the FORCENONSMS parameter of the RECALL command. If you use the FORCENONSMS parameter, the volume to which the data set is recalled is selected as described in [“Recall”](#) on page 119.

Migrated extended format VSAM data sets can be recalled only as extended format VSAM data sets. Migrated extended format sequential access method (SAM) data sets can be recalled as extended format or nonextended format data sets.

If FORCENONSMS is specified on the RECOVER command to force an extended format data set to be recovered or restored to a non-SMS-managed volume, the data set is recovered as a nonextended data set. The maximum size of the non-SMS SAM data set is one volume.

If an out-of-space condition causes the recall of an SMS-managed data set to fail, the RECALL DFDSSOPTION(VOLCOUNT(N(nn) | ANY)) command can be used to ensure the successful recall of the data set. The RECALL DFDSSOPTION(VOLCOUNT(N(nn) | ANY)) command overrides the environment established by the SETSYS VOLCOUNT command. You can use the QUERY SETSYS command to determine the current operating environment.

Recalling direct access data sets

Target volume selection for command recalls of BDAM data sets can be influenced by using the DAOPTION parameter of the RECALL command. This parameter allows you to specify the acceptable target volume device types (in terms of track size) and how the recalled data is accessed (relative track or relative block). For instance, using the RELBLK option of the DAOPTION parameter allows a data set to be recalled to a device type with a smaller track size than the original level 0 volume. In this case, data is moved to “fill out the track” rather than in a track to track image as is the case for the RELATER and SMATTER options.

You can only use the DAOPTION parameter when recalling a data set as non-SMS-managed. You can use the FORCENONSMS parameter to recall a data set that would otherwise be returned as SMS managed.

Moving data sets

Use the MOVE parameter with the MIGRATE command to move individual SMS-managed data sets from one L0 volume to another L0 volume. Specify MOVE with the MIGRATE DATASETNAME, MIGRATE VOLUME or MIGRATE STORAGEGROUP commands. MOVE causes DFSMSHsm to drive the standard allocation ACS routines to determine the volume to which the data set should be moved. DFSMSHsm then performs class transitioning processing for the data set. When VOLUME or STORAGEGROUP processing is being performed, all data sets on each volume are processed, regardless of threshold. When data movement is performed, the management class transition copy technique and serialization error exit values are used.

The MOVE parameter is intended to assist with the process of moving data off of one or volumes onto another. The typical cases for using MOVE are:

- Move data from one storage group to another (the basic case). The steps are as follows:
 1. Update the ACS routines such that new allocations go to the new storage group.
 2. Move existing data from the original storage group to the new storage group. When the move is being done, the ACS routines are already defined to drive the data to the new storage group.
- Move data from one volume to another within the same storage group. Moving within storage groups doesn't require any value here, because the allocation needs to be within the existing storage group. The existing ACS logic is usable because it just selects the same storage group. To prevent any new allocations from going to the source volume, this volume should be put into DISNEW or DISALL state. If allocations to the source volume are not prevented, then the MIGRATE MOVE command may fail if SMS selects one or more volumes on which the data set currently resides.

Note:

1. During MOVE processing, the statistic records (DSR and VSR) are created with the class transition function code.
2. The MOVE processing uses the transition processing messages.
3. The management class Transition Copy Technique setting can be also used to specify copy technique for MOVE processing. Note that if a copy technique other than STANDARD, PRESERVE MIRROR REQUIRED, or FC XRCPRIMARY is specified, then a valid backup copy must exist before the data set is transitioned. This ensures availability of the backup if an error occurs while physically moving.

If the transition technique PRESERVE MIRROR REQUIRED is specified to allow FlashCopy to Global Mirror Primaries, a valid backup copy must exist before the data set residing on a Global Mirror primary volume is transitioned. You can set one of the following patches to override this requirement:

```
PATCH .MGCB.+111 BITS(. . . . .1.)
```

```
PATCH .MGCB.+111 BITS(. . . . .1)
```

4. When moving Db2 objects, DFSMSHsm must be authorized to the Db2 STOP and START commands.

Reconnecting data sets using fast subsequent migration

Using the standard migration and recall method, a data set can be migrated to an ML2 tape volume or cloud container and later be recalled. When the data set is recalled, the migrated copy of the data set remains on the ML2 tape volume or cloud container and is marked *not valid* in the CDS record for the data set. If the recalled data set is later migrated using the standard migration method, it is *moved* to a new ML2 tape volume or cloud container (even if the data set has not changed). This process results in unnecessary data movement and if stored on an ML2 tape volume, causes the original ML2 tape volume to become fragmented which requires the tape to be recycled.

Fast subsequent migration allows an unchanged data set that is recalled from a single ML2 tape volume or cloud container to be *reconnected* to that ML2 tape volume or cloud container during data set or volume migration. Using fast subsequent migration:

- eliminates the unnecessary data movement and tape mounts caused by migrating a data set again that is unchanged.
- involves creating and updating DFSMSHsm control records only.
- reduces the need for recycling the ML2 tape.

Specific fast subsequent migration eligibility criteria must be satisfied both at time of recall and when a fast subsequent migration reconnect is attempted on a data set. See [“Fast subsequent migration \(reconnection\) eligibility requirements”](#) on page 95.

Enabling fast subsequent migration

There are four methods of enabling fast subsequent migration:

- Add the SETSYS TAPEMIGRATION command with the RECONNECT(ALL) or RECONNECT(ML2DIRECTEDONLY) parameter to the ARCCMDxx parmlib member of SYS1.PARMLIB. Then restart DFSMSHsm.
- Issue the SETSYS TAPEMIGRATION command with the RECONNECT(ALL) or RECONNECT(ML2DIRECTEDONLY) parameter after DFSMSHsm is started.
- Add the SETSYS CLOUDMIGRATION command with the RECONNECT(ALL) parameter to the ARCCMDxx parmlib member of SYS1.PARMLIB. Then restart DFSMSHsm.
- Issue the SETSYS CLOUDMIGRATION command with the RECONNECT(ALL) parameter after DFSMSHsm is started.

To use fast subsequent migration after it is enabled, the data set must also satisfy fast subsequent migration eligibility requirements. See [“Fast subsequent migration \(reconnection\) eligibility requirements”](#) on page 95.

Disabling fast subsequent migration

There are four methods of disabling fast subsequent migration:

- Add the SETSYS TAPEMIGRATION command with the RECONNECT(NONE) parameter to the ARCCMDxx parmlib member of SYS1.PARMLIB. Then restart DFSMSHsm.
- Issue the SETSYS TAPEMIGRATION command with the RECONNECT(NONE) parameter after DFSMSHsm is started.
- Add the SETSYS CLOUDMIGRATION command with the RECONNECT(NONE) parameter to the ARCCMDxx parmlib member of SYS1.PARMLIB. Then restart DFSMSHsm.
- Issue the SETSYS CLOUDMIGRATION command with the RECONNECT(NONE) parameter after DFSMSHsm is started.

Fast subsequent migration (reconnection) eligibility requirements

There are two sets of requirements that must be satisfied in order for a data set to be eligible for fast subsequent migration. The first set of requirements must be satisfied when a data set is recalled. The second set of requirements must be satisfied when a fast subsequent migration reconnect is attempted on a data set. In addition to the two sets of eligibility requirements, there are conditions which always prevent fast subsequent migration regardless of whether all other requirements are satisfied.

Conditions that *always prevent* fast subsequent migration are:

- The data set is an HFS file.
- The data set is a VSAM data set with an alternative index (AIX) or path defined.
- The data set has changed since it was recalled.
- The data set spans multiple ML2 tape volumes.
- The MIGRATE command was issued with the CONVERT parameter.
- The ARCHMIG macro is used with the FORCML1=YES parameter to direct the data set to an ML1 volume.
- The SETSYS TAPEMIGRATION command with the RECONNECT(NONE) setting is in effect.
- The SETSYS CLOUDMIGRATION command with the RECONNECT(NONE) setting is in effect for migrations directed to cloud.

Requirements that must be satisfied when the data set is *recalled* are:

- The data set is recalled from a single ML2 tape volume or from cloud storage.
- The SETSYS USERDATASETSERIALIZATION setting is in effect.
- The SETSYS TAPEMIGRATION command with the RECONNECT(ALL) or RECONNECT(ML2DIRECTEDONLY), or SETSYS CLOUDMIGRATION command with the RECONNECT(ALL) setting is in effect.

Requirements that must be satisfied when a *fast subsequent migration reconnect is attempted* on the data set are:

- The data set was marked reconnectable when recalled.
- No changes have been made to the data set since it was recalled.
- The catalog entry for the data set has not been altered since it was recalled.
- The MCDS record for the data set has not been deleted (for example, when migration cleanup is performed).
- The ML2 tape volume has not been recycled since the data set was recalled.

- The SETSYS USERDATASETSERIALIZATION setting is in effect.
- The SETSYS TAPEMIGRATION command with the RECONNECT(ALL) or RECONNECT(ML2DIRECTEDONLY) setting is in effect.
- The cloud provider definition has not changed since the previous migration, or the cloud provider properties remain compatible.

Note:

1. Data sets that were migrated to an ML2 tape volume without requiring a backup copy (such as when using tape mount management) are eligible for fast subsequent migration as long as all other requirements are satisfied.
2. If a data set fails fast subsequent migration it is still eligible for standard migration.
3. A default date for fast subsequent migration is predicted when an eligible data set is recalled. The default date is the date the eligible data set was recalled plus the number of days the data set was unreferenced when it was last migrated. Fast subsequent migration is performed if the eligible data set is not referenced before that date and if the data set satisfies all fast subsequent migration and standard migration requirements.

Delaying migration cleanup of fast subsequent migration eligible data sets

Migration cleanup is performed on the MCDS records of fast subsequent migration eligible data sets a default number of days after the predicted remigration date if the data set was not reconnected (that is, fast subsequent migration was not performed). The default number of days is the default value of the SETSYS MIGRATIONCLEANUPDAYS command *reconnectdays* parameter.

Specify a *reconnectdays* value greater than the default to improve the likelihood of performing fast subsequent migration on eligible data sets by delaying migration cleanup of the MCDS records for those data sets.

Attention: The size of the MCDS might need to be increased to accommodate the extended life of MCDS records for fast subsequent migration eligible data sets.

Programming Interface Information

= Using ARCMDEXT to control fast subsequent migration eligibility

You can use the ARCMDEXT installation exit to redirect migrations that target ML1 volumes so that they target ML2 tape volumes instead. If you specified the SETSYS TAPEMIGRATION command with the RECONNECT(ALL) parameter, you can use this exit to deny reconnection for specific data sets. The ARCMDEXT exit is passed information in the parameter list, when fast subsequent migration is performed, as to whether a specific data set is reconnectable.

End Programming Interface Information

Related reading

- For more information about migration cleanup concept, see [“Migration cleanup” on page 59](#).
- For more information about configuring migration cleanup, see [“Configuring migration cleanup” on page 54](#).
- For more information about the MIGRATE command, see [Chapter 55, “MIGRATE command: Requesting a space management function,” on page 955](#).
- For more information about the SETSYS commands used to enable and disable fast subsequent migration and to delay migration cleanup, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#).
- For more information about the ARCHMIG macro, see [ARCHMIG: Migrating data sets in z/OS DFSMSHsm Managing Your Own Data](#).

- For more information about the ARCMDEXT installation exit, see ARCMDEXT: Space Management Exit in [*z/OS DFSMS Installation Exits*](#).

Chapter 3. Space management of non-SMS-managed storage

Eligibility for DFSMSHsm space management of non-SMS-managed storage is determined on a volume basis. Non-SMS-managed volumes that are to be automatically managed by DFSMSHsm are known as *primary* volumes. That is, all the data sets on a volume are managed to the same specification, whether for days-not-used on the primary volume, age for deletion, or permission for automatic migration. In addition, DFSMSHsm chooses the volume to receive the recalled data sets based on the volume pools you have defined, as well as the characteristics given the volumes when you defined them.

Differences between processing for SMS-managed and non-SMS-managed data sets are as follows:

- Two additional space management functions are available for non-SMS-managed volumes:
 - Deletion (delete by age)
 - Retirement (delete if backed up)
- Deletion of list and utility data sets.
- Level 1 to level 2 migration is controlled on a processing-unit-wide basis rather than on a data set management class basis.
- Target volumes for recall are selected by DFSMSHsm rather than by DFSMSdftp.
- Volumes are individually identified to DFSMSHsm using the ADDVOL command.

Related reading

- Before performing space management functions on extended address volumes (EAVs) you need to understand how EAV space is managed. For details, see [“Space management of extended address volumes \(EAVs\)”](#) on page 127 and review recommendations in [“Recommendations for using extended address volumes \(EAVs\)”](#) on page 129.
- For information about using DFSMSHsm with Linux®, see [Chapter 23, “Linux-z/OS DFSMSHsm dump or restore how-to,”](#) on page 665.

Differences in what the example system shows

The non-SMS-managed-storage space management example performs the same functions as were described for SMS-managed storage:

- Automatic space management
- Interval migration
- Automatic recall
- Command space management

During automatic primary space management, DFSMSHsm deletes certain list, utility, and expired data sets. DFSMSHsm does not perform partial release for unused allocated space on non-SMS-managed data sets but does perform extent reduction. In addition, the fast subsequent migration function allows unchanged data sets that are recalled from ML2 tape to reconnect to that original tape.

Tasks for space management of non-SMS-managed storage

To manage non-SMS-managed storage, you perform the same tasks as for managing SMS-managed storage except for the tasks involved with specifying storage management attributes to SMS. That is, you issue all the same SETSYS, ADDVOL, and DEFINE commands that have been specified so far. You do not define storage groups or management classes. In addition, you perform the following tasks:

- Specify the age for deleting list and utility data sets.
- Specify the minimum age for data sets to migrate if no age is specified when the volume is defined to DFSMSHsm.
- Specify when data sets become eligible for level 1 to level 2 migration.
- Specify recall characteristics for primary volumes.
- Define primary volumes to DFSMSHsm.
- Define pools of volumes.

Specifying the age for deleting list and utility data sets

During automatic space management, DFSMSHsm deletes list and utility data sets from primary volumes. DFSMSHsm does not migrate utility or list data sets during space management, nor does it back up utility and list data sets during availability management. However, it does process them if they are specified in the MIGRATE DATASETNAME(*dsn*) or BACKDS(*dsn*) commands.

All utility data sets must be at least two days old before DFSMSHsm scratches them. The utility data sets that are deleted by DFSMSHsm have the following types of data set names:

Data Set Name
HSM.MDB.Dnnnnn from HSM.MVS.Dccc
mcvtuid.MDB.Dccc
SYSnnnnn.Tnnnnnn.?Fccc
SYSnnnnn.Tnnnnnn.?Vccc
SYSnnnnn.Tnnnnnn.?Accc
N---N.FORMAT.Dnnnnn.Tnnnnnnnc
N---N.SUBMIT.Dnnnnn.Tnnnnnnnc
N---N.MERGE.Dnnnnn.Tnnnnnnnc
N---N.RUN.Dnnnnn.Tnnnnnnnc
N---N.SUB.Dnnnnn.Tnnnnnnnc
N---N.R.Dnnnnn.Tnnnnnnnn
N---N.M.Dnnnnn.Tnnnnnnnn
N---N.SPRINT.Dnnnnn.Tnnnnnnnc
N---N.DOSPR.Dnnnnn.Tnnnnnnnc
*SYSUT
<p>The variables shown in the data set names have the following values:</p> <ul style="list-style-type: none"> • <i>nnnnn</i> is a string of numeric characters whose length is represented by the number of <i>n</i>'s. This string typically consists of a time when preceded by a T or a date when preceded by a D. • <i>ccccccs</i> is an arbitrary length of arbitrary characters for the remaining portion of the data set name. • <i>N---N</i> represents the first qualifier in the data set name. • <i>mcvtuid</i> is the DFSMSHsm-authorized user identification specified with the UID parameter in the DFSMSHsm startup procedure. • <i>????s</i> is a string of arbitrary characters whose length is represented by the number of <i>?</i>'s.

All data set names must comply with the MVS data set naming conventions.

Before a list data set is scratched and uncataloged, it must meet the current DFSMSHsm value for the SCRATCHFREQUENCY parameter of the SETSYS command. List data sets that are deleted by DFSMSHsm have the following last qualifiers:

- LIST
- LINKLIST
- OUTLIST

In the example system, because only DFSMSHsm hosts 1 and 3 do automatic space management for the non-SMS-managed volumes, the following command is added to the ARCCMDxx member in DFSMSHsm hosts 1 and 3:

```
SETSYS SCRATCHFREQUENCY(5)
```

Specifying the minimum age for migrating data sets

The DAYS parameter of the SETSYS command specifies the minimum age for migrating data sets. If you do not specify this parameter, the minimum age is the same as the minimum you can specify for the serialization and processing-unit environment in your system. In the example system, the minimum age is specified as:

```
SETSYS DAYS(7)
```

The value specified by the DAYS parameter of the SETSYS command is used only if no value is specified for the MIGRATE parameter of the ADDVOL command or the MIGRATE parameter of the MIGRATE command. An ADDVOL command with the MIGRATE(0) parameter specified also causes SETSYS DAYS to determine the migration age.

Specifying when data sets become eligible for level 1 to level 2 migration

For non-SMS-managed storage, the age at which data sets become eligible for migration from level 1 to level 2 is the same for all non-SMS-managed data sets in the system. The MIGRATIONLEVEL1DAYS parameter of the SETSYS command specifies the minimum time that a data set must remain unused on the combination of level 0 and level 1 before it becomes eligible to migrate from level 1 to level 2. For example, for a MIGRATIONLEVEL1DAYS value of 31, a data set might stay unused on level 0 for 31 days, or it could stay on level 0 until it is eligible to migrate to level 1 and reside on level 1 until it was unused for a total of 31 days. In the example system, the following command is added to the ARCCMDxx member in DFSMSHsm host 2:

```
SETSYS MIGRATIONLEVEL1DAYS(60)
```

(The command is added only in DFSMSHsm host 2 because only DFSMSHsm host 2 does automatic level 1 to level 2 migration.)

If you want DFSMSHsm to do automatic migration from ML1 to ML2 volumes, you must specify the thresholds of occupancy parameter (of the ADDVOL command) for the ML1 volumes.

Specifying recall characteristics for the type of volume

For non-SMS-managed data sets, DFSMSHsm selects the type of volume to which a data set can be recalled. The characteristics for a volume to be eligible to receive recalled data sets are the same for all non-SMS-managed data sets in the system. The RECALL parameter of the SETSYS command specifies the general volume destinations for the recalled data sets. In the example system, the following command is added to the ARCCMDxx member in all DFSMSHsm hosts:

```
SETSYS RECALL(ANYSTORAGEVOLUME(LIKE))
```


This command causes any volume with the use attribute of storage and availability for automatic recall to be eligible for selection as a destination for any recalled data set not directed to a pool volume.

An alternative value for the subparameter of the RECALL command is PRIVATEVOLUME. The PRIVATEVOLUME subparameter specifies that data sets not directed to a pool can be recalled to any volume with the use attribute of public, private, or storage that has AUTORECALL specified in the ADDVOL command.

The LIKE subparameter that was specified causes DFSMSHsm to select as a target volume for recall only a volume with recall attributes like those of the volume from which the data set migrated. The alternative UNLIKE subparameter allows DFSMSHsm to select a volume with unlike recall attributes.

Related reading

- For more information about pool volumes and recall, see [“Defining pools of volumes” on page 105](#).
- For more information about recall attributes, see [“Recall attributes” on page 104](#).

JES3 considerations

The RECALL parameter should be consistent among the ARCCMDxx members of all the JES3 DFSMSHsm hosts. The subparameter to be used is selected in the DFSMSHsm host that performs the converter/interpreter locate function.

Defining primary volumes to DFSMSHsm

If multiple DFSMSHsm hosts are requested to process automatic primary space management, only the first DFSMSHsm host to process a given volume will perform the function. However, should you find it necessary to limit processing of certain volumes to one DFSMSHsm host, you can do so with the following parameters:

- Specifying the AUTOMIGRATION parameter on the ADDVOL command in the DFSMSHsm host that you want to perform automatic primary space management on the volume.
- Specifying the NOAUTOMIGRATION parameter on the ADDVOL command in the DFSMSHsm hosts that you do not want to perform automatic primary space management on the volume.

At the same time that you make the volume known to DFSMSHsm, you specify the following characteristics:

- Whether to perform automatic space management on the volume
- Whether the volume is available for automatic recall of all data sets
- The space management technique for the volume

The ADDVOL command defines non-SMS-managed volumes to DFSMSHsm. You cannot use the ADDVOL command for SMS-managed volumes. Non-SMS volumes that are added with the AM parameter on the ADDVOL command work in a similar way to a volume added with an AM=Y storage group. See [Table 2 on page 31](#) for a discussion of standard interval migration processing.

Note: A maximum of eight z/OS images can ADDVOL a primary volume at the same time.

For the example system, the following ADDVOL commands are added to the ARCCMDxx member in DFSMSHsm hosts 1 and 2:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTOMIGRATION -  
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL GP0002 UNIT(3390) PRIMARY(AUTOMIGRATION -  
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL GP0003 UNIT(3390) PRIMARY(AUTOMIGRATION -  
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL UG1001 UNIT(3390) PRIMARY(AUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)
```



```

ADDVOL UG1002 UNIT(3390) PRIMARY(AUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1003 UNIT(3390) PRIMARY(AUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2001 UNIT(3390) PRIMARY(AUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2002 UNIT(3390) PRIMARY(AUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2003 UNIT(3390) PRIMARY(AUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0001 UNIT(3390) PRIMARY(AUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0002 UNIT(3390) PRIMARY(AUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0003 UNIT(3390) PRIMARY(AUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

```

The first parameter of the command specifies the volume serial number. Notice that each of the volumes to be managed individually must be added. Primary volumes must be online and permanently resident or reserved.

The second parameter specifies the unit type.

The third parameter specifies that the volume is to be added as a primary volume. Notice that this parameter has subparameters that define whether and how automatic space management is to be performed.

The AUTOMIGRATION subparameter specifies that DFSMSShsm is to perform automatic volume space management on the volume. The alternative NOAUTOMIGRATION prevents DFSMSShsm from performing automatic volume space management.

The AUTORECALL|NOAUTORECALL subparameter specifies whether the volume is generally available as a recall volume. Notice that AUTORECALL is specified only for volumes GP0001, GP0002, and GP0003. The UG and VP volumes are indicated as NOAUTORECALL because they are members of pools. See the discussion of recall to pools in [“Defining pools of volumes”](#) on page 105.

The DELETEBYAGE|DELETEIFBACKEDUP|MIGRATE subparameters specify the management technique for the volume. For all volumes in the example system, DFSMSShsm migrates the data sets that have not been opened in the number of days specified with the MIGRATE subparameter. You can specify any value equal to or larger than the integrity age up to 999. If you specify the DELETEBYAGE subparameter, DFSMSShsm deletes any data set without an expiration date or with an expiration date that has passed, and that has not been opened in the number of days specified with the DELETEBYAGE subparameter. You can specify any value equal to or larger than the integrity age (see [“Integrity age for space management”](#) on page 56) up to 999. If you specify the DELETEIFBACKEDUP subparameter, DFSMSShsm deletes any data set that has not been opened in the number of days specified with the DELETEIFBACKEDUP subparameter if a current backup version of the data set exists. You can specify any value equal to or larger than the integrity age up to 999.

If you do not specify a value for the number of days with the MIGRATE subparameter, DFSMSShsm uses the value specified with the DAYS parameter of the SETSYS command.

The THRESHOLD parameter in these ADDVOL commands operates in the same way as the THRESHOLD attribute in the storage groups. However, for volumes managed by DELETEBYAGE and DELETEIFBACKEDUP techniques, the thresholds are ignored and all eligible data sets on the volume are processed.

Attributes of primary volumes

When you use the ADDVOL command to add a primary volume to DFSMSShsm, you assign attributes to that primary volume. You can choose the following primary volume attributes:

- AUTOMIGRATION or NOAUTOMIGRATION
- AUTORECALL or NOAUTORECALL
- AUTOBACKUP or NOAUTOBACKUP
- BACKUPDEVICECATEGORY
- AUTODUMP or NOAUTODUMP

The use of the first two primary volume attributes is described in this topic. The use of the last three attributes is described in [Chapter 7, “Availability management of non-SMS-managed storage,”](#) on page 303.

Each DFSMSShsm host can decide whether primary volumes are processed by automatic space management, automatic backup, or automatic dump. However, if an automatic process controls a primary volume in any DFSMSShsm host, DFSMSShsm considers the volume to be automatically controlled for that process. For example, assume you specify the following in DFSMSShsm hosts 1 and 2:

```
ADDVOL VOL006 PRIMARY(AUTOMIGRATION) UNIT(3390)
```

Then you specify the following in DFSMSShsm host 3:

```
ADDVOL VOL006 PRIMARY(NOAUTOMIGRATION) UNIT(3390)
```

DFSMSShsm considers VOL006 to be controlled by automatic space management.

Recall attributes

The primary volume attributes of AUTOMIGRATION, AUTOBACKUP, and BACKUPDEVICECATEGORY are also designated as recall attributes. That is, DFSMSShsm considers the values of these attributes in selecting target volumes for recalling non-SMS-managed data sets. The relationship of these attributes to those of the available volumes is what you specify when you use the LIKE and UNLIKE values with the ANYSTORAGEVOLUME and PRIVATEVOLUME subparameters of the RECALL parameter of the SETSYS command.

Because DFSMSShsm hosts 1 and 2 are to do all automatic space management for the non-SMS-managed volumes, you can add the following commands to the ARCCMDxx member in DFSMSShsm host 3:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    AUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)  
  
ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTOMIGRATION -  
    NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)
```



```
ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTOMIGRATION -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)
```

Defining pools of volumes

To ensure that recalled non-SMS-managed data sets go to the correct volumes, you can organize your non-SMS-managed volumes into pools. You can define either pools that accept only specified data set names during recall or pools that accept for recall any data set that migrated from them. In this topic, you are first shown how to define the pools, and then the effects of the pools are described.

You use the DEFINE command to define pools of volumes. The DEFINE command associates a pool ID with a list of volume serial numbers. In the example system, the commands added to the ARCCMDxx member in all DFSMSHsm hosts are:

```
DEFINE POOL(GROUP1 VOLUMES(UG1001 UG1002 UG1003))
DEFINE POOL(GROUP2 VOLUMES(UG2001 UG2002 UG2003))
DEFINE VOLUMEPOOL(SAMPLE VOLUMES(VP0001 VP0002 VP0003))
```

The first two examples specify data set pools. The data set pool list specifies DFSMSHsm primary volumes where DFSMSHsm is to recall data sets whose set of initial characters match that pool ID. For the first example, data sets with an initial character sequence of GROUP1 are recalled to it. For the second example, data sets with an initial character sequence of GROUP2 are recalled to it.

The third example specifies a volume pool. A volume pool associates volumes from which data sets migrate with other volumes, to aid in the selection of target volumes for recall.

You can specify nonresident volumes in a volume pool, to allow recall of data sets that migrated from volumes that no longer exist in the system. Not all volumes in a volume pool are required to be managed by DFSMSHsm. You can specify SMS-managed volumes in volume pools, although they will never be selected from the pool for non-SMS-managed data.

Data sets that migrate from a volume in a volume pool are recalled to an online volume that has been added to the list of volumes managed by DFSMSHsm using the ADDVOL command. The target volume selection is based on the volume pool configuration at the time of recall. To change the definition of a volume pool (adding or deleting volumes from the pool), the DEFINE command must include all the volumes you want as part of the pool, including any volumes you selected with a previous DEFINE command if you still want them as part of the pool. The volume pool is limited to 140 volumes.

All volumes in a pool need to have the same geometry. For example, you could have a volume pool that included a 3390 Model 3, and a 3390 Model 9 because they share the same geometry. This means that a data set originally allocated on one model could be recalled to a different model.

A primary volume can be in any number of data set or volume pools and can also be available for general recall use if it is online, permanently resident or reserved, and has the AUTORECALL subparameter in its ADDVOL command.

In a loosely coupled multiple DFSMSHsm-host environment:

- A data set or volume pool should be the same in all DFSMSHsm hosts.
- Volumes that are available for general recall must be accessible to all DFSMSHsm hosts.
- A volume that is removed from the environment should be removed from DFSMSHsm by the DELVOL command on *all* DFSMSHsm hosts, and from any ADDVOL commands that may be in the startup member. It may remain in the pool definition. In this way, DFSMSHsm can translate a recall request to a currently active volume in the pool.

Pools in systems running under JES2

In systems running under JES2, you can define two kinds of pools: data set pools and volume pools. Data set pools and volume pools are known as *user-defined* pools. You can define these pools at any time, either during the startup procedure or after DFSMSHsm has been started. Similarly, you can use the ADDVOL command to add volumes to DFSMSHsm control at any time.

Any volume that has the AUTORECALL subparameter in its ADDVOL command is eligible to be used for recall of any data set that is not restricted to a pool of volumes. This provides a kind of default pool.

Pools in systems running under JES3

In systems running under JES3, three kinds of pools exist. In addition to the user-defined pools that you define as described earlier, DFSMSHsm configures a general pool from the unit control block associated with primary volumes and from information obtained from the ADDVOL command. The general pool is a collection of all primary volumes that:

- Are online.
- Are permanently resident or reserved.
- Have a primary volume attribute of AUTORECALL.
- Are mounted with the use attribute of storage or private. (The private volumes are included only if you specify the SETSYS RECALL(PRIVATE) command.)

Volumes in the general pool can also be defined in one or more data set or volume pools. The general pool is the default pool where DFSMSHsm recalls migrated data sets if they are not restricted for recall to a user-defined pool.

Specifying pools in systems running under JES3

Because DFSMSHsm returns a list of volumes to JES3 at converter/interpreter time, the system pools that are running under JES3 have restrictions on when they can be established. You can specify pool configurations and recall options only at designated times. You must specify the pool configuration command in the ARCCMDxx member that is used when you start DFSMSHsm with HOSTMODE=MAIN. (A pool configuration that is specified when you are starting HOSTMODE=AUX is ignored.) You cannot change pool configurations until the MAIN DFSMSHsm restarts. Pool configurations and recall options must be consistent in all z/OS images that run DFSMSHsm with the appropriate HOSTMODE and perform the JES3 locate function.

The ADDVOL, DEFINE, DELVOL, and SETSYS commands determine the configuration of pools of volumes. When DFSMSHsm is used with JES3, the DFSMSHsm general pool and user-defined pools can be configured only as a result of the commands in the ARCCMDxx member processed when DFSMSHsm is being initiated. Certain variations of these DFSMSHsm commands are accepted only during that initialization period. The one exception is the DELVOL command. The DELVOL command is accepted at any time after initialization by any DFSMSHsm that is running in the JES3 image. Therefore, JES3 does not expect a data set on a volume that is unusable. For example, you could use the DELVOL command to delete a primary volume that has I/O errors, so that DFSMSHsm does not tell JES3 to use that volume.

To ensure that pools function correctly in a system running under JES3, consider the following conditions:

1. Before you can define pools in each z/OS image, you must specify the following commands for each DFSMSHsm host that you are starting in that image:
 - A SETSYS JES3 command. The JES3 parameter of the SETSYS command determines the need for the general pool. For a JES3 system, DFSMSHsm builds the general pool when you start HOSTMODE=MAIN. Because information from the ADDVOL command is used to build the general pool, DFSMSHsm must know that JES3 is the job entry system before the first ADDVOL command is issued for a primary volume. Once the job entry subsystem is determined, you cannot change it until DFSMSHsm is restarted.
 - A SETSYS RECALL command with its appropriate subparameter. The ANYSTORAGEVOLUME and PRIVATEVOLUME subparameters of the RECALL parameter of the SETSYS command are not accepted after the initialization period of DFSMSHsm has passed, because the method of recalling data sets determines the volumes that qualify for the general pool.
 - An ADDVOL command for primary volumes that DFSMSHsm is to manage. You must specify the PRIMARY parameter of the ADDVOL command for a volume before that volume is included in any pool. Subsequent ADDVOL commands for the same DFSMSHsm primary volume must not change the information used by DFSMSHsm to define pools. The volume must still:

- Be online (see the note in item “2” on page 107)
- Be permanently resident or reserved
- Have the same automatic recall parameters specified

The ADDVOL command can be issued at any time for volumes other than DFSMSHsm primary volumes just as in a DFSMSHsm environment without JES3.

Note: Be sure that you include the ADDVOL command specifications in the ARCCMDxx member of SYS1.PARMLIB so that DFSMSHsm recognizes the volumes at each startup. If ADDVOL commands for ML1 volumes are not in the ARCCMDxx member, DFSMSHsm does not recognize that they are available unless you issue an ADDVOL command at the terminal for each ML1 volume.

2. The POOL parameter of the DEFINE command can be issued only during initialization of DFSMSHsm. DFSMSHsm does not accept more than one DEFINE command for the same pool ID during initialization. All DEFINE commands issued for volumes to be included in a data set pool must be preceded by an ADDVOL command for each volume.

Note: For data set pooling, volumes that are offline at DFSMSHsm startup time are added to DFSMSHsm as offline and are not added to the data set pool. When these volumes become available, DFSMSHsm must be restarted to add the volumes as online and include them in the data set pool.

3. The VOLUMEPOOL parameter of the DEFINE command can be issued only during initialization of DFSMSHsm. DFSMSHsm will not accept more than one DEFINE command for the same pool ID during initialization. At least one volume in the volume pool must have been added with a previous ADDVOL command to the list of volumes DFSMSHsm manages or owns.
4. You can specify other parameters of the DEFINE command as if in a DFSMSHsm environment without JES3.
5. Be sure that you define the same data set and volume pools in all z/OS images that can process the volumes.
6. To ensure that recall is to a volume that you choose, do not refer to any data set eligible for space management before the appropriate pool configuration in that z/OS image has been defined to DFSMSHsm.
7. Specify each DFSMSHsm primary volume with the same parameters (for a given volume) in each z/OS image. The mount status of permanently resident or reserved prevents DFSMSHsm primary volumes from being demounted. One of these attributes must be in effect for each DFSMSHsm primary volume.
8. When a local z/OS image assumes the function of the global image, any required DFSMSHsm functions that are performed by the previous global image must be active on the new global image. Specify the same pool configuration on each image to protect this type of JES3 activity.
9. Each JES3 z/OS image can have unique user-defined pools. However, if your system uses unique user-defined pools whose volumes are not accessible to the global z/OS image, you must use job classes or JES3 JCL to direct all jobs that access the unique user-defined pools to the z/OS image that can access them. This is a requirement when both the global and the local z/OS images request JES3 locates. If any z/OS image has access to a primary volume in a data set pool, it must have access to all primary volumes in that data set pool, or the job could fail. If you decide not to give all JES3 z/OS images access to all primary volumes, use caution when you define and operate with unique data set pool configurations.
10. If any device containing a volume in a data set pool is designated as a device managed by JES3, all other volumes in that data set pool must reside on JES3-managed devices. JES3 schedules jobs only on z/OS images that have access to all volumes that the jobs require.
11. Any z/OS image that runs the job must have access to all the volumes passed to JES3 by DFSMSHsm.

Example of JES3 pooling

Figure 13 on page 108 shows three z/OS images in a JES3 and DFSMSHsm environment. The following describes some pertinent facts about the figure:

- Volumes A, B, C, D, F, G, X, and Y make up the primary volume list. All the z/OS images have access to the online storage volumes X and Y. However, because only X and Y have the AUTORECALL primary volume attribute, non-pooled data sets are recalled only to volumes X and Y.
- Local z/OS images 1 and 3 have access to volumes A and B.
- Volumes A and B are both in data set pools USER1 and USER3. Jobs that access these volumes can be run only by z/OS image 1 and z/OS image 3, even though z/OS image 2 can set up the jobs.
- z/OS image 1 has access to volume Q. DFSMSHsm does not manage volume Q.
- z/OS image 3 has access to volumes F and G, which make up data set pool USER2.
- Volumes C and D are in a volume pool. If a data set migrated from either volume C or D with a high-level qualifier other than USER1, USER2, or USER3, the migrated data set is recalled to either volume C or D.

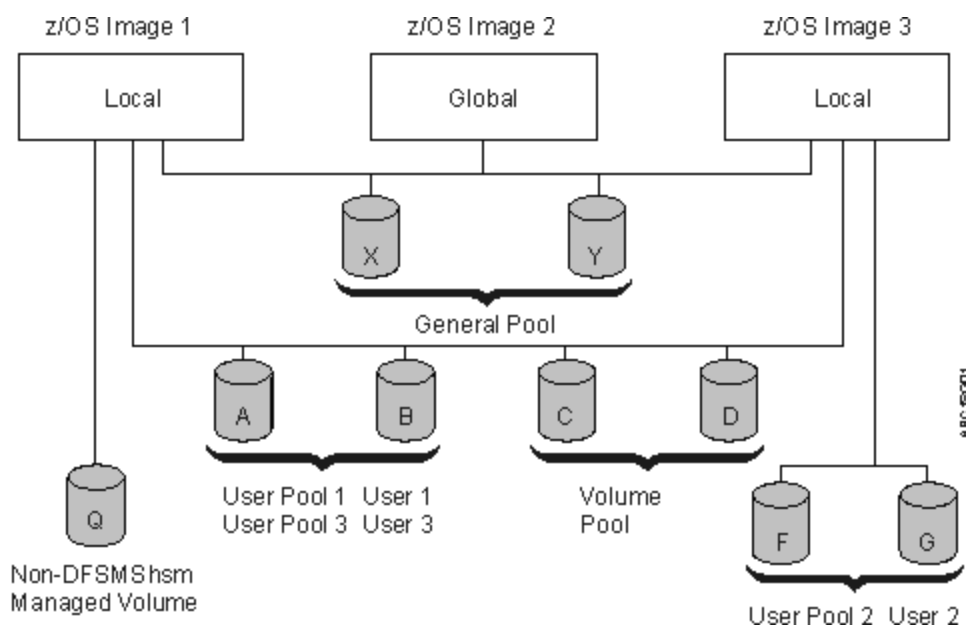


Figure 13. Multiple-Processing Units in a JES3 and DFSMSHsm Environment

You must issue these commands on all DFSMSHsm hosts on all three z/OS images:

```
SETSYS JES3 RECALL (ANYSTORAGEVOLUME)
ADDVOL X UNIT(3390) PRIMARY(AUTORECALL)
ADDVOL Y UNIT(3390) PRIMARY(AUTORECALL)
ADDVOL A UNIT(3390) PRIMARY(NOAUTORECALL)
ADDVOL B UNIT(3390) PRIMARY(NOAUTORECALL)
ADDVOL C UNIT(3390) PRIMARY(NOAUTORECALL)
ADDVOL D UNIT(3390) PRIMARY(NOAUTORECALL)
ADDVOL F UNIT(3350) PRIMARY(NOAUTORECALL)
ADDVOL G UNIT(3350) PRIMARY(NOAUTORECALL)
DEFINE POOL(USER1 VOLUMES(A B))
DEFINE POOL(USER3 VOLUMES(A B))
DEFINE POOL(USER2 VOLUMES(F G))
DEFINE VOLUMEPOOL(SHARED VOLUMES(C D))
```

Any of the three z/OS images in [Figure 13](#) on [page 108](#) can successfully run a JES3 locate routine if the appropriate catalogs are accessible.

Automatic space management processing for non-SMS-managed storage

Processing of non-SMS-managed storage differs from that of SMS-managed storage.

In the example system, DFSMShsm performs space management on the system configuration shown in Figure 14 on page 110. Remember, however, that the system for which the example system is built is that shown in Figure 4 on page 21 and that Figure 14 on page 110 is merely a subset that is used for simplification. When space management starts (first on DFSMShsm host 1 and later on DFSMShsm hosts 2 and 3), the space management is being done for all of the systems shown in Figure 4 on page 21. Table 10 on page 109 shows the characteristics for each data set in the various groups of volumes shown in Figure 14 on page 110. The names of the data sets are not precisely as the system would generate them. They have been shortened and modified for convenience in using them as examples.

Table 10. Data Set Characteristics for Automatic Space Management Example (Non-SMS-Managed Volumes)

Data Set Name	Inactive Age (Days)	Size (KB)	GDG Status
GROUP1.SAMPLE.A1	16	217	None
GROUP1.SAM.LIST	6	21	None
GROUP2.SAMPLE.B1	5	210	None
U1234.G3.G4	20	20	None
U1234.G3.G5	16	605	None
U2345.G3.G6	61	350	None
LARGE.ML.C1	16	1100	None

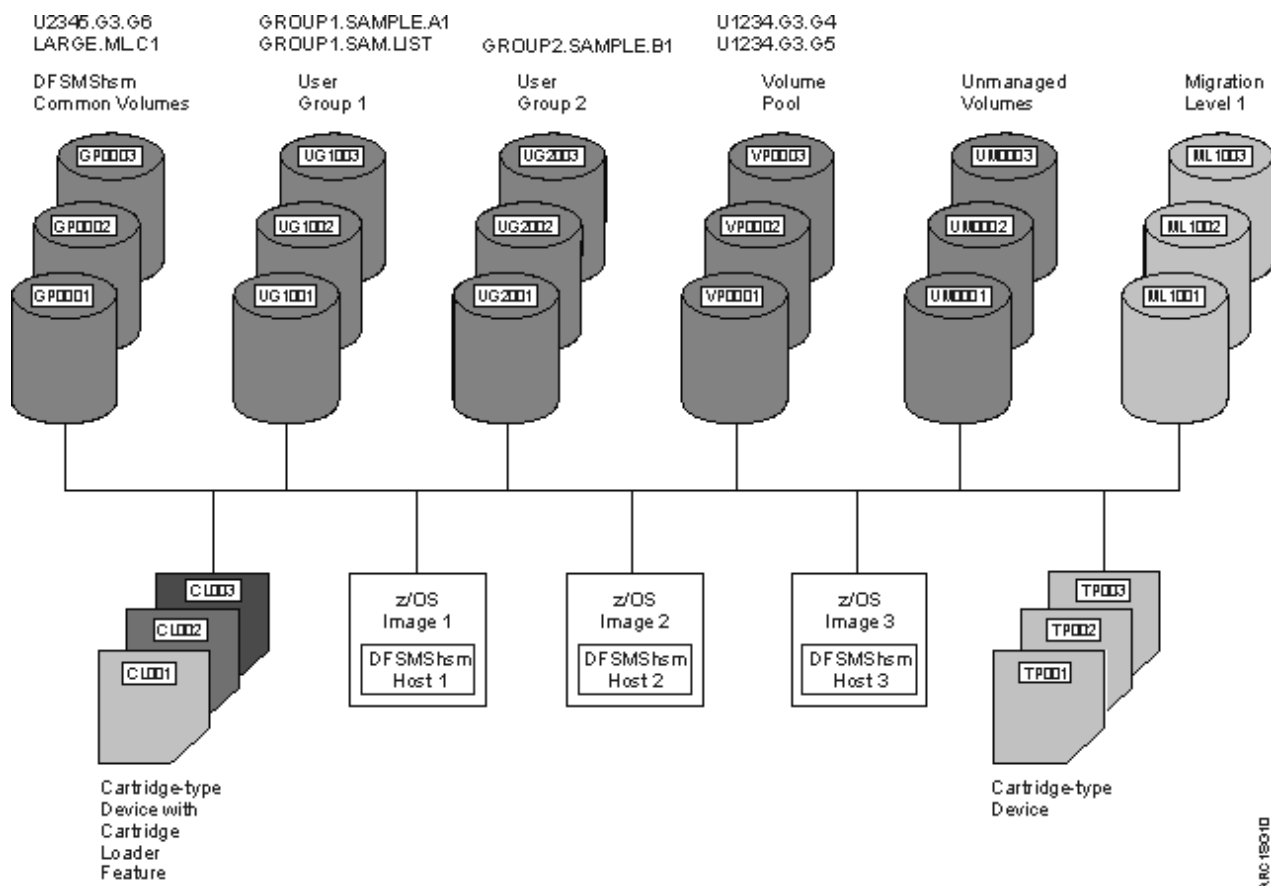


Figure 14. Example System of Non-SMS-Managed Volumes before Automatic Space Management

Automatic secondary space management

Although migration cleanup and level 1 to level 2 migration are performed on non-SMS-managed data sets, they operate the same whether the system is processing SMS-managed or non-SMS-managed storage. In the example system, they are performed by DFSMSHsm host 2.

During migration cleanup, DFSMSHsm uses the EXPIREDDATASETS(SCRATCH) parameter of the SETSYS command to determine whether to delete expired non-SMS-managed data sets. It uses the *statdays* and *recalldays* subparameters of the MIGRATIONCLEANUPDAYS parameter of the SETSYS command to determine when to delete statistics records and MCD records for recalled data sets respectively. The use of these parameters is discussed in [“Migration cleanup”](#) on page 59.

During level 1 to level 2 migration of non-SMS-managed data sets, DFSMSHsm uses the MIGRATIONLEVEL1DAYS parameter of the SETSYS command to determine when to migrate data sets from migration level 1 to migration level 2.

For performance reasons, level 1 to level 2 migration is not performed during automatic secondary space management in a non-SMS and direct-to-tape migration environment. DFSMSHsm assumes there are no migrated data sets on any ML1 volumes that need to be moved to ML2 volumes in this environment. If, for any reason, there are migrated data sets residing on any ML1 volumes that are to be moved to ML2 volumes in a non-SMS and direct-to-tape migration environment, the MIGRATE ML1 command should be used.

Related reading

For more information about the use of the MIGRATIONLEVEL1DAYS parameter, see [“Specifying when data sets become eligible for level 1 to level 2 migration”](#) on page 101.

Automatic primary space management

During automatic primary space management, DFSMSHsm does space management on each primary volume with the AUTOMIGRATION primary volume attribute. The space management technique that you specify with the ADDVOL command determines the kind of space management that DFSMSHsm does: data set deletion, data set retirement, or migration.

Automatic primary space management for non-SMS-managed volumes is performed in two passes.

Pass 1

During pass 1, DFSMSHsm examines every data set on the volume (independent of the occupancy of the volume), and checks the integrity age to ensure that it can process the data set. The inactive age must be more than or equal to the integrity age. The inactive age is the difference between the date of processing (current date) and the value in the date-last-referenced field of the data set VTOC entry. (For a discussion of integrity age, see [“Integrity age for space management” on page 56.](#)) However, for non-SMS-managed data sets, DFSMSHsm does not perform the following actions:

- Issue a message for unmatched data sets
- Delete temporary data sets
- Release unused space in non-VSAM data sets
- Expire data sets by inactive age or age since creation

For non-SMS-managed data sets, DFSMSHsm does perform the following actions:

- Delete list and utility data sets
- Delete data sets that have passed their expiration dates
- Determine eligible data sets for migration, including fast subsequent migration
- Determine eligible data sets to be backed up
- Determine eligible data sets for extent reduction

For each data set on the volume, DFSMSHsm uses the parameters specified in the SETSYS and ADDVOL commands in that DFSMSHsm host.

Programming Interface Information

<p>As DFSMSHsm considers each data set for eligibility for processing, it passes control to the space management and backup installation exit (ARCSAEXT) if the installation exit is enabled. (Refer to z/OS DFSMS Installation Exits manual for a discussion of ARCSAEXT.) If the installation exit indicates that processing is to continue (or the installation exit is not enabled), DFSMSHsm processes the data set. If the installation exit indicates that processing is not to continue, DFSMSHsm begins processing the next data set.</p>
--

End Programming Interface Information

Deleting list and utility data sets

As DFSMSHsm processes each data set, it deletes those utility data sets that are more than two days old and whose names follow one of the formats listed under [“Specifying the age for deleting list and utility data sets” on page 100](#). It also deletes those list data sets having names with the final qualifier as specified under [“Specifying the age for deleting list and utility data sets” on page 100](#), and older than the age specified with the SCRATCHFREQUENCY parameter of the SETSYS command.

Deleting expired data sets

For non-SMS-managed data sets, DFSMSHsm deletes those data sets that have passed their expiration date. For non-VSAM data sets this would be the expiration date in the data sets' VTOC entries and for VSAM data sets this would be the expiration date in the data sets' catalog entries. For these data sets, the EXPIREDDATASETS parameter of the SETSYS command controls whether the data sets are deleted.

Note: For multivolume non-VSAM data sets, DFSMSHsm looks at the VTOC of the first volume.

Determining eligible data sets for migration

For non-SMS-managed data sets, the criteria for whether a data set is eligible for migration include the inactive age of the data set compared to the *days* value specified with the migration technique parameter of the ADDVOL command for the volume on which the data set resides. If the inactive age is larger than or equal to the *days* value, the data set becomes eligible for migration. If no *days* value has been specified on the ADDVOL command or the *days* value is equal to zero, DFSMSHsm uses the value specified with the DAYS parameter of the SETSYS command. The *days* value, whether taken from the ADDVOL command or the DAYS parameter of the SETSYS command, is called the minimum migration age.

Related reading

- For more information about how to determine the inactive age, see [“Pass 1” on page 111](#).
- For more information about the reconnection function, see [“Reconnecting data sets using fast subsequent migration” on page 94](#).

Determining eligible data sets for backup

DFSMSHsm must determine which data sets are eligible for backup so that it can schedule the data sets for backup if they migrate or prevent migration if the migration target volume is a tape volume. A non-SMS-managed data set is eligible for backup if all of the following apply:

- The data-set-changed indicator in the data set VTOC entry indicates that the data set has been changed.
- The volume on which the data set resides has the AUTOBACKUP attribute in *any* DFSMSHsm host in the system.
- The data set is supported for automatic backup, as indicated in the availability management data mover support table in [z/OS DFSMSHsm Implementation and Customization Guide](#).

Determining eligible data sets for extent reduction

The criteria for whether a data set is eligible for extent reduction are the same as for SMS-managed data sets.

Pass 1 processing example

Figure 15 on page 113 shows the results of pass 1 processing for non-SMS-managed volumes in the example. Data set GROUP1.SAM.LIST is deleted because it is a list data set that is older than the SCRATCHFREQUENCY value set with the SETSYS command.

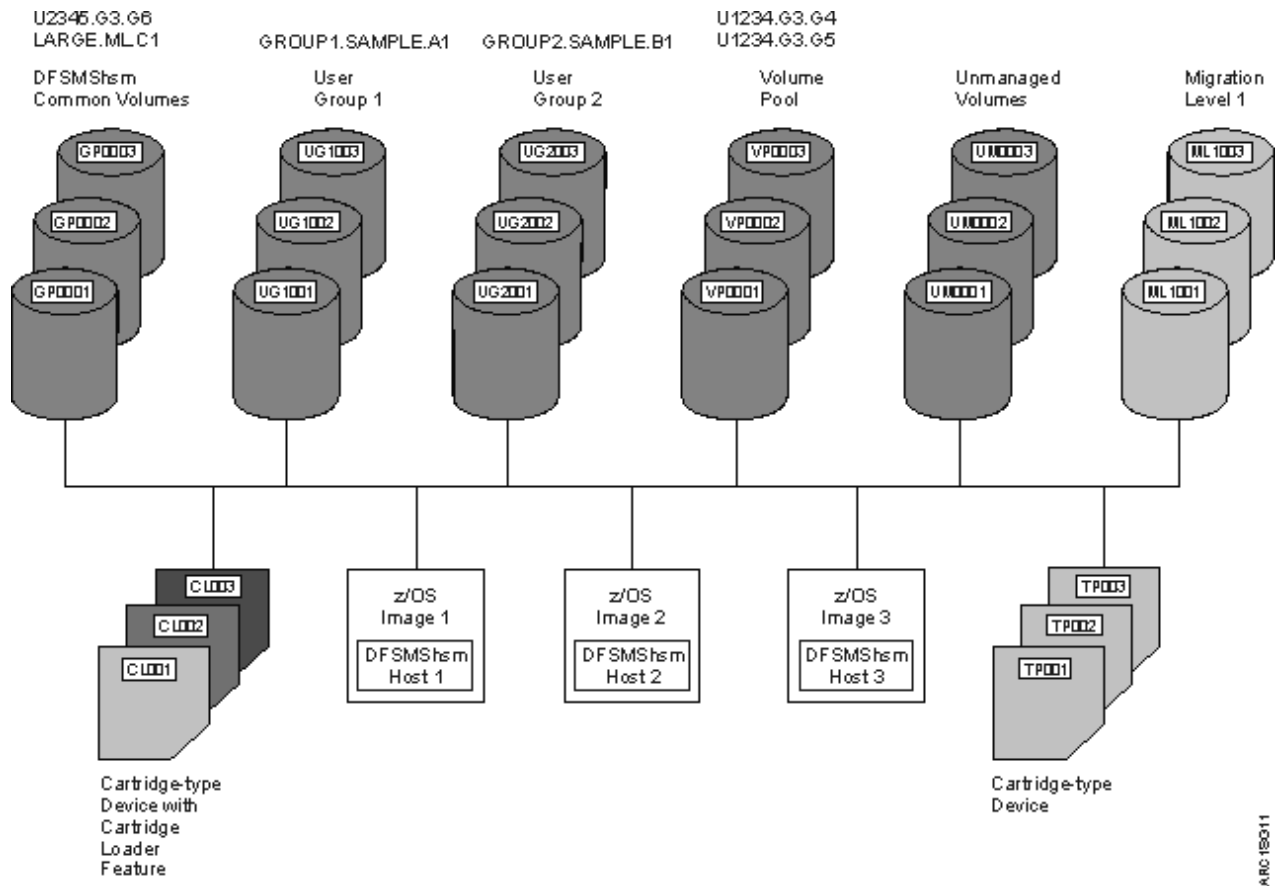


Figure 15. Example System of Non-SMS-Managed Volumes after Primary Volume Management Pass 1

Pass 2

Although the functions performed in pass 2 of primary volume space management for non-SMS-managed volumes are similar to those for phase 2 processing of SMS-managed volumes, the manner of processing is considerably different. Pass 2 processing does the following tasks:

- Sorts the candidates for migration into priority order
- Performs space management on the volume

Sorting the candidate data sets into priority order

For volumes that have the space management technique of migration, DFSMSHsm sorts the names of the data sets into migration priority order, high priority to low priority. No consideration is given to generation data sets; all data sets on the volume receive their priority on the basis of age and size. Note, however, that the age used for the computation is the inactive age as opposed to the eligibility age used for SMS-managed data sets.

Performing space management on the volume

During primary space management for non-SMS-managed volumes, DFSMSHsm processes the volume according to the one technique specified by the ADDVOL command for the volume. Thus, all the eligible data sets on the volume are managed to the same inactive age and by the same management technique. The available techniques are:

- Delete by age (deletion)
- Delete if backed up (retirement)
- Migration (with or without thresholds of occupancy)

Data set deletion

For those primary volumes with the DELETEBYAGE space management technique, DFSMSHsm scratches low-activity cataloged and uncataloged data sets based on the inactive age of the data sets. You specify the inactive age when you specify the space management technique. DFSMSHsm scratches only data sets whose expiration dates have been reached or data sets without expiration dates. For data set deletion, DFSMSHsm ignores any thresholds you specify and scratches all eligible data sets.

As DFSMSHsm examines each data set for deletion, it passes control to the data set deletion installation exit (ARCADEXT).

Related reading

For more information about the ARCADEXT installation exit, see [*z/OS DFSMS Installation Exits*](#).

Data set retirement

For those primary volumes with the DELETEIFBACKEDUP space management technique, DFSMSHsm checks the backup control data set (BCDS) to be sure that a current backup version of the data set exists before DFSMSHsm scratches the data set based on its inactive age. You specify the inactive age when you specify the space management technique. DFSMSHsm scratches a data set even if the expiration date has not been reached.

DFSMSHsm creates backup versions when automatic or command backup runs. If a backup version is made on the date the data set is last opened, DFSMSHsm does not consider the backup version to be current. This means that DFSMSHsm must make a backup version of a cataloged data set on the calendar day *after* the last activity of the data set. For non-VSAM data sets, the backup must be made after the last-referenced day. For VSAM data sets, the backup must be made after the last-update day. Therefore, if you want to use data set retirement often, schedule backup to run soon after midnight.

By running backup soon after midnight, you reduce your exposure to having the last backup made on the same date that the data set was last changed. If the data set is changed before midnight, automatic backup only occurs on the next calendar date, so the dates would not be the same. There are two possible scenarios where exposures might occur:

- Data sets that were changed and then backed up by command on the same day
- Data sets that were changed between midnight and the time that automatic backup encountered them in the course of its processing

Table 11 on page 114 shows what DFSMSHsm considers to be the current backup version of migrated and unmigrated data sets.

Table 11. Current Backup Version of Data Sets	
Type of Data Set	Current Backup Version
Not migrated VSAM	DFSMSHsm made the backup version since the last day someone updated the data set.
Not migrated, cataloged, non-VSAM	The update flag is off and the date-last-referenced that existed when the data set was backed up is earlier than the date the data set was last backed up.
Uncataloged, non-VSAM	The update flag is off and DFSMSHsm made a backup version of the data set from the volume being processed.
Migrated VSAM	The update flag is off and DFSMSHsm created the backup version after the date DFSMSHsm last migrated the data set.
Migrated non-VSAM	The update flag is off and the date-last-referenced that existed when the data set was migrated is earlier than the date the data set was last backed up.

During data set retirement, DFSMSHsm marks the latest backup version as a retirement version and gives it special protection against inadvertent deletion.

As DFSMSHsm examines each data set for retirement, it passes control to the data set deletion installation exit (ARCADEXT).

Migration

For those primary volumes with the MIGRATE space management technique defined on the ADDVOL command, DFSMSHsm uses the minimum migration age or the low threshold of occupancy specified for the volume to determine how many data sets are to be migrated. You specify the minimum migration age when you specify the space management technique. During primary volume space management, DFSMSHsm uses the priority order determined in the earlier step if thresholds have been specified for the volume. If no thresholds have been specified, DFSMSHsm just migrates each candidate data set that it finds on the volume. The end of DFSMSHsm's migration of data sets depends on whether you have specified thresholds of occupancy for the primary volume.

As DFSMSHsm examines each data set for migration, it passes control to the data set migration installation exit (ARCMDEXT).

With thresholds of occupancy

Data sets migrate in priority order from the primary volume until the low threshold of occupancy is reached or until the age of the oldest remaining data set is less than the minimum migration age.

Example: If a primary volume has a low threshold of occupancy of 80% and a minimum migration age of four days, DFSMSHsm migrates data sets until at least 20% of the volume is available or until all data sets not referred to within the last three days have migrated.

Without thresholds of occupancy

DFSMSHsm uses the minimum migration age of the data sets in selecting data sets to migrate from the volume.

Example: If the minimum migration age is seven days, data sets migrate from the primary volume until DFSMSHsm finds that all remaining data sets have been referred to within the last six days. The minimum migration age is limited by the integrity age.

Related reading

- For more information about the affect of using ARCMDEXT for data set retirement or migration, see [z/OS DFSMS Installation Exits](#).
- For more information about the data set integrity age, see [“Integrity age for space management”](#) on [page 56](#).

Space management of VSAM data sets

DFSMSHsm migration functions that exist for non-VSAM data sets also exist for VSAM data sets cataloged in the integrated catalog facility catalog.

The migration of a non-SMS-managed VSAM data set is performed only on a complete VSAM sphere. The base cluster and all alternate index clusters defined on the base cluster are migrated as part of the data set and are not specified separately. Paths are included automatically in the data set command migration.

When DFSMSHsm migrates a VSAM data set, it tracks a maximum of eight component names for that data set. [Figure 16 on page 116](#) shows component name limitations when DFSMSHsm is recalling a migrated VSAM data set.

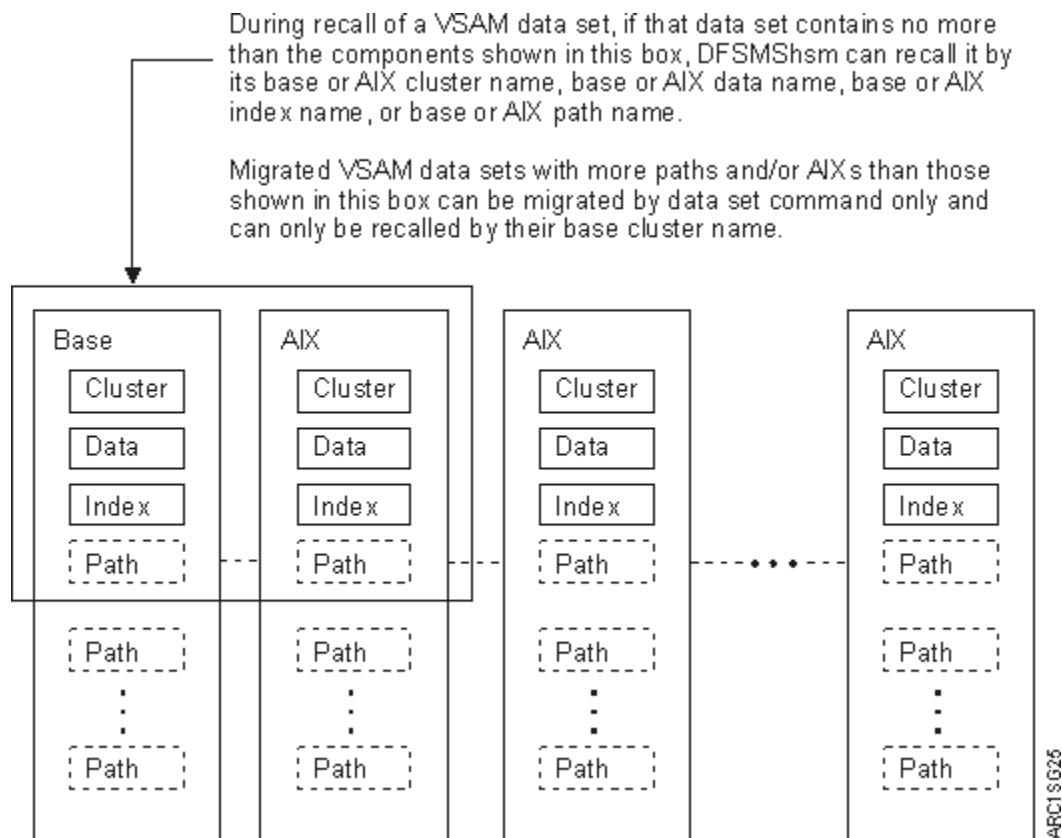


Figure 16. Component Name Limitations When Recalling Migrated VSAM Data Sets

A non-SMS-managed VSAM data set can be migrated only if it is completely contained on one volume. DFSMSHsm does not migrate a sphere that has a component that needs to be verified.

Data sets cataloged in the integrated catalog facility catalog with more than one alternate index (AIX), more than one path on the base, or more than one path defined on the AIX are not selected for migration during volume processing. These data sets can be migrated by data set command migration.

DFSMSHsm does not migrate a VSAM data set marked for forward recovery, or one with retained locks.

Related reading

For more information about the data sets not supported by space management, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

JES3 considerations

For the recall of SMS-managed data sets, JES2 and JES3 appear to be the same. However, for non-SMS-managed data sets, the following differences can occur when DFSMSHsm is used on a single DFSMSHsm-host or a multiple DFSMSHsm-host configuration in a JES3 environment using the benefits of JES3 data set reservation as provided by JES3 setup.

- Volumes are considered in terms of data set pools, volume pools, and general pools. The general pool, configured by DFSMSHsm, is the default pool that contains primary volumes that are online and permanently resident or reserved and that have the automatic recall attribute. From this set, volumes are selected for undirected recall.
- For migrated data sets, JES3 associates the requested data set with a subset of five or fewer candidate pool volumes supplied by DFSMSHsm. JES3 treats the data set as a multiple-volume data set.
- For data sets that have not migrated, JES3 associates the requested data set to the serial number of the primary volume on which the data set resides. DFSMSHsm prevents migration of the data set for today and three more days.

- DFSMSHsm recalls the migrated data set to a specific volume serial number in the candidate volume list when the data set is opened or to the best candidate primary volume in the subset of five or fewer pool volumes when the data set is allocated.

This discussion is concerned only with the function of delaying migration. Associating migrated data sets with volumes for recall is described in [“Recall” on page 119](#), a discussion of automatic recall for non-SMS-managed data sets.

During the post-scan phase, JES3 calls its locate module to locate the cataloged data sets. When the JES3 locate routine finds a primary volume serial number for a requested data set in the computing system catalog, JES3 associates the specified data set with that volume serial number. Asynchronously with the JES3 locate routine, DFSMSHsm passes a management work element (MWE) to prevent migration of the requested data set for up to three days (default value) by recording the primary volume serial number and a prevent-migration date in the MCDS data set (MCD) record. The prevent-migration date is the last day that migration is prevented. If the job does not run within a prevent-migration period, the prevent-migration date expires, and the data set can migrate automatically or by command.

A JES3/DFSMSHsm migration window exists between the asynchronous processing of the JES3 locate routine and the DFSMSHsm management work element. If migration of the data set occurs in the JES3/DFSMSHsm migration window, DFSMSHsm modifies the MCD record so that the recall of the migrated data set occurs to that primary volume serial number that was returned to JES3. If you enter a command to migrate a data set associated with a volume that has a current prevent-migration date, DFSMSHsm fails the command. A message is issued to indicate that JES3 has an outstanding request for the data set, and migration is prevented until the prevent-migration date has expired.

The prevent-migration period is a default value of 3 in the management communication vector table (MCVT). The default value is added to the current date to create the prevent-migration date and the hold-volumes date in the MCD record. You can change this value. Although the value can range from 0–255, the value should be consistent with the maximum time a job can exist in JES3 before it is run.

JES3 directed-recall and prevent-migration requests are processed as DFSMSHsm commands. In a multiple DFSMSHsm-host environment, these requests may not be processed until the completion of a command that is currently being processed. Normally, the delay has little effect on the time required to process the JES3 request. However, some performance degradation may be noticed if the DFSMSHsm command processor is waiting for the availability of a resource.

Related reading

For more information about using the PATCH command for the prevent-migration period, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Performing extent reduction on the volume

After DFSMSHsm performs migration for the non-SMS-managed volumes, it performs extent reduction. Extent reduction for non-SMS-managed data sets is performed in the same way as for SMS-managed data sets.

Note: DFSMSHsm does not do extent reduction if you have specified SETSYS TAPEMIGRATION(DIRECT) and you have not specified an ML1 volume. The extent-reduction process would cause contention between migration and recall for the target tape ML2 volume.

Installation exits during automatic primary space management

Programming Interface Information

Control is passed to installation exits during automatic primary space management for non-SMS-managed data sets in the same way as for SMS-managed data sets. Delete by age and delete if backed up are not applicable to SMS-managed data sets; therefore, ARCADEXT is not called in processing SMS-managed data sets.

Pass 2 processing example: Figure 17 on page 118 shows the results of automatic primary space management after the second pass. Data set GROUP1.SAMPLE.A1 migrated because its inactive age is

greater than the minimum migration age specified on the ADDVOL command. Because it is small enough that compaction will probably reduce it to less than 110KB (1KB = 1024 bytes), it migrated to an SDSP data set. Data set GROUP2.SAMPLE.B1 did not migrate because it has not passed the minimum migration age.

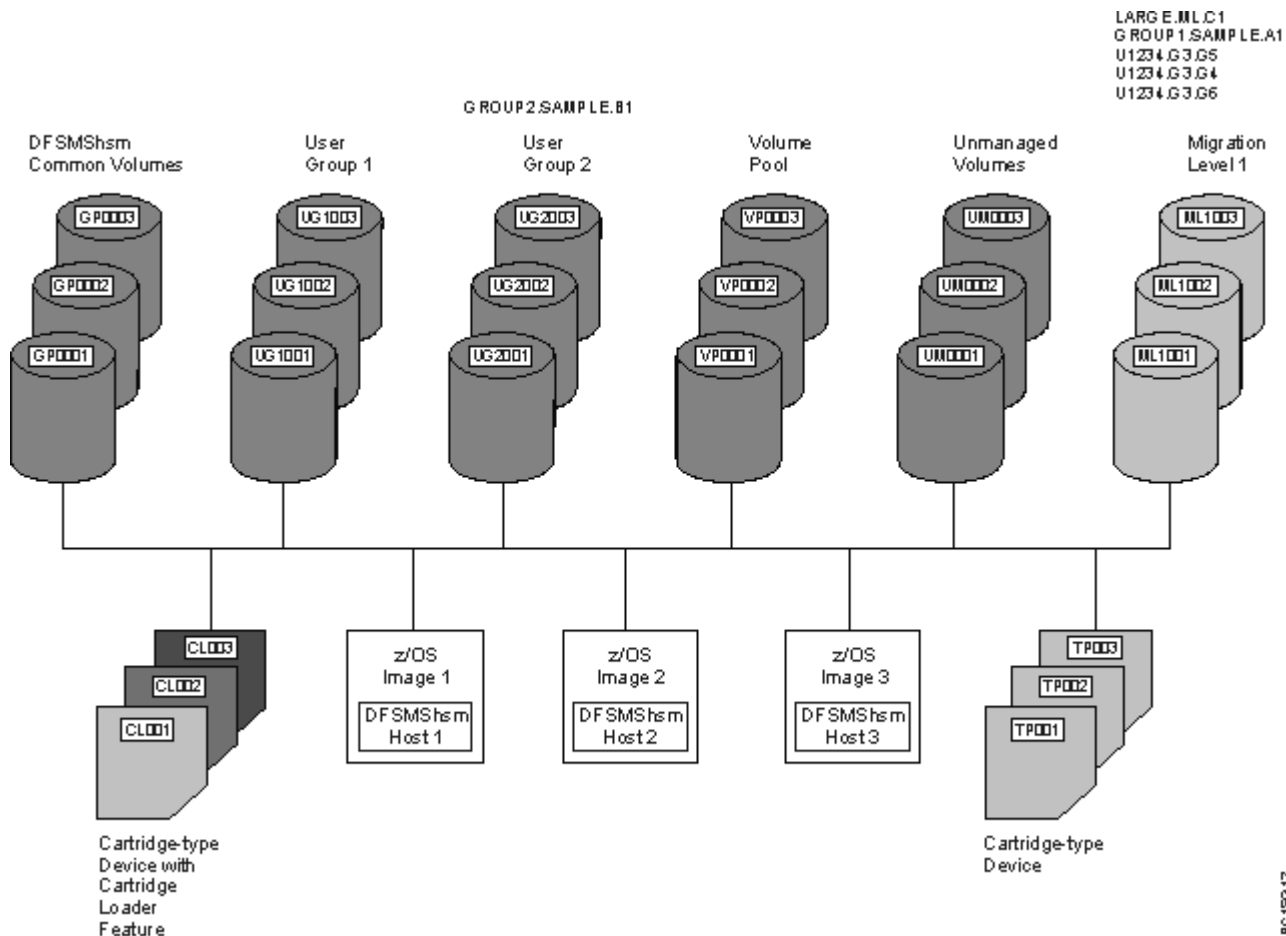


Figure 17. Example System of Non-SMS-Managed Volumes after Primary Volume Management Pass 2

Data sets U1234.G3.G4 and U1234.G3.G5 migrated because of their inactive ages. Assuming that both were on the same volume, U1234.G3.G5 migrated first because of the priority algorithm.

Data sets U2345.G3.G6 and LARGE.ML.C1 migrated because of their inactive ages. Observe that although U2345.G3.G6 has exceeded the MIGRATIONLEVEL1DAYS parameter specified in the SETSYS command, it migrated to ML1. If it is not recalled, it migrates to ML2 the next time that level 1 to level 2 migration is performed.

Observe also that LARGE.ML.C1 is of a size that would put it in management class LARGEMC so that it would migrate directly to tape if it were SMS-managed. However, the example leaves it migrating to ML1 DASD because direct-to-tape migration for non-SMS-managed data sets has not been provided. Its migration could be prevented, or it could be directed to tape by using the migration installation exit (ARCMDEXT) that is called for each data set that is migrated.

End Programming Interface Information

Related reading

For more information about the ARCMDEXT installation exit, see [z/OS DFSMS Installation Exits](#).

Automatic interval migration

For non-SMS-managed volumes, interval migration occurs only from primary volumes that possess the following conditions:

- Have defined thresholds of occupancy
- Have exceeded their high threshold of occupancy
- Have the primary volume attribute of automatic migration
- Have been added for automatic migration in a DFSMSHsm host that has interval migration specified
- Have the space management technique of migration

DFSMSHsm performs interval migration for non-SMS-managed data sets in the same way as for SMS-managed data sets. However, after DFSMSHsm finishes migrating eligible data sets on non-SMS-managed volumes during interval migration, it performs extent reduction for the volumes.

Recall

Recall for non-SMS-managed data sets that are not recalled to SMS-managed volumes differs from that for SMS-managed data sets primarily in that, for non-SMS-managed data sets, DFSMSHsm selects the volumes to which the data set is recalled. For recall target volume selection, DFSMSHsm considers volumes in the following terms:

- Data set pools
- Volume pools
- General pool (JES3), a single pool that DFSMSHsm configures

Related reading

- For more information about how to define pools, see [“Defining pools of volumes” on page 105](#).
- For more information about recall requests and the common recall queue, see [“Common recall queue” on page 129](#).

Recall volume selection

When DFSMSHsm receives a non-directed recall request (either by command or by allocation with no volume specified), it builds a list of up to five primary volumes with the most available space. The space on the volumes is based on the last space check performed on the primary volumes. DFSMSHsm selects the primary volumes as follows:

1. If the data set being recalled has a high-level qualifier that matches that of a data set pool, the five volumes come from the data set pool.
2. If the volume from which the data set migrated is part of a volume pool, the candidate volumes come from the volume pool.
3. If the data set is not restricted to be recalled to a user-defined pool, DFSMSHsm selects volumes that were assigned to the default pool. Whether a volume is selected is based on:
 - The subparameters selected for the RECALL parameter of the SETSYS command
 - The volumes having the AUTORECALL subparameter specified in the ADDVOL command and having MIGRATE as the space management technique
 - The volumes being online
 - The use attribute of the volume
 - Attribute of storage for the RECALL(ANYSTORAGEVOLUME) parameter
 - Attribute of storage, public, or private for the RECALL(PRIVATEVOLUME) parameter

In building the list of candidate volumes from the default pool, DFSMSHsm considers the recall attributes described in [“Recall attributes” on page 104](#) and the LIKE|UNLIKE subparameters specified with the

RECALL parameter of the SETSYS command. If LIKE is specified, DFSMSHsm builds the list only from volumes with recall attributes that match the recall attributes of the volume from which the data set migrated. If fewer than five volumes have like attributes, the list of candidate volumes contains fewer than five volumes.

If UNLIKE is specified, DFSMSHsm builds the list from all the volumes in the default pool in the following order:

1. If five or more volumes exist with like recall attributes, all the volumes in the candidate list have recall attributes that match those of the volume from which the data set migrated.
2. If not enough like volumes exist, DFSMSHsm selects as many volumes with recall attributes that match those of the volume from which the data set migrated as possible, then selects unlike volumes in the following order:
 - a. Volumes with matching backup attributes (AUTOBACKUP|NOAUTOBACKUP and BACKUPDEVICECATEGORY)
 - b. Volumes with matching migration attributes (AUTOMIGRATION|NOAUTOMIGRATION)
 - c. Volumes with no matching attributes

Within this group, each subgroup is placed in order by the amount of available space on the volume. Thus, the volumes with like attributes are placed (in order by free space) ahead of the volumes with unlike attributes.

DFSMSHsm never requires matching recall volume attributes for data sets under the following conditions:

- A data set being recalled to a user-defined pool
- A data set migrated by command from a volume with the DELETEBYAGE management technique
- A data set migrated by command from a volume not managed by DFSMSHsm
- A data set that migrated from a primary volume that was added to only one host in a multiple DFSMSHsm-host environment

A primary volume managed by DELETEIFBACKEDUP or DELETEBYAGE space management techniques will never be selected by DFSMSHsm as the target of a nondirected recall.

Calling the recall exit

Programming Interface Information

After the list of candidate volumes has been assembled, DFSMSHsm passes control to the recall installation exit (ARCRDEXT) if the installation exit is active. DFSMSHsm passes the complete list of eligible volumes, not just the five that it has selected, to the installation exit. The five volumes that DFSMSHsm has selected are at the top of the list. The installation exit can change the volume selection so that different volumes are the five volumes at the top of the list, thereby changing the volumes that DFSMSHsm uses for the recall. Another point of divergence from the processing for SMS-managed data sets is that control is not passed to the installation exit during the recall of SMS-managed data sets.

Note: In a common recall queue environment, the ARCRDEXT exit is invoked by the host that selects the request for processing.

End Programming Interface Information

Allocating recalled data sets to volumes

After the list of candidate volumes has been assembled from the installation exit, DFSMSHsm or DFSMSdss requests that the data set be allocated to the first volume on the candidate list. If the allocation fails for lack of space, DFSMSHsm or DFSMSdss requests allocation to the next volume on the list. If this allocation also fails for lack of space, DFSMSHsm or DFSMSdss tries the next volume, and proceeds down the list until either a volume with adequate space is found, or all the volumes on the list have been tried.

When DFSMSHsm is the data mover, a recall can fail even though enough total space exists on the volume if the space is too fragmented to allocate the total space for the data set in five extents.

Automatic recall during non-VSAM data set opening

If the user specifies both volume and unit information during allocation of a non-SMS-managed volume or if the allocation process is not done, a migrated data set will still be on a migration volume when the open routine tries to open the data set. When the non-VSAM open routine does not find the data set VTOC entry for a non-VSAM data set, DFSMSHsm intercepts the abnormal end and tries to recall the data set. Enough space must exist for DFSMSHsm to recall the data set to the volume the open routine specifies. Otherwise, the recall fails and the open routine abnormally ends.

If a user allocates an uncataloged data set and mistakenly specifies the wrong volume, the user receives a VTOC entry-not-found abnormal end. However, if a migrated data set exists with the same data set name as the data set the user is allocating, DFSMSHsm can recall the migrated data set to a user's volume. The user might be able to open the data set, if it is unprotected, even though it could belong to someone else.

Automatic recall during VSAM data set opening

If the user specifies both volume and unit information during allocation of a non-SMS-managed volume or if the allocation process described in “Recall volume selection” on page 119 is not done for the data set, a migrated data set to be opened will still be on a migration volume when the open routine tries to open the VSAM data set. The VSAM open routine issues a catalog locate request to get information about the cluster from the catalog. DFSMSHsm intercepts the locate request, recalls the data set to the primary volume it migrated from, and returns the catalog information that the VSAM open routine requested. Enough space must exist for DFSMSHsm to recall the data set to the volume the open routine specifies; otherwise, the VSAM open routine fails.

Automatic recall for data set command scratching

A non-SMS-managed, non-VSAM data set could be on a migration volume when it is about to be scratched. If a scratch is requested for a data set from a volume with a volume serial number of MIGRAT, DFSMSHsm translates the request into a request to delete a migrated data set.

If the scratch request does not specify MIGRAT and results in a VTOC entry-not-found condition, the data set still could have migrated. If the scratch routine cannot find the VTOC entry for the data set on the specified volume, DFSMSHsm receives a request to delete the data set if the volume specified with the VTOC entry-not-found condition is the same volume the data set migrated from last. If both of these conditions are met, DFSMSHsm processes the scratch request; otherwise, the scratch fails.

A data set is not scratched or deleted by DFSMSHsm if the expiration date of the data set in the data set VTOC entry is not reached and if the purge bit in the scratch parameter list is not on. When DFSMSHsm receives the request to delete the migrated data set, it recalls the data set, and the scratch request continues for the recalled data set. The volume that the scratch request specified is used for the data set recall must contain enough available space to recall the data set. Otherwise, the recall and the scratch request for the data set fail.

JES3 considerations

When the JES3 converter/interpreter is processing a data set that is migrated, it requests a list of recall volumes from DFSMSHsm. DFSMSHsm selects a list of recall volumes as described in “Recall volume selection” on page 119 and returns the list to the JES3 converter/interpreter. DFSMSHsm also records the list of volumes in the MCD record. For a subsequent request for that data set from the JES3 converter/interpreter, DFSMSHsm returns the same list of volumes.

When DFSMSHsm returns the list of volumes, the JES3 converter/interpreter associates those volumes with the data set. Because DFSMSHsm returned five or fewer volumes, JES3 associates the data set as a multiple-volume data set.

When volumes are selected from a pool, DFSMSHsm uses a count of the number of migrated data sets associated with each pool volume in the general pool and the available volume space determined by the

last space check. The migrated data sets are those that have been directed to a volume but have not been recalled yet. Therefore, the algorithm spreads the association of migrated data sets to all volumes in the pool.

DFSMSHsm holds each subset of five or fewer volumes today and three more days by recording in the MCD record a hold-volumes date and the five or fewer volume serial numbers. The *hold-volumes date* is the last date that DFSMSHsm restricts recalls for the data set to the five or fewer volumes.

DFSMSHsm recalls migrated data sets when MVS allocates (nonspecific request and the volume serial number is MIGRAT) or opens (specific request and the VTOC entry is not found) the data set on a volume managed by DFSMSHsm. For a subset of five or fewer pool volumes—either data set, volume, or general—DFSMSHsm directs the recall of the migrated data set to one of the five or fewer volumes. DFSMSHsm fails the recall request if not enough space is available on any of the volumes.

If you direct the recall of a migrated data set to a DFSMSHsm-managed volume other than the subset associated by JES3 and before the hold-volumes date expires, DFSMSHsm fails the recall. A message is issued to indicate that the volume serial number you requested is not on the JES3 list. If you enter a command to delete a migrated data set before the hold-volumes date expires, DFSMSHsm fails the command. A message is also issued to indicate why the delete request failed.

If space is not made available on these volumes and you must recall the data set before the hold-volumes date expires, you can reset the hold-volumes date by issuing the command:

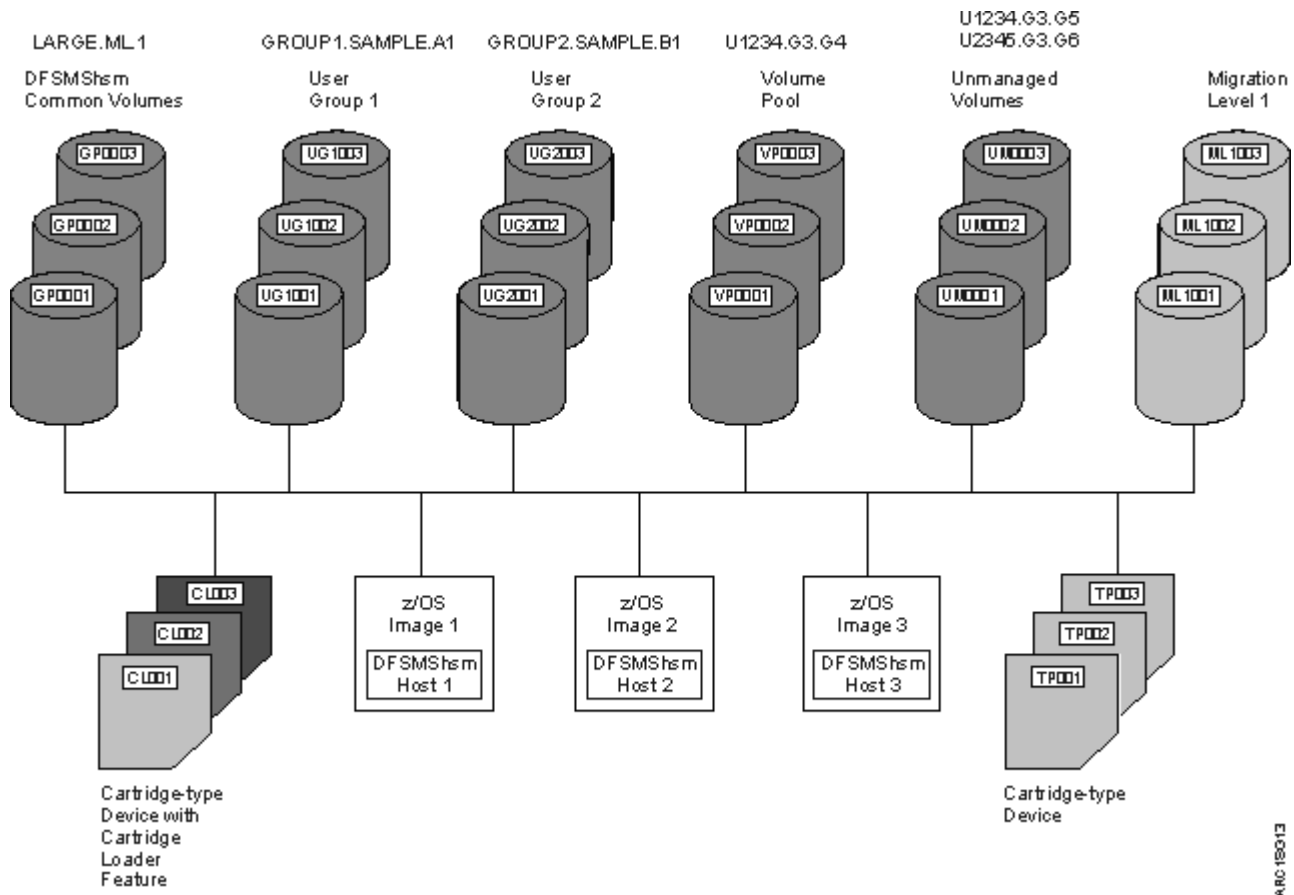
```
HSEND CMD FIXCDS D dsn PATCH(X'60' X'000000')
```

Be sure that the appropriate volume and pool configuration of the z/OS image that will eventually process a job is described to the z/OS image where the JES3 converter/interpreter locate request is initiated during JES3 setup of the job. The DFSMSHsm hosts in those z/OS images must have the same SETSYS RECALL subparameter specified.

Recall example

Figure 18 on page 123 shows the results of recalls (either automatic or by command) that allow DFSMSHsm to select the volume to which the data set is recalled. Because GROUP1.SAMPLE.A1 has the high-level qualifier of GROUP1, for which a data set pool that contains the volumes in User Group 1 was defined, it is recalled to one of the volumes in that user group. Because U1234.G3.G4 migrated from a volume in Volume Pool and a volume pool that contains those volumes was defined, U1234.G3.G4 returns to one of the volumes in the pool. LARGE.ML.1 returns to one of the common volumes (general pool in JES3) because neither of the following applies:

- It does not have the high-level qualifier to direct it to one of data set pools defined for the user groups.
- It did not migrate from one of the volumes in the volume pool.



ARC1801E

Figure 18. Example System of Non-SMS-Managed Volumes after Undirected Recall of Data Sets

Command space management

For non-SMS-managed storage, you can perform the same command space management as you can for SMS-managed storage:

- Delete migrated data sets
- Migrate data sets
- Migrate level 0 volumes
- Migrate level 1 volumes
- Recall data sets

For non-SMS-managed data sets, migrating level 0 volumes works somewhat differently. In addition, for non-SMS-managed data sets you can:

- Migrate from a single primary volume all data sets not used within the number of days you specify
- Delete by age from a single primary volume all data sets not used within the number of days you specify
- Delete if backed up from a single primary volume all data sets not used within the number of days you specify
- Migrate data sets from all non-SMS-managed primary volumes
- Control migration of individual data sets, groups of data sets, or volumes
- Recall a data set to a specific volume
- Recall an SMS-managed data set to a non-SMS-managed volume

Command volume space management

Two forms of the MIGRATE command control volume space management for DFSMSHsm-managed volumes:

```
MIGRATE VOLUME(valid)
MIGRATE VOLUME(valid DELETEBYAGE(days) |
DELETEIFBACKEDUP(days) | MIGRATE(days))
```

The MIGRATE VOLUME command cannot be used for ML1 volumes.

For the first form, DFSMSHsm performs space management for the volume in the same way as if the volume were being processed by automatic primary space management. That is, it performs all the functions described in [“Automatic primary space management”](#) on page 111 for the volume, using the parameters specified in the ADDVOL command for the volume. Thus, the management technique and the inactive days specification from the ADDVOL command apply to this processing of the volume.

For the second form of the MIGRATE command, the processing described in [“Automatic primary space management”](#) on page 111 is performed, but the management technique and the number of inactive days is taken from the MIGRATE command. Thus, you can apply different technique and inactive day specifications to the volume when you do command volume migration.

Related reading

For more information about the commands to use for volume space management of DFSMSHsm-owned volumes, see [Chapter 13, “Managing DFSMSHsm media,”](#) on page 527.

Command volume space management (primary)

You can cause all non-SMS-managed primary volumes to be processed by issuing the following command:

```
MIGRATE PRIMARY
```

This command has the same effect for all level 0 volumes that have been added to the system with an ADDVOL command as if the primary volume space management part of automatic primary space management for non-SMS-managed volumes had begun.

Controlling migration of data sets and volumes

You can use the SETMIG command to control migration for specific data sets, groups of data sets, or volumes. The control you can exercise is to allow all migration, to allow migration only by command, or to allow no migration. For example, if you issue the following command:

```
SETMIG VOLUME(GP0002) NOMIGRATION
```

no migration can occur from volume GP0002 until the SETMIG command is reissued to allow migration or until DFSMSHsm is reinitialized. The SETMIG command for volumes and data sets is not normally included in the ARCCMDxx member. The SETMIG command for levels (groups of data sets) is typically included in the ARCCMDxx member.

Recalling a data set to a specific volume

You can recall a non-SMS-managed data set to a specific non-SMS-managed volume. For an example, see [Figure 17 on page 118](#). If data set GROUP1.SAMPLE.A1 were recalled with either an automatic or command undirected recall, it would return to one of the volumes in User Group 1. However, if you issue the following command:

```
RECALL GROUP1.SAMPLE.A1 UNIT(3390) VOLUME(VP0003)
```


the data set is recalled to volume VP0003. If the ACS routine converts the data set to an SMS-managed data set (returns a storage class), SMS selects the volume to which the data set is recalled.

Recalling an SMS-managed data set to a non-SMS-managed volume

You cannot recall an SMS-managed data set to a specific SMS-managed volume. You can, however, recall an SMS-managed data set to a specific non-SMS-managed volume and remove the data set from SMS management. For an example, see [Figure 11 on page 80](#) and [Figure 17 on page 118](#). If you issue the following command:

```
RECALL GDSMC.G3.V5 UNIT(3390) VOLUME(GP0003) FORCENONSMS
```

the data set is recalled to GP0003 and is no longer under SMS management.

Recalling an SMS-managed BDAM data set to a non-SMS-managed volume

You can direct the recall of an SMS-managed BDAM data set to a non-SMS volume with a smaller track size than the volume from which it was migrated, using the DAOPTION and the FORCENONSMS parameters of the RECALL command. If, for instance, data set DA00001.V1 was migrated from a 3390 volume, you could recall that data set to a 3380 volume (GP3380) by issuing the following command:

```
RECALL DA00001.V1 UNIT(3380) VOLUME(GP3380) -  
  DAOPTION(RELBLK) FORCENONSMS
```

The data set is recalled to GP3380, no longer under SMS management, and is ready to be accessed through relative block accessing.

Recalling an SMS-managed sequential extended-format data set to a non-SMS-managed volume

You cannot recall an SMS-managed sequential extended-format data set to a non-SMS-managed volume by modifying your ACS routine to filter the data set to such a volume. However, if you issue the following command:

```
RECALL dsname FORCENONSMS
```

you can recall an SMS-managed sequential extended-format data set to a non-SMS-managed volume as a data set that is not sequential extended-format.

Chapter 4. Other space management considerations

In addition to the aspects of space management discussed in [Chapter 2, “Space management of SMS-managed storage,”](#) on page 23 and [Chapter 3, “Space management of non-SMS-managed storage,”](#) on page 99, you need to consider the following topics:

- [“Space management of extended address volumes \(EAVs\)”](#) on page 127
 - [“Track management for SMS-managed volumes”](#) on page 128
 - [“Track management for non-SMS-managed volumes”](#) on page 128
 - [“Controlling allocations to extended address volumes \(EAVs\)”](#) on page 129
 - [“Recommendations for using extended address volumes \(EAVs\)”](#) on page 129
- [“Common recall queue”](#) on page 129
- [“Processing priority of recalls and deletions”](#) on page 135
- [“Migration volume contention priorities”](#) on page 136
- [“Specifying the desired number of ML2 partial tapes”](#) on page 141
- [“Preventing the processing of data sets during space management”](#) on page 149
- [“Deferred generation data sets”](#) on page 149
- [“Security considerations for automatic space management”](#) on page 149
- [“Access method services considerations”](#) on page 152
- [“GUARANTEED SPACE attribute considerations”](#) on page 152
- [“DFSMSHsm retrieval of management classes and storage groups”](#) on page 154
- [“Deletion of dynamic I/O device considerations”](#) on page 155

Space management of extended address volumes (EAVs)

An extended address volume (EAV) is defined to be any volume that has more than 65,520 cylinders. Volumes that are configured with 65,520 cylinders or less are defined to be non-EAVs. An EAV will consist of two areas of managed space. The first 65,520 cylinders of an EAV will be considered the *track-managed space*. The last cylinder in this managed space is at cylinder address 65,519. The space above 65,520 cylinders will be considered the *cylinder-managed space* on the volume. The first cylinder in this managed space is at cylinder address 65,520.

Data set size is a determining factor for the order in which data sets are migrated. Managing an entire extended address volume (EAV) with one threshold is not sufficient because average data set size in the cylinder-managed space will likely exceed the average data set size in track-managed space. This will result in data sets allocated in the cylinder-managed space being migrated at a higher rate than data sets allocated in the track-managed space, potentially causing no space in the track-managed space being freed. To prevent this, a unique threshold for the track-managed space (track-managed threshold) is used to manage the free space separate from the free space on the entire volume (volume-level threshold).

DFSMSHsm examines both volume level thresholds and track-managed thresholds for L0 EAVs. If either threshold has been exceeded then migration eligibility will be performed for data on the volume. If track-managed threshold values are not specified, the default is to use the volume-level threshold values. Data set eligibility will be further qualified by the location of the first three extents of the data set. If any of the first three extents are allocated in track-managed space the data set will be eligible to be processed when the track-managed space or entire volume space is managed. If none of the first three extents are allocated in track-managed space, the data set will be eligible to be processed when the entire volume space is being managed. [Table 12 on page 128](#) summarizes the eligibility criteria for migration.

Table 12. Eligibility Criteria for Migration

Track-managed threshold exceeded	Volume threshold exceeded	Data set selection
Yes	No	Only data sets with one or more of the first three extents allocated in track-managed space
No	Yes	All data sets
Yes	Yes	All data sets
No	No	Data sets are not selected

Space management (interval migration and primary space management) will be considered complete after DFSMSHsm determines both thresholds have been met. This applies only to hourly (interval migration) space management of non-SMS LO EAVs and when the AUTOMIGRATE attribute is not equal to I (immediate) for SMS-managed EAVs. Hourly space management (interval migration) of SMS-managed EAVs having the AUTOMIGRATE=I attribute in storage groups will be performed if either the midpoint between the high and low volume threshold values is met or if the midpoint between the high and low track-managed threshold values is reached or exceeded. Daily space management (primary space management) will be performed regardless of the high volume threshold and high track-managed threshold values.

Track management for SMS-managed volumes

DFSMSHsm will use the track-managed low and high threshold values and the volume level low and high threshold values associated with each storage group.

Track management for non-SMS-managed volumes

The ADDVOL command has an optional keyword TRACKMANAGEDTHRESHOLD(*high low*).

- Alias: TMT
- TMT is only valid with the PRIMARY keyword
- *High* and *low* are required parameters

TRACKMANAGEDTHRESHOLD can be used to define the high and low thresholds for the track-managed space of a non-SMS PRIMARY EAV. If TRACKMANAGEDTHRESHOLD is not specified, then the corresponding values for THRESHOLD will be used. If no threshold values are specified, then all threshold values will use the default values. If TRACKMANAGEDTHRESHOLD is specified for a non-EAV, the ADDVOL command will fail. For example:

```
ADDVOL PRIM01 PRIMARY UNIT(3390) TRACKMANAGEDTHRESHOLD(80 40) THRESHOLD(85 60)
```

In the above example the parameter TRACKMANAGEDTHRESHOLD(80 40) defines a high track-managed threshold of 80 percent and low track-managed threshold of 40 percent. If TRACKMANAGEDTHRESHOLD is not specified, then the corresponding values for THRESHOLD will be used.

If no THRESHOLD keyword values are specified, then all threshold values will use the default values. The default high value is 100, and the default low value is 0. Therefore, a primary volume has no valid default thresholds in this case. If the attributes of a volume are being changed but track-managed space threshold values are not specified, the track-managed space thresholds are not changed. If, however, the new high track-managed threshold is less than or equal to the low track-managed threshold or is equal to 100, the volume threshold values will be used as track managed threshold values.

For more information, see [“TRACKMANAGEDTHRESHOLD: Specifying the high and low threshold for the track-managed space of a non-SMS level 0 extended address volume”](#) on page 710 in Chapter 26, [“ADDVOL command: Adding or changing DFSMSHsm volumes,”](#) on page 697.

Controlling allocations to extended address volumes (EAVs)

You can use the USEEAV parameter for a system on either the SETSMS command or in the IGDSMSxx parmlib member to specify whether SMS is to select extended address volumes during volume selection processing. This check applies to both new allocations and to extending data sets to a new volume. You can enter an operator command to change the value of this keyword.

Related reading

- There are a number of important implications to specifying or defaulting to USEEAV=NO, see [Modifying the SYS1.PARMLIB IGDSMSxx member in z/OS DFSMSdfp Storage Administration](#).
- The setting for USEEAV is stored in the SSI vector table. To look at the SSI vector table, see [Formatting the SSI vector table in z/OS DFSMSdfp Diagnosis](#).
- See also [SETSMS command to alter the setting of USEEAV in z/OS DFSMSdfp Storage Administration](#).

Recommendations for using extended address volumes (EAVs)

- You can use EAVs as backup, ML1, and ML2 volumes. You can enable allocation of backup, ML1, and ML2 data sets in cylinder-managed space using SETSYS parameter USECYLINDERMANAGEDSPACE. See [“USECYLINDERMANAGEDSPACE: Specifying whether to allow use of cylinder managed space for migration copies and backup versions” on page 1202](#).
- You should evaluate the SMS and non-SMS breakpoint values to limit the over allocation of small data sets in the cylinder-managed space on backup, ML1, and ML2 extended address volumes. Depending on the breakpoint value, over allocation might still occur for large data sets, however, the percent of over allocation is small in comparison to the total disk space request. Small data sets should be allocated in track-managed space and large data sets allocated in cylinder-managed space to limit over allocation. For more information, see the topic about [Extended Address Volumes in z/OS DFSMS Using Data Sets](#).
- EAVs configured as L0 volumes should **not** be used as a repository for small data sets. Doing so will negatively impact performance.

Common recall queue

The common recall queue (CRQ) is a single recall queue that is shared by multiple DFSMSHsm hosts. This queue is implemented through the use of a coupling facility (CF) list structure. The CRQ balances the recall workload across multiple address spaces and enables optimization for priority and single-tape mounts. In some parallel sysplex configurations, all hosts are not connected to all devices. The CRQ enables hosts that are not connected to a device, like a tape drive, to place recall requests on the CRQ. Once on the CRQ, the request can be selected by hosts that are connected to the device.

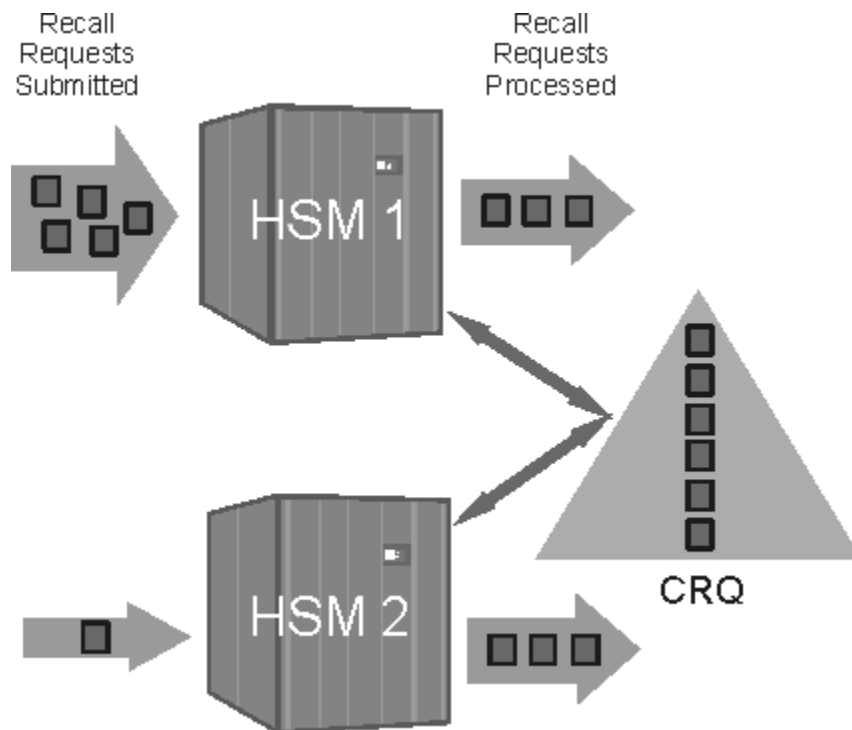


Figure 19. The Common Recall Queue

As Figure 19 on page 130 shows, a CRQplex consists of one or more DFSMSHsm hosts that are connected to the same CRQ. There may be multiple CRQplexes within a single HSMplex. In general, each HSMplex maintains a unique CRQ.

This CRQ support provides the following enhancements to recall processing:

Workload Balancing

Alternating with all other participating hosts in the CRQplex, each host attempts to select requests from the CRQ. Each host selects requests while it has available recall tasks and the CRQ is not empty.

Priority Optimization

Because all recall requests are placed and sorted on a common queue, the highest priority outstanding requests in the CRQplex are selected before lower priority requests. This prioritization occurs regardless of which system the requests originated on.

Tape Mount Optimization

When a recall task mounts an ML2 tape to perform a recall, that task processes all outstanding single tape recall requests in the CRQ that require that tape. Thus, a single tape mount recalls multiple recall requests that require the same tape, even if those requests originated on different hosts.

Request Persistence

Outstanding recall requests on the CRQ are maintained even when the originating DFSMSHsm host shuts down or fails, or the originating system fails. Additionally, those requests are eligible for processing by other active hosts, even while the original host is unavailable.

Flexible Configurations

The CRQ support enables configurations in which not all hosts are connected to all devices, and in which not all hosts process recalls. For example, you can configure a host that is not connected to any tape drives to process only recalls that do not require ML2 tapes. Also, you can configure the HSMplex so that all hosts accept recall requests and place them on the CRQ, but only a subset of the hosts actually process the requests.

Related reading

For more information about how to configure each host in an HSMplex for a CRQ environment, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Connecting to a common recall queue list structure

Before the CRQ function can be used, an appropriately defined list structure must be configured in the active coupling facility resource manager (CFRM) policy. After configuring the CFRM policy, update your DFSMSHsm ARCCMDxx member to include SETSYS COMMONQUEUE(RECALL(CONNECT(*basename*))).

When SETSYS COMMONQUEUE(RECALL(CONNECT(*basename*))) is specified in ARCCMDxx, DFSMSHsm connects to the specified structure during startup processing. If the SETSYS command is issued after startup, DFSMSHsm connects to the structure and then moves any requests on the host's local recall queue to the CRQ.

If the attempt to connect fails due to an environmental error, DFSMSHsm retries the connection when the coupling facility signals that the structure is available. Even though the connection may be unsuccessful, the DFSMSHsm host is associated with the specified CRQ until a SETSYS COMMONQUEUE(RECALL(DISCONNECT)) command is issued.

A CRQ cannot be shared between separate HSMplexes. To enforce this, each CRQ is associated with the HSMplex that allocated it. DFSMSHsm fails any attempt to connect to a CRQ that is not associated with the HSMplex in which a connection request originates.

Related reading

For more information about configuring the list structure, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Populating the common recall queue

A host that is participating in CRQplex-wide recall processing attempts to place all of its recall requests onto the CRQ. Requests are placed on the CRQ if the following conditions are true:

- The host has a successful connection to the CRQ.
- The hold status of the host and CRQ eligibility allows the request to be placed on the CRQ. The following conditions must be true to allow placement on the CRQ:
 - The host must not have HOLD COMMONQUEUE(RECALL(PLACEMENT)) or higher level of Hold in effect.
 - For WAIT requests, at least one host that is connected to the CRQ must be eligible to process the request. This is what is meant by “CRQ eligibility”. This requirement prevents a WAIT request from being placed onto the CRQ and waiting indefinitely to be processed. For example, if HOLD RECALL(TAPE) is in effect on all hosts connected to the CRQ, then all WAIT recall requests that require an ML2 tape will fail. If at least one host does not have HOLD RECALL(TAPE) in effect, then those requests are placed.
- The CRQ is less than 95% full. Or, if it recently exceeded the 95% threshold, it has dropped back below 85%.

If recall requests are unexpectedly not placed onto the CRQ, issue the QUERY ACTIVE command and examine message ARC1540I. It will indicate which of the above conditions is not being met.

Any request that cannot be placed onto the CRQ is placed on the local queue of the originating host.

Recall requests that can be placed onto the CRQ are sorted by WAIT or NOWAIT status and priority. WAIT requests are always a higher priority than NOWAIT requests. Requests of the same priority are sorted in FIFO order, with the exception that NOWAIT requests of priority 50 are subsorted with requests from multiple users. This subsorting prevents a single user with many requests from blocking other users' requests for an extended period of time.

The host that initiates the recall request invokes user exit ARCRPEXT, if the exit is enabled. The processing host invokes user exit ARCRDEXT, if the exit is enabled. You can ensure consistent processing by using common RD exits throughout the CRQplex. The RP exit may be used to assign requests that originate on one host a higher or lower priority in relation to the priorities set by RP exits on other hosts.

The default priority for these requests is 50. Refer to *z/OS DFSMS Installation Exits* for a description of these user exits.

The SETSYS EMERGENCY command does not affect the placement of recall requests onto the CRQ. This allows a host that is in emergency mode to place its requests onto the CRQ and enables other active hosts to process those requests. Use the HOLD COMMONQUEUE(PLACEMENT) command to prevent recalls from being placed onto the CRQ when the emergency mode is in effect.

Selecting requests from the common recall queue

Alternating with other hosts, each host attempts to select requests from the CRQ as long as the host has available recall tasks and the CRQ is not empty. Hosts select the highest priority request that they are capable of processing. Also, hosts select recall requests that are on the host's local queue before requests on the CRQ that are of the same or lower priority. (Requests may be on a host's local queue if the requirements for placement, as mentioned in the previous topic, have not been met.)

When a host selects a request that requires a single ML2 tape, that host processes all single tape requests on the CRQ from that tape. After processing each request from the tape, the processing host selects the next highest priority request that requires that tape. Requests with the same priority are processed in first-in-first-out order. After processing all the requests and after a short wait for any additional requests, the tape is demounted.

If recall requests are unexpectedly not being selected for processing, issue the QUERY ACTIVE command and examine message ARC1541I. A status of anything other than CONNECTED or a hold level of anything other than NONE are probable reasons why certain recall requests are not being selected.

Another cause for certain recall requests not to be selected is that they require an unavailable ML2 tape. If the ML2 tape does not become available within 30 minutes, these requests will time out.

Processing recall requests on the common recall queue

A host processes a request selected from the CRQ just like a request that is submitted on its own system, with two exceptions. First, messages that are routed to TSO are sent to the original host to issue to the appropriate user. Second, when the processing host completes a recall request, it notifies the original host which then posts any WAIT requests as complete.

The processing host issues console and operator messages; messages directed to the DFSMSHsm logs go to the logs for the processing host. Only the processing host creates and stores functional statistics records (FSRs). The originating host, if different from the processing host, does not have an FSR for the recall. The processing host accumulates the daily statistics record (DSR) statistics, and the DSR stores recall statistics for the entire HSMplex.

A recall request can be canceled until the point that it has been selected for processing by a host in the HSMplex. The cancel request must be submitted on the same host that originated the recall request that is to be canceled.

Note: The DFSMSHsm host that places the recall request on the CRQ is the host that checks the user's authority to issue the recall command. Authority checking is performed using the DFSMSHsm AUTH command or with the RACF FACILITY class. In a common recall queue environment, there is the possibility that when a recall command is placed on the CRQ (which is not checked by RACF), a DFSMSHsm host can select the same recall request. This same recall request is not checked by RACF, because RACF checking is performed before placing the recall on the CRQ.

Disconnecting from the common recall queue

To disconnect a host from the CRQ, use the optional parameter SETSYS COMMONQUEUE(RECALL(DISCONNECT)). When the DISCONNECT command is received, the DFSMSHsm host disconnects itself from the CRQ by performing the following actions:

- Directing all new recall requests to its local queue instead of the CRQ
- Moving all of its requests currently on the CRQ to its local queue

- Completing any remote requests that it had previously selected
- No longer selecting requests from the CRQ

You must wait for the disconnection to complete before attempting to connect to another CRQ. A message indicates when the disconnection is complete.

Note: Disconnecting from the CRQ should be done only on an exception basis.

Processing of common recall queue requests during shutdown

With the CRQ, individual DFSMSHsm hosts automatically disconnect from the structure during shutdown processing. Therefore, for normal processing, do not issue SETSYS COMMONQUEUE(RECALL(DISCONNECT)) before shutting down DFSMSHsm. Doing so moves all of the host's requests back to the local queue, which causes WAIT requests to be failed with message ARC1192I and to be purged from the CSA.

When a host is connected to the CRQ when shutdown processing is initiated, an outstanding WAIT recall request that is on the CRQ is failed, but a new NOWAIT request replaces this request, so that this request is preserved. The new request has the same priority as the original request, but is a NOWAIT request instead of a WAIT request. The new request is eligible for processing by any eligible host still connected to the CRQ. NOWAIT requests remain available in the CRQ.

Note: Batch WAIT requests are not converted. They remain in the CSA queue so that the request can time-out if it is not processed in a timely manner.

Deleting the common recall queue structure

If you need to delete the CRQ structure, perform the following steps:

1. Verify that the CRQ is empty by using the following command:

```
QUERY COMMONQUEUE(RECALL)
```

2. Disconnect all DFSMSHsm hosts from the CRQ structure by using the following command:

```
SETSYS COMMONQUEUE(RECALL(DISC))
```

3. Verify that all DFSMSHsm hosts have disconnected by using the following command:

```
D XCF,STR,STRNAME=SYSARC_basename_RCL
```

4. Delete the CRQ structure by using the following command:

```
SETXCF FORCE,STRUCTURE,STRNAME=SYSARC_basename_RCL
```

Processing failures in a common recall queue environment

DFSMSHsm processes the following types of failures or errors:

- Coupling facility or coupling facility structure failures
- Coupling facility link communication failures
- z/OS image failures
- DFSMSHsm primary address space failures
- Full list structure failures

Coupling facility or coupling facility structure failure

If the coupling facility fails in such a way that the CRQ list structure is lost, each DFSMSHsm host responds in the following ways:

1. Each host continues processing requests on its local queue and remote requests that are already selected.
2. Each host places onto its local queue the local copies of its outstanding requests and all new requests.
3. If a host starts a local request that is currently being recalled by a remote host, the user is notified that the request is currently being processed by another host.
4. Each host attempts to reconnect to the CRQ until the connection is successful. After a host successfully reconnects to the CRQ, it moves all of its unselected local requests from its local queue onto the CRQ. Requests that are currently running continue to completion.
5. The host resumes normal processing.

Coupling facility link failure

If one or more hosts lose their connection to the CRQ, the CRQ list structure remains intact, but those hosts can no longer participate in the CRQ activities. A host that loses its connection performs the same steps as with a coupling facility failure. Hosts that remain connected while another host has lost its connection respond in the following ways:

- Each host finishes the remote requests that originated on the host that lost connectivity.
- Each connected host continues to select and process the requests that the failed host placed onto the CRQ before losing connectivity.

z/OS image failure

If a z/OS image fails, the DFSMSHsm host on that system also fails, but all of the recall requests that originated on that host remain intact on the CRQ. The coupling facility notifies the remaining connected hosts of the failure. The requests that originated on the failed host remain eligible for selection. In-process requests on the failed host remain on the queue and are made available for other hosts to restart them.

If the failing host is processing a request from ML2 tape, then all requests from that tape cannot be selected until the failing host is restarted. The tape is marked as "in-use" by the failing host, and that indication will not be reset until the failing host restarts. If the failing host is not restarted for some time, enter the LIST HOST(*hostid*) RESET command, where *hostid* is the DFSMSHsm host id of the host that has failed.

DFSMSHsm primary address space failure

If a DFSMSHsm host fails, all of its recall requests on the CRQ remain intact and eligible for processing by other active hosts. The coupling facility notifies the remaining connected hosts of the failure. In-process requests on the failed host remain on the queue and are available for other hosts to restart them.

If the failing host is processing a request from ML2 tape, then all requests from that tape cannot be selected until the failing host is restarted. The tape is marked as "in-use" by the failing host, and that indication will not be reset until the failing host restarts.

Full list structure failure

If the list structure becomes 95% full, DFSMSHsm routes all new requests to the local queues. Routing recall requests to the CRQ resumes once the structure drops below 85% full. The list structure is not allowed to become 100% full so that requests that are in-process can be moved between lists within the structure without failure. When the structure approaches maximum capacity, you should increase the size by altering the structure to a larger size or by rebuilding it. A rebuild must be done if the maximum size has already been reached. (The maximum size limit specified in the CFRM policy must be increased before the structure is rebuilt.) DFSMSHsm relies on the XES structure full monitoring support to issue warning messages about the fullness of a structure.

Rule: If one of the above failures occurs, issue the AUDIT COMMONQUEUE command after each of the hosts has restarted or reconnected, or both.

Related reading

For more information about altering or rebuilding a list structure, see [z/OS MVS Programming: Sysplex Services Guide](#) and [z/OS MVS System Commands](#).

Processing priority of recalls and deletions

You may receive complaints from end users that an inordinate amount of time was required to service a long series of NOWAIT requests for recalls or deletions. The delay occurs only if DFSMSHsm is heavily loaded with recall and deletion requests in relation to the number of tasks that are permitted to process them. The delay is caused by the priority scheme that DFSMSHsm uses to prevent one user from monopolizing the system and from tape mount optimization processing.

The same queue of management work elements (MWEs) processes all recall and deletion requests that are submitted to one DFSMSHsm host. This queue is called the recall queue. DFSMSHsm gives all WAIT requests a higher priority (position) on the queue than NOWAIT requests. The NOWAIT requests are interleaved according to the user ID of the submitter. All users start one NOWAIT request before any user starts a second NOWAIT request. Processing of the requests is generally from the oldest to the newest requests.

An exception to the order of processing is when doing recalls from tape, where DFSMSHsm reduces the number of mounts by doing several recalls from a given tape while the tape is continuously mounted. If a recall task has just completed a recall from a tape, it processes the highest-priority request that is currently on the queue for that same tape, no matter where the request is positioned on the queue. DFSMSHsm continues to process only recalls from this single tape for a duration of time that is specified by the TAPERECALLLIMITS parameter. After the time specified by the TAPERECALLLIMITS parameter is reached, DFSMSHsm determines whether a higher-priority recall request that needs this tape exists on another DFSMSHsm host. If so, it ends the recall processing on the tape by the current DFSMSHsm host, freeing the tape for use by other hosts. If no higher-priority request exists, DFSMSHsm continues to process any recalls requested from the tape. DFSMSHsm checks after every recall for a higher-priority recall request from other hosts.

When performing recalls from tape, DFSMSHsm also determines if there is a higher-priority tape recall request from the same host, for a different tape, that cannot be performed because DFSMSHsm is already processing at the maximum permitted tape recall task limit. If so, it ends recall processing on the tape that has met the specified TAPERECALLLIMITS processing time, enabling the higher-priority recall request from DASD or a different tape to take place.

Programming Interface Information

You can use the return-priority installation exit ARCRPEXT to change the priorities of these requests as the requests are put on the queue. ARCRPEXT allows you to assign a relative queuing priority such that WAIT requests can be prioritized relative to other WAIT requests and NOWAIT requests prioritized relative to other NOWAIT requests. You can, for example, give recalls a higher priority than deletes, or give interactive requests a higher priority than requests from batch processing. You can even negate the interleaving scheme described earlier in this topic for interactive NOWAIT requests. (In the case of a NOWAIT request that is initiated from a volume recover request, the queuing priority is initialized with the value that is propagated from the volume recover request passed earlier to the exit.) For more details on ARCRPEXT, see [z/OS DFSMS Installation Exits](#).

Note: With regard to the common recall queue, ARCRPEXT is invoked by the host that the request originated on.

End Programming Interface Information

You can also affect the order in which data sets are recalled from a single tape by specifying the SETSYS command with TAPEDATASETORDER. For more information about the TAPEDATASETORDER keyword, see [“Specifying the order in which data sets from a single tape are processed”](#) on page 136.

Specifying the order in which data sets from a single tape are processed

You can use the SETSYS TAPEDATASETORDER command to specify the order in which you want DFSMSHsm to recall or recover data sets from a tape. The command provides the installation with additional flexibility to handle circumstances in which ascending FBID order offers significant reduction in tape repositioning.

To recall data sets in ascending FBID order, enter the following command:

```
SETSYS TAPEDATASETORDER(RECALL(FBID))
```

To recall data sets in PRIORITY order, enter the following command:

```
SETSYS TAPEDATASETORDER(RECALL(PRIORITY))
```

You can specify the order in which data sets are recovered from a tape, by using the following command:

```
SETSYS TAPEDATASETORDER(RECOVER(FBID | PRIORITY))
```

When deciding whether to specify PRIORITY or FBID, you should consider that a lower priority request might be processed prior to a higher priority request. If you do not specify the TAPEDATASETORDER option, DFSMSHsm uses PRIORITY as default. For more information about the TAPEDATASETORDER, see [“TAPEDATASETORDER: Specifying order for processing tape” on page 1188](#).

Migration volume contention priorities

Because DFSMSHsm migration volumes can also be involved in recall, recycle, tape copying, and aggregate backup activities, you must understand the usage priorities for your migration volumes. DASD ML1 volumes with small data set packing data sets allocated on them and tape ML2 volumes can interact with other functions that have a greater usage priority. When this interaction occurs, you may not get the results you are expecting.

Related reading

- For more information about the contention issues that can arise with DASD ML1 volumes with SDSP data sets allocated on them and tape ML2 volumes, see [“SDSP data set contention” on page 142](#) and [“Tape migration level 2 volume contention” on page 136](#).
- For more information about the contention details related to the common recall queue, see [“Common recall queue environment” on page 141](#).

Tape migration level 2 volume contention

A tape ML2 volume can be in use in only one of the following ways at any single point in time:

- As a target volume for migration data output. Primary space management, interval migration, secondary space management, and data set and volume command management all generate migration data output.
- As a target volume for recycle.
- As a source volume for recycle.
- As a source volume for recall.
- As a source volume for TAPECOPY.
- As a source volume for ABACKUP.
- As a target volume for ARECOVER.

This topic describes what DFSMSHsm does to resolve volume contention among the different tasks that request the same tape ML2 volume.

Single DFSMSHsm-host environment

This topic describes the processing that occurs when a tape migration volume is needed but is not available because another task is using the tape volume.

Migration or recycle output needs volume

Independent of what type of migration or recycle task needs an output tape, if the tape initially selected is in use by any task in DFSMSHsm, then the output task selects another target tape.

Recall needs volume

If recall needs a data set that is on a tape volume being used for ML2 output, then recall causes the output task to relinquish that tape and continue its output using a different tape. This applies to all forms of migration: primary space management, interval migration, secondary space management, command data set migration, and command volume migration. It also applies to recycle output.

The recall function also causes tapes to be taken away from other tasks, such as recycle input, TAPECOPY input, and from recall itself.

- For recycle input, the recycle function ends processing on the input connected set and moves on to the next best connected set.
- For TAPECOPY input, the tape takeaway function is optional. After the first recall attempt there is a time allowance before the function initiates the takeaway. This gives TAPECOPY a chance to complete. All the copying done up to the point of the tape takeaway is discarded if the TAPECOPY task does not complete.
- There is a form of recall versus recall contention, and takeaway within a host. If the maximum number of tape recall tasks are running and taking a long time, a high-priority recall for a different tape may not get started because there are no available tasks. This is possible when many requests exist for each tape and you are using tape mount optimization. For this condition DFSMSHsm provides a task takeaway consideration at a specified amount of time after a tape is first mounted for recalls. After the specified amount of time is met, a higher-priority request causes recall processing to end on the current tape, freeing a task for the highest-priority request.

Because a takeaway does not happen instantaneously, the recall task does not wait, but leaves this recall on the queue to be retried at approximately two-minute intervals. This enables recalls that are readily performed to proceed. When migration or recycle processing is complete for the data set being processed, the allocated volume is released and a new volume is selected for migration or recycle processing.

The recall function can also contend with ABACKUP and ARECOVER.

- If ABACKUP is reading an ML2 tape and recall needs that tape, there are two possibilities:
 - If the recall request is a NOWAIT request, recall repeatedly tries to access the tape until it reaches the specified limit for retries. A NOWAIT recall does not take tapes away from ABACKUP.
 - If the recall request is a WAIT request, recall repeatedly tries to access the tape. However, it does not flag ABACKUP to give up the tape until ten minutes after the first time recall finds the tape is in use. This ten minute delay gives ABACKUP time to complete its use of the tape. After ABACKUP is notified, recall repeatedly tries to access the tape until (unlikely in this case) it reaches the specified limit for retries.
- If ARECOVER is writing an ML2 tape and recall needs that tape, then recall keeps retrying to access that tape until it reaches the specified limit for retries. Recall does not take tapes away from ARECOVER.

When recall completes its use of a tape taken away from other tasks, the tape is released and is available for continued use by other tasks.

If a tape needed by recall does not become available to the recall task after 15 retries, DFSMSHsm sends message ARC0380A to the operator. The operator can give one of the following answers:

WAIT

This reply causes the recall task to reset its count of attempts, and begin another sequence of up to 15 retries at two-minute intervals.

CANCEL

This reply causes the recall task to end without even allocating the tape, which causes the recall to fail. If you know that the tape will not become available in a reasonable amount of time, use this choice.

MOUNT

This reply causes the recall to go ahead and request that the subject tape be allocated and mounted. Use this choice only when you are certain the tape can be mounted. An example is if DFSMSHsm says that the tape is in use by a host that is down and is not running.

Related reading

- For more information about the delay time associated with ABACKUP and how to change the delay time, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about changing the number of recall retries, see [z/OS DFSMSHsm Implementation and Customization Guide](#), under supported patches. If the number of retries is patched to zero, the recall immediately fails.

ABACKUP needs input volume

If ABACKUP needs the migration tape that is in use by recall, ABACKUP performs retries at 20-second intervals until the tape becomes available. It continues to retry for a period of 30 minutes, whichever occurs first. If the time expires, then the operator has the option to retry or to fail the ABACKUP. There is no takeaway function by ABACKUP from recall.

The ABACKUP function can cause takeaway from recycle input and from TAPECOPY input.

RECYCLE

The recycle function ends the input connected set and moves on to the next best connected set.

TAPECOPY

The TAPECOPY function is optional, and there is a time allowance after the ABACKUP function is first attempted before it initiates the takeaway. This occurs so that the TAPECOPY might complete, rather than discard, its work.

If ABACKUP needs the migration tape that is in use by migration or recycle output, ABACKUP initiates a tape takeaway. It then performs retries at 20 second intervals until the tape becomes available, or for a period of 30 minutes, whichever occurs first. If the time expires, the operator has the option to redo the set of retries or to fail the ABACKUP processing, as described in message ARC6254A.

If ABACKUP needs the migration tape that is in use by another ABACKUP, then it skips processing that tape and proceeds to process other migration and user tapes. It later reattempts to process the tape that was in use.

Recycle input volume in use

If migration or recall is using the volume that is needed as a source volume for recycle, and the recycle request is a specific RECYCLE VOLUME(volser) request, the recycle fails. If the recycle request is a generic RECYCLE request, then the recycle of that connected set is skipped and message ARC0845I is issued to indicate such. Recycle processing continues with the next eligible connected set. If someone issues a RECYCLE command for a specific volume and that volume is the target recycle volume, DFSMSHsm selects another target ML2 volume. DFSMSHsm then recycles the specified volume.

Table 13 on page 138 is a summary of possible volume contention in a single DFSMSHsm-host environment or within the same DFSMSHsm host in a multiple DFSMSHsm-host environment.

Table 13. Volume Contention in a Single DFSMSHsm-Host Environment

Function That Has Volume	Function That Needs Volume	Result
Migration	Migration	No contention.
Recall	Migration	Migration selects a new target volume.
Recycle (source)	Migration	Migration selects a new target volume.

Table 13. Volume Contention in a Single DFSMSHsm-Host Environment (continued)

Function That Has Volume	Function That Needs Volume	Result
Recycle (target)	Migration	No contention.
Migration, Recycle	Recall	Recall initiates a tape takeaway, and periodically retries the recall. When migration or recycle processing is complete for the data set being processed, the allocated volume is released and a new volume is selected for migration or recycle processing.
Recall	Recall	Terminate in favor of a higher priority recall from a different tape if running at maximum tape tasks, and it has been greater than specified amount of time since tape began to be continuously mounted.
ABACKUP	Recall	Recall periodically retries accessing the data set on the tape. If Recall is a WAIT request, after a delay, it initiates a tape takeaway, which causes ABACKUP to skip any remaining data sets that are needed by ABACKUP on the tape and process from other migration and user tapes. ABACKUP later tries to process those data sets that were skipped earlier.
ARECOVER	Recall	Recalls are retried until the tape is available or a retry limit is reached.
TAPECOPY (source)	Recall	If tapecopy takeaway is enabled, Recall retries for 15 minutes before requesting tape takeaway. If TAPECOPY has not completed its copy within one minute of the tape takeaway request, the copy fails.
Migration	Recycle (source)	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recall	Recycle (source)	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recycle (source)	Recycle (source)	No contention.
Recycle (target)	Recycle (source)	Recycle selects a new target volume.
Migration	Recycle (target)	No contention.
Recall	Recycle (target)	Recycle selects a new target volume.
Recycle (source)	Recycle (target)	Recycle selects a new target volume.
Recycle (target)	Recycle (target)	Recycle selects a new target volume.
Migration, Recycle	ABACKUP	ABACKUP initiates a tape takeaway and periodically retries the data set. When migration or recycle processing is complete for the data set being processed, the allocated volumes are released. A new target tape is selected for migration or recycle output contention, or a new connected set is selected if there is recycle source contention.
Recall	ABACKUP	ABACKUP is retried until the tape is available or a retry limit is reached.
ABACKUP	ABACKUP	ABACKUP skips over the tape in use and processes from other migration tapes. It reattempts to process the skipped tape later.
TAPECOPY (source)	All except Recall and ABACKUP	Function fails.
TAPECOPY (source)	ABACKUP	If tapecopy takeaway is enabled, ABACKUP retries for 15 minutes before initiating a tape takeaway. If TAPECOPY has not completed its copy within one minute of noting the takeaway request, the copy fails.

Multiple DFSMSHsm-host environment

This topic describes the processing that occurs when a tape ML2 volume is needed but is not available because either another task in this DFSMSHsm host, or a task in another host is using the tape volume. The contention processing is so similar for multiple tasks or hosts to the processing described in

the preceding topics for a single host environment, that only the differences are discussed here. The differences include the following situations:

Recall needs tape in use by recall on a different host

The following occurs when DFSMSHsm is doing recall mount optimization, which means that several recalls occur end to end while a tape is continuously mounted. Since this could go on for an indefinite length of time, DFSMSHsm has the SETSYS option TAPERECALLLIMITS. TAPERECALLLIMITS provides a time value after which a tape can be taken away from recall by recall on another DFSMSHsm host. The takeaway occurs only if the request on the other host is of higher priority than the request that would otherwise be performed without demounting the tape. The priority is typically based on time of arrival, but the priority can be changed by using the ARCRPEXT installation exit.

Note: A common recall queue environment needs only one tape mount. The host that mounts the tape processes all requests regardless of which hosts originate the requests.

Recycle needs tape in use by recycle on a different host

Since recycle can occur concurrently on different DFSMSHsm hosts, it is possible for a user on one host to request recycle of a tape that is currently in use by recycle on another host. In this case the request fails, regardless of whether the request was for use of the tape for recycle input or recycle output.

Table 14 on page 140 is a summary of possible volume contention in a multiple DFSMSHsm-host environment.

Table 14. Volume Contention in a Multiple DFSMSHsm-Host Environment

Function That Has Volume	Function in Other Host That Needs Volume	Result
Migration	Migration	No contention.
Recall	Migration	Migration selects a new target volume.
Recycle (source)	Migration	Migration selects a new target volume.
Recycle (target)	Migration	No contention.
Migration, Recycle	Recall	Recall initiates a tape takeaway, and periodically retries the recall. When migration or recycle processing is complete for the data set being processed, the allocated volume is released and a new volume is selected for migration or recycle processing.
Recall	Recall	A specified period of time after an ML2 tape is mounted, and when a recall completes, the tape is taken away if a request exists on another host that is of higher priority than the request that normally would have been performed on the recalling host from the already mounted tape.
ABACKUP	Recall	Recall periodically retries accessing the data set on the tape. If Recall is a WAIT request, after a delay, it initiates a tape takeaway, which causes ABACKUP to skip any remaining data sets that are needed by ABACKUP on the tape and process from other migration and user tapes. ABACKUP later tries to process those data sets that were skipped earlier.
ARECOVER	Recall	Recalls are retried until the tape is available or a retry limit is reached.
TAPECOPY (source)	Recall	If tapecopy takeaway is enabled, Recall retries for 15 minutes before requesting tape takeaway. If TAPECOPY has not completed its copy within one minute of noting the tape takeaway request, the copy fails.
Migration	Recycle (source)	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recall	Recycle (source)	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recycle (source)	Recycle (source)	Recycle skips the volume because another host is recycling it.

Table 14. Volume Contention in a Multiple DFSMSHsm-Host Environment (continued)

Function That Has Volume	Function in Other Host That Needs Volume	Result
Recycle (target)	Recycle (source)	Recycle fails.
Migration	Recycle (target)	No contention.
Recall	Recycle (target)	Recycle selects a new target volume.
Recycle (source)	Recycle (target)	Recycle selects a new target volume.
Recycle (target)	Recycle (target)	No contention.
Migration, Recycle	ABACKUP	ABACKUP initiates a tape takeaway and periodically retries the data set. When migration or recycle processing is complete for the data set being processed, the allocated volumes are released. A new volume is selected for migration or recycle output contention, or a new connected set is selected if there is recycle source contention.
Recall	ABACKUP	ABACKUP is retried until the tape is available or a retry limit is reached.
ABACKUP	ABACKUP	ABACKUP skips over the tape in use and processes from other migration tapes. It reattempts to process the skipped tape later.
TAPECOPY (source)	All except Recall and ABACKUP	Function fails.
TAPECOPY (source)	ABACKUP	If tapecopy takeaway is enabled, ABACKUP retries for 15 minutes before initiating a tape takeaway. If TAPECOPY has not completed its copy within one minute of noting the takeaway request, the copy fails.

Common recall queue environment

With all hosts in an HSMplex connected to the same common recall queue (CRQ), no contention for the same tape exists within the recall function. This makes the recall tape takeaway from recall function obsolete. With multiple CRQplexes in the same HSMplex, recall tape takeaway from recall manages contention between the separate CRQplexes. With one or more hosts not connected to a CRQ, recall tape takeaway from recall manages contention between the unconnected hosts and the CRQ.

For hosts connected to a CRQ, recall task takeaway becomes involved only when recalls are directed to a host's local queue. It then becomes possible for that host's tape tasks to process multiple requests from single tapes while a higher priority tape request is on its local queue.

Using the recall takeaway function, any host that is connected to the CRQ can process the recalls from a tape that was marked for takeaway by another host that was connected to the same CRQ.

Related reading

For more information about CRQplex configuration, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Specifying the desired number of ML2 partial tapes

ML2 tapes are written by the various migration functions and by the recycle of ML2 tapes. A relatively small number of ML2 partially filled tapes are needed as substitutes for tapes that are receiving output when they are taken away by a recall task. The partially filled ML2 tapes that are used as substitutes are not currently associated with specific ML2 output tasks for any hosts.

These substitute tapes are ML2 "floaters." To have the recycle process limit the number of floaters to *nnn*, specify:

```
SETSYS ML2PARTIALSN0TASSOCIATEDGOAL(nnn)
```

To allow an unlimited number, specify:


```
SETSYS ML2PARTIALSNOTASSOCIATEDGOAL(NOLIMIT)
```

Floater tapes are writeable ML2 tapes that are not assigned or associated with a specific task in a specific host. When a floater substitutes for a tape that is taken away, it becomes an associated tape, restricted for output to that one specific task.

An output tape taken away by recall is not immediately available to receive output again. It becomes available for output on the host where the recall occurred as the recall completes. A long queue of recalls from that tape would likely cause a delay in considering such a tape for output again.

DFSMSHsm hosts other than the one doing the recall do not become aware of a tape's availability for output until their own in-storage selection table is depleted and rebuilt. For performance reasons, in-storage selection tables are not rebuilt more often than every 45 minutes.

A configuration where a storage group uses multiple SMS tape libraries can affect the number of floaters that are needed. As functions such as primary space management, interval migration, or recycle start, they can allocate a drive from any of the libraries associated with the storage group. A tape must be in the same library as the drive that is allocated. This tends to increase the number of floaters needed, even without a high tape take-away rate. The tape associated with the previous processing of a task may not be part of the library that contains the allocated drive. In this case, a floater is used, and the previously associated tape is placed in its host's TVT.

From the preceding descriptions, you can see how data usage and configuration can result in a need for a higher number of ML2 floaters. Having floaters is a trade-off between having a few additional under-utilized ML2 tapes, or recycling the additional few under-utilized ML2 tapes. The DFSMSHsm design is to extend a partially filled ML2 tape instead of using an empty tape. However, in cases where partially filled ML2 tapes do not exist, DFSMSHsm requests a scratch tape.

It is *only* when recycle of ML2s occurs that the value specified by ML2PARTIALSNOTASSOCIATEDGOAL is used. The default used for ML2PARTIALSNOTASSOCIATEDGOAL is 10. This value applies to the total number of ML2 not-associated partial tapes in an entire HSMplex, but only for tapes that meet the recycle selection criteria of PERCENTVALID and any SELECT filters. If the number of such floaters, at the time a recycle command runs, exceeds the specified value, then the excess tapes are included as candidates for recycle. The floaters that are thus recycled are those having the lowest percentage of valid data.

You can use the LIST TTOC SELECT(NOTASSOCIATED) command to get a list of all not-associated ML2s if you want to know how many or which ones they are. If you observe a large number of not-associated ML2 partials being recycled, and this seems to occur consistently, then you probably want to increase the SETSYS specification to reduce the recycle load. This is just a trade-off between having the few under-utilized tapes or recycling the tapes so you do not have them (at least for a while).

You can use the LIST TTOC SELECT(ASSOCIATED) command to get a list of all ML2 partials that are associated with specific ML2 output tasks. If some of these are associated with tasks that are unused, or seldom used, then you may want to specify DELVOL UNASSIGN so that they will be part of the set of floaters. To find out which ML2 tapes are associated with what tasks, display the various host MCR records using the FIXCDS command. If they are old-technology cartridges, or recorded in old technology not being written anymore, you should use the DELVOL MARKFULL command so they can be recycled when they meet that criteria.

One approach for determining a unique setting for your installation is to mark the existing ML2 partial tapes as full. Then set ML2PARTIALSNOTASSOCIATEDGOAL to NOLIMIT for a while, and issue the LIST TTOC command with SELECT(ASSOCIATED) and SELECT(NOTASSOCIATED) every few days. When the sum of these two categories settles down or reaches a maximum, your migration workload indicates that you can set ML2PARTIALSNOTASSOCIATEDGOAL to or slightly below the number of not-associated partials indicated by the LIST TTOC command.

SDSP data set contention

Because SDSP data sets reside on ML1 volumes, they can be involved in various ways with DFSMSHsm processing of level 0 volumes and ML2 volumes. For example, an SDSP data set can be a target for automatic primary space management and, at the same time, the SDSP can be a source for a user's recall

of a small data set from an SDSP data set. SDSP data sets can be included in aggregate backup and SDSP data sets require cleanup after a user recalls a migrated data set. Additionally, the AUDIT and FREEVOL functions can be run against volumes on which SDSP data sets reside.

Because SDSP data sets are potentially involved in various DFSMSHsm processes, there are times when concurrent DFSMSHsm processing activities cause contention for SDSP data sets. [Figure 20 on page 144](#) shows that an SDSP data set can be effected by various DFSMSHsm processes. [Table 15 on page 145](#) shows the usage priorities for SDSP data sets and describes how DFSMSHsm resolves contention for SDSP data sets.

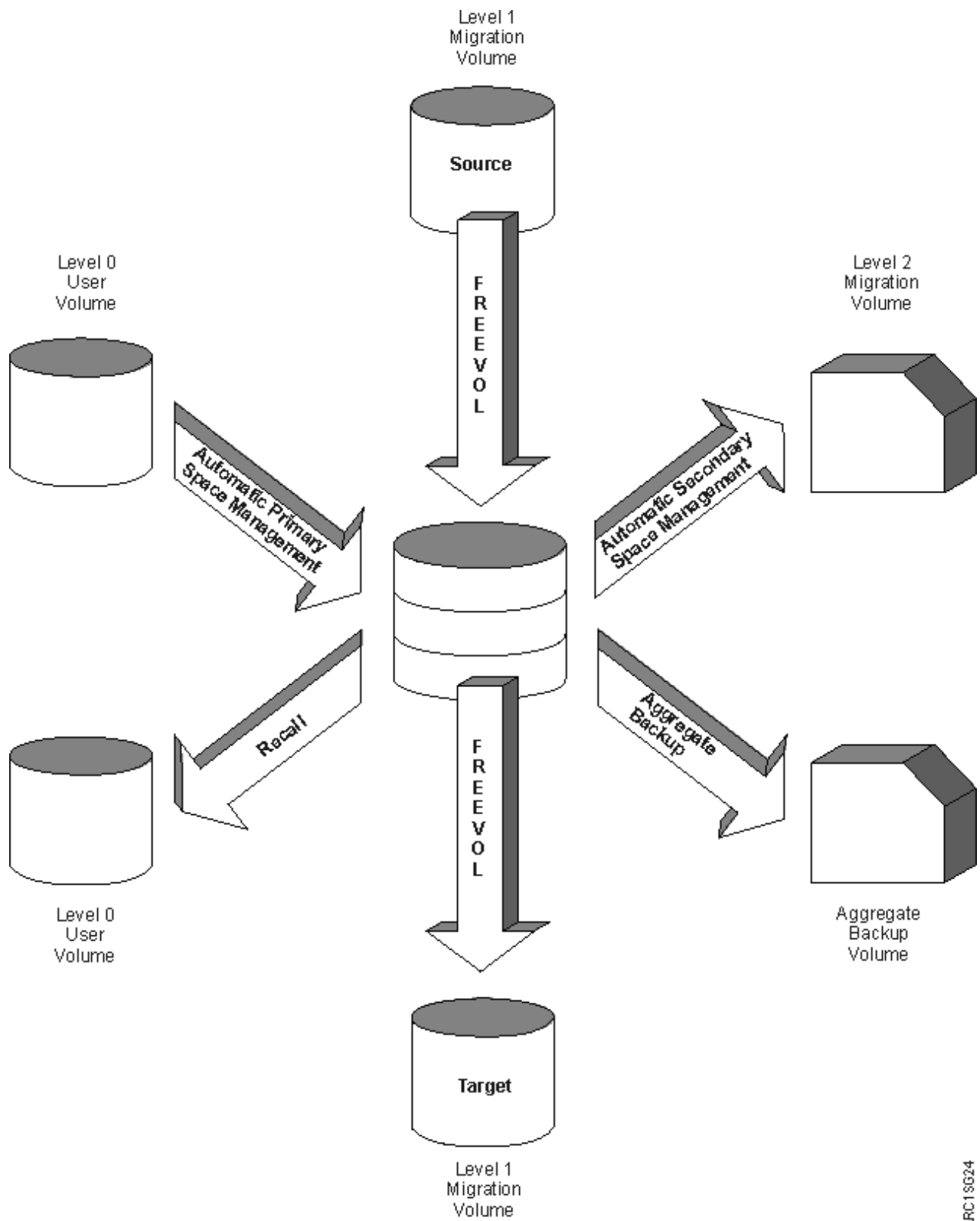


Figure 20. SDSP Contention Environment

ARC1SG24

Table 15. SDSP Contention in Single and Multiple DFSMSHsm-Host Environments

Function That Has SDSP	Function That Needs SDSP	Result
Migration (L1 to L2) Recall FREEVOL (source) AUDIT MEDIACONTROLS (source) Aggregate backup	Migration (L0 to L1)	<p>If all other SDSP data sets are presently in use, this small user data set is skipped and, most likely, can migrate to an SDSP during your next migration window.</p>
Migration cleanup or migration (L1 to L2)	Migration (L0 to L1)	<p>Migration cleanup and level 1 to level 2 migration are functions of automatic secondary space management. Because SDSP data sets can be open for a long time during migration cleanup or during level 1 to level 2 migration, it is likely that they will be in use. If all SDSP data sets are presently in use, this small user data set is skipped and, most likely, can migrate during your next migration window.</p> <p>Note: Do not run automatic primary space management and automatic secondary space management concurrently.</p>
Migration (L0 to L1)	Recall	Migration processing gives up its use of this SDSP data set to the recall function that requests the SDSP data set. If all SDSP data sets are presently in use, the migration of this small user data set is skipped and, most likely, can migrate during your next migration window.
Recall	Recall	The first recall request to access the SDSP data set completes its processing. Other recall requests retry the SDSP data set request and, most likely, will be successful.
FREEVOL (source) AUDIT MEDIACONTROLS	Recall	Because FREEVOL and AUDIT MEDIACONTROLS are relatively long-running processes, the recall request will either retry until successful or retry until timing out and failing.
Migration cleanup	Recall	Migration cleanup gives up its use of this SDSP data set to the recall function that requests the SDSP data set. Migration cleanup skips its processing of the SDSP data set.
Aggregate backup	Recall	Recall processing retries the recall request and, most likely, will be successful after the aggregate backup has finished.
Migration (L0 to L1) Recall AUDIT MEDIACONTROLS Migration cleanup Aggregate backup	FREEVOL (source)	FREEVOL processing fails.

Table 15. SDSP Contention in Single and Multiple DFSMSHsm-Host Environments (continued)

Function That Has SDSP	Function That Needs SDSP	Result
Migration (L0 to L1) Recall AUDIT MEDIACONTROLS Migration cleanup Aggregate backup	FREEVOL (target)	If the target SDSP data set cannot be made available to FREEVOL processing, this small data set is skipped and cannot be moved from the source SDSP data set.
Migration (L0 to L1) Recall FREEVOL AUDIT MEDIACONTROLS Migration cleanup Aggregate backup	AUDIT MEDIACONTROLS	If this SDSP data set is in use by other functions, the AUDIT request for the SDSP data set fails.
Migration (L0 to L1) Recall FREEVOL AUDIT MEDIACONTROLS Aggregate backup	Migration cleanup	If the SDSP that is needed by migration cleanup is in use, this small user data set is skipped and it is deleted the next time automatic secondary space management is run.
Migration (L0 to L1)	Aggregate backup	Migration processing gives up its use of the SDSP data set. If all SDSP data sets are presently in use, this small user data set is skipped and, most likely, can migrate during your next migration window.
Recall	Aggregate backup	Aggregate backup will retry its request for the SDSP and will eventually be granted access.
FREEVOL (source)	Aggregate backup	Aggregate backup fails.
FREEVOL (target)	Aggregate backup	FREEVOL processing gives up its use of this SDSP data set to the aggregate backup request. If no other SDSP data sets are available, FREEVOL processing skips the data sets.
AUDIT MEDIACONTROLS	Aggregate backup	Aggregate backup fails.
Migration cleanup	Aggregate backup	The SDSP data sets are released from migration cleanup. Migration cleanup skips processing from this SDSP data set.
Aggregate backup	Aggregate backup	The first aggregate backup request to reach the SDSP gets access to the SDSP. Other aggregate backup requests will retry until successful.

The functions of aggregate backup, aggregate recover, FREEVOL, and AUDIT should be run at times when they will not be contending for the use of the same SDSPs.

Multitasking considerations for SDSP data sets

It is important to plan the number of SDSP data sets in relation to the number of concurrent migration tasks and the amount of processing done by functions with a higher usage priority for the SDSP data sets.

It seems obvious that one SDSP data set is required for each concurrent migration task. However, there are some DFSMSHsm activities that have a higher usage priority for SDSP data sets. These processing activities are:

- Recall
- Aggregate backup
- FREEVOL
- AUDIT MEDIACONTROLS
- Automatic secondary space management

Any of these activities can gain control of your SDSP data sets and leave you with fewer than the expected number of SDSP data sets for migration.

When an activity with a higher usage priority for SDSP data sets has or requests an SDSP data set, that SDSP data set is no longer a candidate for migration. The small data set that is in need of migration must find a different, available SDSP data set or it is skipped and left unprocessed until your next migration window.

Additionally, if all SDSP data sets should become full (as a result of migrations to them) the filled SDSP data sets are not candidates for further migration. Full SDSP data sets are not seen by migration processing and, as a result, any small user data sets are migrated as large data sets to ML1 volumes.

Example: The following three-part example (see Figures Figure 21 on page 147, Figure 22 on page 148, and Figure 23 on page 148) shows how SDSP data sets can become unavailable for use as level 0 to level 1 migration targets. In Part 1, three concurrent migration tasks are moving three small user data sets from level 0 user volumes to ML1 volumes (with SDSP data sets defined on the ML1 volumes).

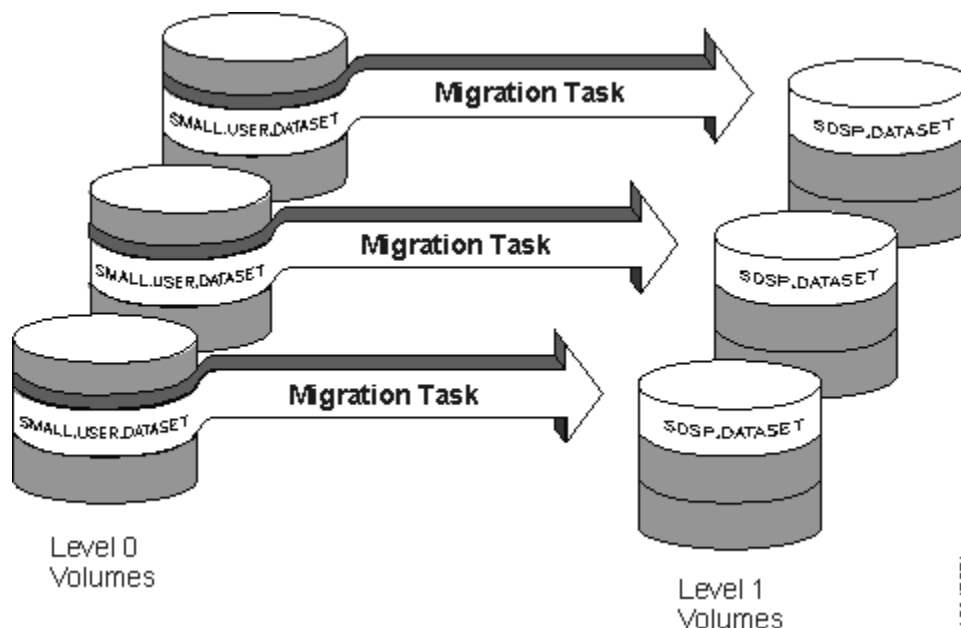


Figure 21. Part 1—Ideal Multitasking Migration

In Part 2, a recall of a small user data set from an SDSP data set during level 0 to level 1 migration has effectively eliminated one concurrent migration task. The small-user data set, whose migration was preempted by a recall, is skipped and will not be migrated until the next migration window.

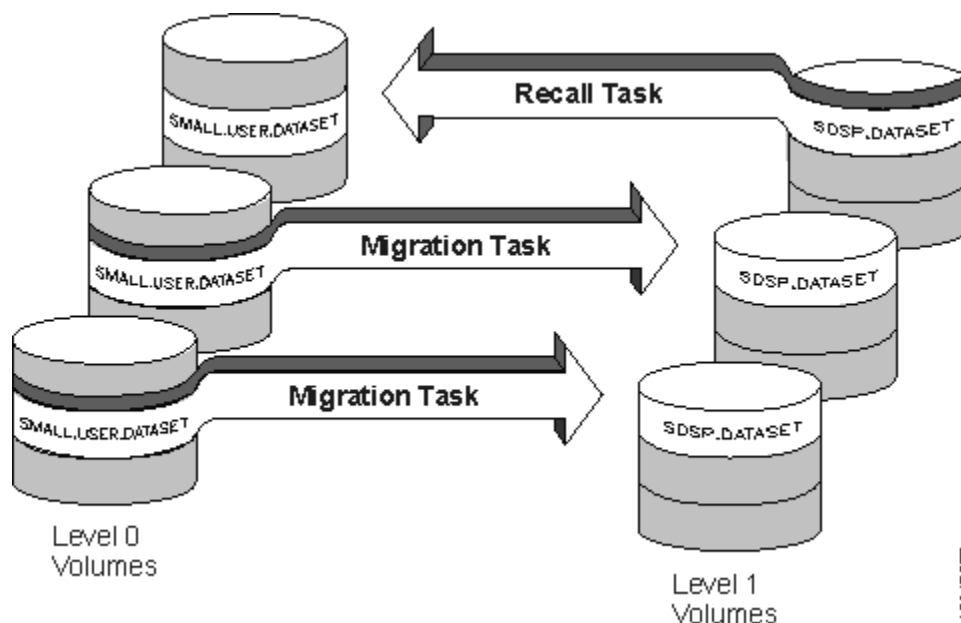


Figure 22. Part 2—Recall Processing Has a Higher Priority than Migration

In Part 3, all SDSP data sets have become full. They are no longer seen as candidates for level 0 to level 1 migration destinations and the small-user data sets migrate as large data sets.

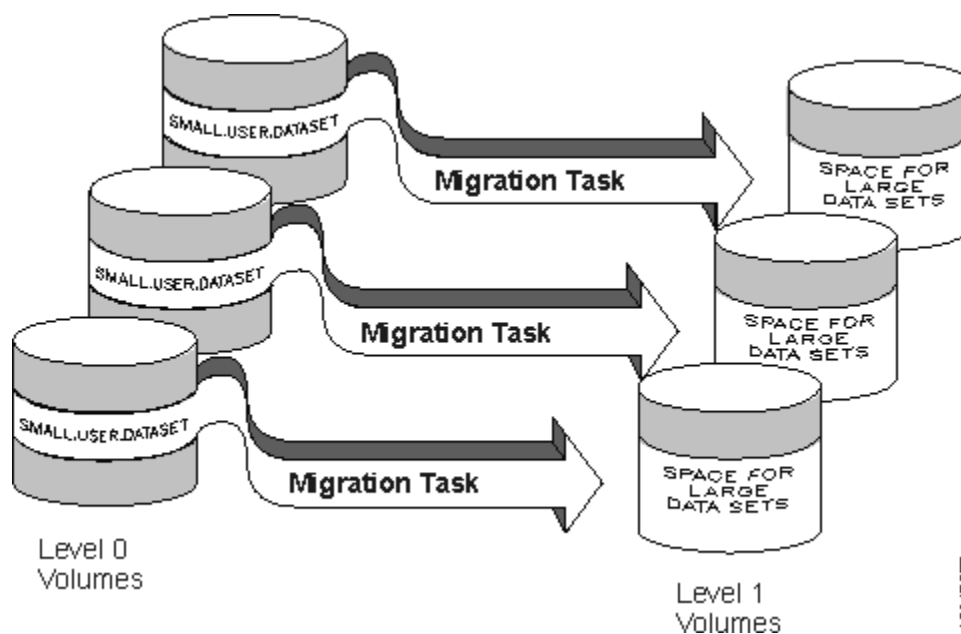


Figure 23. Part 3—All SDSP Data Sets Are Full

In addition to planning the timing of activities whose SDSP usage priority is high, you must ensure that you do not allow automatic secondary space management to run at the same time that automatic primary space management is running. If SDSP data sets are opened and allocated for automatic secondary space management, they cannot concurrently be targets for primary space management processing.

The point here is that other activity can affect your migrations and you need to plan and monitor those activities that can cause your small user data set migrations to be skipped. Therefore, you should define ample SDSP data sets to handle your worst case scenario (remember that one ML1 volume can have at most a single SDSP data set) and run automatic secondary space management at a different time than you run automatic primary space management.

Related reading

For more information about the SDSP parameter of the SETSYS command, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSShsm control parameters,” on page 1057.](#)

Preventing the processing of data sets during space management

Any of the following parameters or values specified with the following commands can prevent an otherwise eligible volume or data set from being processed by space management:

- The DEBUG parameter of the SETSYS command.
- The NOMIGRATION parameter of the SETMIG command.
- The EXITON and EXITOFF parameters of the SETSYS command, which specify the data set migration installation exit or the data set deletion installation exit. DFSMSShsm supplies sample exits that you can tailor to fit your needs.

Any of the following parameters or values specified with the following parameters can cause space management to stop processing or to fail to start processing:

- The MIGRATION(AUTO) parameter of the HOLD command. This stops or prevents automatic volume space management and automatic secondary space management from starting.
- The MIGRATION parameter of the HOLD command. This stops or prevents automatic and command volume space management, automatic secondary space management, and command data set migration from starting.
- The PRIMARYSPMGMTSTART parameter of the SETSYS command when it does not have a starting time specified. The ending time can also prevent space management from processing additional volumes.
- The SECONDARYSPMGMTSTART parameter of the SETSYS command when it does not have a starting time specified. The ending time can also prevent space management from processing additional data sets.
- The EMERGENCY parameter of the SETSYS command. This specifies that an emergency mode is in effect, and no DFSMSShsm processing related to moving user data is performed.

Note: This parameter does not affect the placement of recall requests onto the common recall queue. This enables this host to continue to place recalls onto the common recall queue that are eligible to be processed by other active hosts.

- The STOP command. This causes an orderly shutdown of DFSMSShsm.
- The REQUEST parameter of the SETSYS command. The operator can deny permission to start or restart automatic space management.

Deferred generation data sets

Deferred generation data sets are generation data sets (GDSs) for which a failure occurred before the data set was properly associated with the generation data group (GDG). These data sets are temporary in the sense that at some time they may become associated with the GDG. DFSMSShsm treats deferred generation data sets as permanent non-GDS data sets.

Security considerations for automatic space management

DFSMSShsm bypasses data set security checking during automatic space management, except after automatic recall. After automatic recall, security checking is performed by OPEN. This is to ensure that the user opening the recalled data set is authorized to use the data set.

DFSMSShsm also bypasses security checking when it processes operator commands entered at the system console or commands issued by a DFSMSShsm-authorized user.

DFSMSShsm checks authority for a requester to access data sets when a requester who is not DFSMSShsm-authorized issues an HMIGRATE, HDELETE, or HRECALL command. Security checking is not done when DFSMSShsm-authorized users issue the user commands. If users are not authorized to manipulate data,

DFSMSHsm does not permit them to migrate data sets, delete migrated data sets, or explicitly recall data sets. DFSMSHsm always does security checking for the scratch intercept when deleting migrated data sets. [Table 16 on page 150](#) shows the RACF authority required to perform each space management function.

Table 16. RACF Authority Required for Space Management Functions

DFSMSHsm Function	RACF Resource Access Authority Required
Migrate a data set	Update
Recall a data set using HRECALL (TSO users only)	Execute
Delete a migrated data set	Alter

Space management of data sets with discrete data set profiles

Discrete data set profiles provide a one-to-one relationship between a specific data set on a specific volume and its security authorization requirement. When you use only discrete data set profiles, you should use the RACFIND parameter of the SETSYS command. For a description of discrete data set profiles, see [“Discrete data set profiles” on page 618](#).

Note: Supported multivolume non-VSAM data sets cannot be protected by discrete profiles. Use generic profile protection for these data sets.

Migrating RACF-protected data sets

When an existing data set migrates to a DASD migration volume, the RACF indicator in the data set VTOC entry of the migration copy is set if the RACFIND parameter of the SETSYS command is in effect.

Note:

1. To maintain data set integrity on your system's small data set packing data sets, protect them with a discrete or generic profile.
2. When you migrate a data set, the *volser* in the discrete profile is not updated. Attempts to access the data set without recalling it may cause the ICH408I INSUFFICIENT ACCESS AUTHORITY message to be issued, depending on access intent and access allowed. The message will contain the user, group, data set name, volume(MIGRAT), access intent, and access allowed.

This message is issued because the catalog lookup before the RACHECK returns a volume serial number of MIGRAT. Therefore, MIGRAT is specified as the volume in the RACHECK. If RACF cannot find a discrete profile for the data set on *volser* MIGRAT, then a check is made for a generic profile. If a generic profile does not exist or does not give the requested access, the error message is issued.

This error message will not be issued if specific generic profiles are used as opposed to discrete profiles.

Deleting RACF-indicated migrated data sets

When DFSMSHsm deletes a migrated data set, it deletes the RACF discrete data set profile for the following components:

- Non-VSAM data sets.
- VSAM clusters eligible for volume migration. These clusters can contain up to eight specific components:
 - Base cluster
 - Base data object
 - Base index object
 - One base path
 - One alternate index (AIX)

- AIX data object
- AIX index object
- One AIX path
- VSAM clusters with RACF discrete profile protection only on the base cluster.

If you want to delete migrated VSAM clusters that are not eligible for level migration and whose RACF protection is not on the base cluster, perform the following actions:

1. Recall the data set.
2. Use an Access Method Services command to delete the cluster and any associated RACF profiles.

Recalling RACF-indicated data sets

If the RACF indicator was on when DFSMSHsm migrated the data set, the RACF indicator is set on when DFSMSHsm recalls the data set. When DFSMSHsm recalls a non-VSAM data set, DFSMSHsm changes the volume serial number and device type information, if necessary, in the RACF discrete data set profile to indicate the volume where DFSMSHsm has recalled the data set. When DFSMSHsm recalls a VSAM data set, it does not update the RACF discrete data set profile, because the profile contains the volume serial number of the catalog, which cannot change during recall or recovery.

To recall a RACF-indicated data set that has no RACF profile, perform the following actions:

1. Create a generic profile or recreate the RACF discrete profile. To create the discrete profile, use the RACF ADDSD command and specify the NOSET parameter.
2. Recall the data set without specifying where the data set should reside.

Password protection of data sets on DASD

First, you must remember that password protection applies only to non-SMS-managed data sets.

Password protection in MVS is implemented differently for non-VSAM and VSAM data sets. The passwords for all non-VSAM data sets are stored in one system password data set. The password for each VSAM data set is stored in a catalog record for the specific VSAM data set. Therefore, VSAM passwords are maintained during DFSMSHsm processing as a part of regular catalog record creation and updating.

If you are recalling a non-VSAM data set, the required password is the read-level password currently in the system password data set. If you are recalling a VSAM data set, the password is the one that existed when DFSMSHsm migrated the data set. If you are deleting a migrated non-VSAM data set, the current password is the write-level password.

Be aware of the following conditions when you attempt to change password protection while data sets are migrated or backed up:

- You cannot completely remove password protection from a migrated non-VSAM data set because the password-protection indicator is stored in control information in the migrated data. Also, the data set name of the migrated data set is not the user data set name. Therefore, DFSMSHsm cannot remove a password for a migrated non-VSAM data set from the system password data set because no password exists. Even if the password were deleted while the data set was migrated, the VTOC entry after recall would indicate that your data set is password-protected. You would need to add a password to the system password data set before you access the recalled, non-VSAM data set.
- You can change the password of a migrated non-VSAM data set by using the IEHPROGM utility. However, to change the password of a migrated VSAM data set, you must use Access Method Services. If you use Access Method Services, a migrated VSAM data set is recalled.
- If you delete a password-protected VSAM data set while it is migrated, all password protection is deleted. If you delete a password-protected non-VSAM data set while it is migrated, DFSMSHsm scratches the data set and deletes the catalog entry. However, the password still exists in the system password data set, and you should delete it.

- If you try to add password protection to a migrated non-VSAM data set, the password is added to the system password data set. However, the migrated data set does not have protection, and when DFSMSHsm recalls the data set, the VTOC entry does not indicate password protection.
- If you try to add password protection to a migrated VSAM data set, you must use Access Method Services. For a migrated VSAM data set, DFSMSHsm recalls the data set.

Note: If DFSMSHsm tries to create a catalog entry in a catalog having update protection, the operator receives a password prompt for an unauthorized DFSMSHsm function from an unauthorized DFSMSHsm user.

Access method services considerations

Access Method Services commands allow you to specify either FILE or INFILE to refer to DD statements that specify a data set name, a volume serial, or both, to be used for processing the commands. If a DD statement referred to by either FILE or INFILE specifies both a data set name and a non-SMS-managed volume, and if the data set has been migrated by DFSMSHsm, the Access Method Services command might not be performed successfully. If an error occurs for this condition, perform one of the following two actions before you reissue the Access Method Services command:

- If you need not specify the volume serial number on the DD statement, specify the DD statement without specifying the volume serial number and unit name. This specification causes the data set to be allocated when the job step is initiated, and the migrated data set is recalled for the allocation request.
- If you *must* specify the volume serial number on the DD statement [such as when you specify AMP = (AMORG) for VSAM data sets], recall the data set either:
 - Explicitly by issuing a DFSMSHsm RECALL or HRECALL command
 - Implicitly by referring to the data set without specifying the volume serial number in a step that precedes the step where the Access Method Services command is being requested

GUARANTEED SPACE attribute considerations

The storage class attribute of GUARANTEED SPACE allows the volumes on which a data set resides to be chosen explicitly when that data set is created. If GUARANTEED SPACE is used and any of the volumes specified are not part of one of the storage groups to which the storage class maps, the creation of the data set fails. The use of GUARANTEED SPACE requires some special considerations if DFSMSHsm is to manage data sets that are assigned to such a storage class.

When DFSMSHsm is the data mover, it uses the Access Method Services functions of EXPORT and IMPORT to migrate and recall VSAM data sets. The IMPORT function redefines a VSAM data set by using a copy of the catalog information saved in the EXPORT copy of the data set. This define request explicitly specifies the volumes from which the data set was exported. The explicit specification of the volume affects the way data sets are imported with the GUARANTEED SPACE attribute. Likewise, for non-VSAM data sets, VTOC data set services uses the volume serial number passed by DFSMSHsm if the storage class is GUARANTEED SPACE.

Results of recalling GUARANTEED SPACE VSAM data sets

When DFSMSHsm is the data mover, data sets recalled with the GUARANTEED SPACE attribute are normally placed on the volume from which they migrated. Consider the effects on a migrated, multiple-volume VSAM data set. A subsequent recall preserves the volume-to-component relationship that originally existed as long as the data set remains SMS-managed. For a data set in a GUARANTEED SPACE storage class, the relationship includes returning the data set to the same volumes from which it migrated. Thus, if a data set was originally defined with the data component on one volume and the index component on another, a migrate/recall sequence preserves the relationship between the components and the volumes. Similarly, if a key range data set has three key ranges on three separate volumes, the relationship is preserved.

If a volume is specified on the RECALL command, that volume is passed to IMPORT. When the IMPORT function redefines the VSAM data set, the specified volume overrides the volume or volumes from which

the data set has been exported. If the specified volume is eligible, the data set is imported to that single volume. If there is not enough space, the recall fails. If the volume is not eligible, another volume may be selected. If the data set resides on multiple volumes when it is migrated, it is recalled to a single volume.

Possible problems

In an environment in which the storage groups and storage classes do not change, the preceding results are acceptable for DFSMSHsm to manage the data sets. However, you can expect problems in an environment that is subject to storage group reconfiguration or to changes in the mapping of data sets to storage classes. These problems may discourage the use of DFSMSHsm space management for these data sets and may require special contingency procedures for availability management. The problems that occur relate to the following conditions:

- Changes in the storage group
- Changes in the storage class to data set mapping
- Conversions of data sets to GUARANTEED SPACE

Note: Use of extended address volumes in mixed environment where the volume cannot be brought online to all systems can also result in problems recalling and recovering data sets that belong to a Storage Class having the Guaranteed Space attribute.

Changes in the storage group

Consider the consequence of making a specific volume request in a storage group configuration that has changed. Assume that a data set with a GUARANTEED SPACE request has migrated. Some time later, one or more of the volumes on which the data set resided are removed from the storage groups that contain GUARANTEED SPACE data sets. A subsequent recall of the data set fails because at least one of the requested volumes is no longer part of a storage group for a GUARANTEED SPACE request. The result is the same for both single-volume data sets and multiple-volume VSAM data sets if the specific volume required is unavailable in the necessary storage group. Use of EAVs in mixed environments where the volume cannot be brought online to all systems can also result in problems recalling/recovering data sets that belong to a Storage Class having the Guaranteed Space attribute.

If a storage group is changed to remove volumes that data sets migrated from, you must perform one of the following actions:

- Change the ACS storage class routine so that a storage class with GUARANTEED SPACE is not assigned to the data set.
- Change the volume configuration again so that the storage group contains the volumes from which the data set migrated.
- Change the set of storage groups to which the GUARANTEED SPACE storage class maps to include one that already has the necessary volumes defined. This is applicable only if the volumes are already SMS-managed.
- Use the RECALL command with the FORCENONSMS and the VOLUME parameters. This RECALL command causes the data set to be recalled to a non-SMS-managed volume as a non-SMS-managed data set. You can then move, copy, or convert the data set as necessary.

Note: If the data set was multiple-volume when it migrated, it is recalled to a single volume.

Changes in the storage-class-to-data set mapping

Consider the consequence of making a specific volume request in a storage class mapping that has changed. Assume that a data set is defined and the ACS routines have assigned it to a storage class that does not have the GUARANTEED SPACE attribute or does not assign it to any storage class. The ACS routine is then changed to assign the data set to a storage class with the GUARANTEED SPACE attribute. Also, the original storage class is changed to include the GUARANTEED SPACE attribute. Therefore, the recall might fail because the requested volumes are not in the storage groups that receive GUARANTEED SPACE requests.

If an SMS-managed data set's original storage class is changed to specify the GUARANTEED SPACE attribute before the data set is recalled, you must perform one of the following actions:

- Change the ACS storage class routine so that a storage class without GUARANTEED SPACE is assigned to the data set.
- Change the storage class of the data set to a storage class without GUARANTEED SPACE, without recalling the data set, by using the IDCAMS ALTER command.
- Change the volume configuration in the appropriate storage groups to contain the volumes from which the data set migrated. The volumes may need to be converted to SMS management first. It may be difficult to determine all of the volumes involved. DFSMSHsm does not save all of the original volume serial numbers, only that of the volume on which the base data component resided.
- Change the set of storage groups to which the GUARANTEED SPACE storage class maps to include one that already has the necessary volumes defined. This is applicable only if the volumes are already SMS-managed.
- Use the RECALL command with the FORCENONSMS and the VOLUME parameters. This RECALL command causes the data set to be recalled to a non-SMS-managed volume as a non-SMS-managed data set. You can then move, copy, or convert the data set as necessary.

Note: If the data set was multiple-volume when it migrated, it is recalled to a single volume.

Conversions of data sets to GUARANTEED SPACE

If the data set was non-SMS-managed when it migrated and is assigned a storage class with GUARANTEED SPACE during recall, it is assigned to a volume in a storage group that contains GUARANTEED SPACE data sets.

If the data set was SMS-managed when it migrated but did not have GUARANTEED SPACE, the volumes from which it migrated might not be in a storage group that receives GUARANTEED SPACE requests. To avoid errors, you must perform one of the following actions before the recall:

- Change the target storage class with GUARANTEED SPACE to map to a storage group that contains the original volumes.
- Add the original volumes to a storage group mapped to by the target storage class with GUARANTEED SPACE.

DFSMSHsm retrieval of management classes and storage groups

If you are changing the definitions of management classes, you should know when the new definitions will take effect. DFSMSHsm retrieves a list of all currently defined management classes before the following functions begin:

- Automatic volume space management, which includes primary space management and interval migration
- Automatic secondary space management
- Command volume space management

If one of the volume functions is in progress or has just completed, and another volume function is about to begin, DFSMSHsm uses the list of management classes that exists from the prior processing. When secondary space management begins, a list of management classes is retrieved from the active SMS configuration and is not shared with the other functions in the preceding list.

When DFSMSHsm begins automatic volume space management, it retrieves the storage group definitions to determine which SMS-managed volumes are eligible for space management.

A storage group must meet the following criteria for DFSMSHsm to consider it for processing:

- It must be type POOL.
- It must have a status of either ENABLED, QUIESCED NEW, or DISABLED NEW.
- It must have AM=Y, AM=I, or AM=P.

Deletion of dynamic I/O device considerations

Because DFSMSHsm serializes the UCB of each ML1 volume as soon as the volume is selected for use, you will not be able to delete the device the ML1 volume resides on until processing is complete. The serialization of the UCBs occurs during:

- Automatic volume space management
- Automatic secondary space management
- Command volume migration
- Command level 1 to level 2 migration

Space management considerations for sequential extended-format data sets

DFSMSHsm does migration and recall processing of single-stripe and multiple-stripe sequential extended-format data sets, using DFSMSdss as the data mover. The data sets must be SMS-managed.

Volume space management processing

Because sequential extended-format data sets span volumes, DFSMSHsm migrates a data set only when the volume that contains the first stripe is processed. Migration is skipped for this data set when volumes that contain subsequent stripes are processed.

DFSMSHsm expires an sequential extended-format data set only when the first stripe (volume) is being processed, regardless of whether the data mover is DFSMSdss or DFSMSHsm. Expiration is skipped when subsequent stripes (volumes) are being processed.

You can command migrate single-stripe or multiple-stripe sequential extended-format data sets.

Automatic primary space management processing

DFSMSHsm releases any allocated unused space only for a single-stripe sequential extended-format data set; allocated unused space will not be released for a multiple-stripe sequential extended-format data set.

Extent reduction will not be performed on either single-stripe or multiple-stripe sequential extended-format data sets.

If a sequential extended-format data set is small enough and meets all criteria—regardless of whether it is single-stripe or multiple-stripe—it can be migrated to a small-data-set-packing (SDSP) data set.

DFSMSHsm space management for sequential extended-format data sets will not occur for any of the following reasons:

- A sequential extended-format data set is not migrated if it is larger than the available free space on one DASD migration volume. Ensure that there is a DASD migration volume large enough to fit the entire data set or else migrate it directly to an ML2 tape volume or volumes.
- A multistripe sequential extended-format data set is not migrated if SETSYS USERDATASETSERIALIZATION is not specified for DFSMSHsm.
- A multistripe sequential extended-format data set is not migrated if it is protected by a RACF discrete profile.

Recall processing

DFSMSHsm recalls a sequential extended-format data set (indicated in the MCD record) using DFSMSdss as the data mover.

DFSMSdss recalls a sequential extended-format data set in the following ways:

- A sequential extended-format data set that was multiple-stripe when it was migrated could be recalled as a multiple-stripe or single-stripe sequential extended-format data set, depending on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.
- Conversely, a sequential extended-format data set that was single-stripe when it was migrated could be recalled as a multiple-stripe or single-stripe sequential extended-format data set, depending on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.
- A data set that was not sequential extended-format when it was migrated is always recalled as a data set that is not sequential extended-format. DFSMSHsm does not convert a data set that is not sequential extended-format to a sequential extended-format data set through recall processing.
- A sequential extended-format data set can be recalled as a data set that is not sequential extended-format on a non-SMS-managed volume by specifying FORCENONSMS on the RECALL command.
- A data set that was sequential extended-format when migrated is recalled as a non sequential extended-format but still SMS-managed data set if no SMS volumes supporting sequential extended-format data sets are available.

Recall fails if the ACS filter routines are modified to direct a data set that was sequential extended-format when it was migrated to a non-SMS-managed volume.

Note: If reblocking is required for a sequential extended-format data set, the reblocking is performed.

Object storage in the cloud

DFSMSHsm can use object storage in the cloud as a destination for migrated data set images. When data movement is offloaded to the storage controller, other management functions are performed directly using the z/OS Client Web Enablement Toolkit or the z/OS DFSMSdfp Cloud Data Access (CDA) services.. Using object storage in the cloud allows DFSMSHsm to avoid extra processing such as RECYCLE.

System set up to use the cloud

For DFSMSHsm to use the cloud, you must first set up your system. You can find a full list of the requirements and configuration options in the [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

- For transparent cloud tiering (TCT), DASD volumes in a DS8700 or higher with transparent cloud tiering need to be attached to the system. Only data sets that reside on a device with transparent cloud tiering can be migrated to the cloud through DFSMSHsm.

Note: When configuring DFSMSHsm in a Plex environment, it is highly advised that all DFSMSHsm's running in the Plex are cloud enabled. Doing so allows any DFSMSHsm instance within the Plex to run without error DFSMSHsm functions such as: AUDIT, Secondary Space Management, old cloud migration copy scratch, etc.

DASD volumes

For TCT, ensure that the first volume in an LSS that other volumes are attached from is attached to the system. Failure to do this can lead to attention messages from the DS8000 being missed.

SETSYS

When DFSMSHsm is enabled, it reconnects eligible data sets without performing data movement. If SETSYS CLOUDMIGRATION(RECONNECT(NONE)) is in effect, a subsequent migration of a data set that was otherwise eligible to be reconnected, causes the previous migration copy to be deleted and a new migration copy created.

Cloud password

With an SMS cloud configuration, DFSMSHsm supports CDACREDENTIALS or CLOUDCREDENTIALS option to manage passwords.

- **CDACREDENTIALS** – DFSMSHsm obtains the password for the cloud construct from CDA where the password has been entered by the security administrator. DFSMSHsm will store the encrypted password for future cloud requests. IBM recommends the CDACREDS option when using an SMS cloud configuration.
- **CLOUDCREDENTIALS** – When the CLOUDCREDENTIALS option is specified on the SETSYS CLOUD command to provide DFSMSHsm with cloud password for the user ID that is defined in the SMS network connection construct to DFSMSHsm. Great care needs to be taken with this password. DFSMSHsm keeps the password in encrypted form for later use. Anyone with access to this password can access the cloud directly and delete the migration copies, thus making DFSMSHsm unable to recall those data sets. It is recommended that the security administrator issue this command to protect the access to the cloud.

With a CDA provider configuration, DFSMSHsm does not store the credentials for traditional cloud providers. DFSMSHsm will store the password obtained from CDA for a TS7700 cloud provider in the same way as the CDACREDENTIALS option.

Using the Cloud

Workload distribution with TCT

Currently, the disk subsystem has a maximum of twelve threads per CEC for processing data to or from the cloud. The workload of migrate and recall processing must be evenly distributed. If the workload is not evenly distributed, queuing will occur within the storage controller.

During automatic space management and MIGRATE STORAGEGROUP processing, DFSMSHsm selects volumes for processing such that no more than twelve requests are issued for any volume within the same CEC. Once twelve volumes within the same CEC are being processed, no more volumes within that CEC are selected for processing until one of those volumes completes processing. Additionally, only two large data sets (greater than 5 GB) are migrated per CEC at any one time. These large data sets are revisited later if there are currently two being migrated.

The twelve concurrent request values can be changed via PATCH of MCVT.+494 X'nnnn'. For more information, see [Setting the maximum number of CEC volumes to be migrated to Cloud simultaneously in z/OS DFSMSHsm Implementation and Customization Guide](#).

MIGRATE DATASET and MIGRATE VOLUME processing do not participate in the distribution of work, so care must be taken when issuing these commands during Automatic Space Management and MIGRATE STORAGEGROUP processing, as queuing may still occur.

Migration

Retaining certain types of data on ML1 DASD or ML2 tape is undesirable. Migrating those data sets to the cloud can be desirable when the data sets are large, and not expected to be referenced for a long time.

When you migrate a data set to the cloud, it uses offboard data movement that is provided by the hardware. The elapsed time for data set migrations and recalls to and from cloud storage are longer than the time for migrations and recall functions on FICON attached storage. With cloud storage, data transfer is performed with TCP/IP across Ethernet connections. IBM recommends that installations perform tests to determine the expected elapsed time degradation unique for their environment. You can use DFSMSHsm FSR records to determine both elapsed time and CPU consumption for these functions. Based on the expected degradation, migrate to cloud storage only those data sets for which the degradation is acceptable. For example, migrated data sets that must be recalled within a specific time frame to achieve an application SLA, should be migrated to cloud storage only when the expected degradation is acceptable. Data set types that are both large in size, rarely or never recalled, are recommended for migration.

Because the backup format is different between an ML1/ML2 migration copy and a cloud migration copy, a data set cannot be moved from ML1/ML2 directly to the cloud. First, recall the data set to L0 DASD, then remigrate the data set to the cloud.

Note: When the catalog entry for the data set is migrated to the cloud, it has a *volser* of MIGRAT, like other migrations. However, the device type field in the catalog entry is x'00018000'. The third byte x'80' indicates that it is a tape, and the second byte x'01' indicates it is a cloud type.

If a data set was previously migrated to the cloud and has since been recalled, but is being remigrated back to the cloud, if it is not re-connectable DFSMSHsm will scratch the old cloud migration copy, depending on configuration, in one of three ways:

1. DFSMSHsm will scratch the old migration copy synchronously, before remigrating the data set.
2. DFSMSHsm will scratch the old migration copy asynchronously in conjunction with remigrating the data set, if asynchronous deletion of old migration copies from cloud is enabled.
3. DFSMSHsm will defer scratching of the old migration copy to when Secondary Space Management runs and the SETSYS MIGRATIONCLOUDDAYS *cloud_days* sub-parameter has been satisfied.

Recall

Recall of data sets from the cloud will be performed by the RECALL or HRECALL commands, or when the data set is referenced. No change to the command syntax is necessary. After a Recall, the migration copy of the data set will not be deleted. This enables subsequent reconnection if desired. When Secondary Space Management is performed, the Migration Control Data Set records will be cleaned up and the migration copies in the cloud will be deleted if the RETAIN DAYS number has been exceeded. DFSMSHsm will not delete empty containers, however. If empty containers are an issue, they can be deleted using a cloud management tool.

Delete

When you delete a migrated data set with the DELETE or HDELETE command, the migration copy in the cloud is also deleted.

Listing information

You can use the DFSMSHsm LIST command to get more information about data sets that have been migrated to the cloud. The LIST DATASETNAME MCDS command will show where the migration copy is stored in the cloud. Details such as the *cloud_network_connection_name* representing the cloud, the container name, and object pre fix are included in the output.

The LIST CLOUD command allows you to get successively more detailed views on the contents of the cloud. One way to use it would be to first get a view of what DFSMSHsm created containers are in the cloud through issuing the LIST CLOUD(*cloud_network_connection_name*) TERM command. Next, a more detailed view of one of the containers could be obtained through issuing LIST CLOUD(*cloud_network_connection_name*) CONTAINER(*container_name*) TERM. This will return a list of all the pseudo-directories within that container. Each pseudo-directory is a single migration copy of a data set.

A migration copy of a single data set is made up of many objects. They are grouped together in a pseudo-directory. DFSMSHsm object names use forward slash characters to delineate different levels. DFSMSHsm Storage Administrators may be familiar with the DFSMSHsm migration copy data set names. DFSMSHsm uses this same naming convention in the pseudo-directory name or object pre fix. If details are wanted about the objects that make up a particular migration copy, the LIST CLOUD(*cloud_network_connection_name*) CONTAINER(*container_name*) PREFIX(*pseudo-directory_name*) TERM command can be issued.

Auditing information

The AUDIT DATASETCONTROLS(MIGRATION) command has been updated to include migration copies in the cloud. While examining these records, if the record indicates the data set was migrated to the cloud, DFSMSHsm will check the cloud storage to verify that the expected number of objects for that prefix exist in the cloud. If the expected number of objects does not match what is found in the cloud it is likely that the migration copy has been corrupted.

Additionally, you can use the new `AUDIT MEDIACONTROLS(CLOUD(cloud_network_connection_name))` command to request DFSMShsm examine the DFSMShsm containers found in the cloud. DFSMShsm will examine each container in turn, comparing what is found in the container with the DFSMShsm migration control data set records. The `FIX` parameter will cause DFSMShsm to repair many of the errors it finds.

Chapter 5. Space management procedures

This topic contains information about:

- Interrupting and restarting primary space management, secondary space management, or event-driven migration.
- Defining new primary space management or secondary space management process windows.
- Allowing another host to take over secondary space management functions.

Interrupting and restarting primary space management, secondary space management, or event-driven migration

Primary space management, secondary space management, or event-driven migration can be interrupted during the course of their processing either intentionally or inadvertently. If you are not using the secondary host promotion function, when DFSMSHsm restarts, it resumes its work near the point where it left off.

If you are in a multiple DFSMSHsm-host environment, and are using the secondary host promotion function, a secondary host can take over the unique functions that were being performed by a primary host that has failed. Secondary host promotion also allows another host to take over the secondary space management functions from either a primary or secondary host that has failed.

This topic describes:

- When primary space management, secondary space management, or event-driven migration needs to be restarted.
- Automatic restart of primary space management or secondary space management.
- Automatic restart of event-driven migration.
- Manual restart of primary space management or secondary space management.
- How to define a new primary space management or secondary space management processing window.

When primary space management, secondary space management, or event-driven migration needs to be restarted

Table 17 on page 161 lists causes of interruptions to primary space management, secondary space management, and event-driven migration. It also lists how to restart these functions.

Table 17. Restarting primary space management, secondary space management, or event-driven migration

Cause of Interruption	How to Restart
System outage	IPL system, then start DFSMSHsm (S DFSMSHSM)
Stop DFSMSHsm (F DFSMSHSM,STOP)	Restart DFSMSHsm (S DFSMSHSM)
Hold automatic volume space management HOLD MIGRATION[(AUTO)]	Release automatic volume space management RELEASE MIGRATION[(AUTO)]
Hold automatic secondary space management HOLD MIGRATION[(AUTO)]	Release automatic secondary space management RELEASE MIGRATION[(AUTO)]
Cancel DFSMSHsm	Start (S DFSMSHSM)
SETSYS EMERGENCY	SETSYS NOEMERGENCY

Table 17. Restarting primary space management, secondary space management, or event-driven migration (continued)

Cause of Interruption	How to Restart
Migration held because output media is not available	ADDVOL, then RELEASE MIGRATION[(AUTO)]

Automatic restart of primary space management or secondary space management

If you are not using secondary host promotion, and there is an interruption caused by a system outage, a cancel of DFSMSHsm, or a stop of DFSMSHsm, the start of DFSMSHsm causes DFSMSHsm to attempt to restart primary space management or secondary space management. Primary space management or secondary space management automatically restarts if all of the following conditions exist:

- The time of the DFSMSHsm restart is in the primary space management or secondary space management processing window and the planned start time has not been changed.
- Primary space management or secondary space management is not held.
- DFSMSHsm is not in EMERGENCY mode.
- The operator gives permission if the SETSYS REQUEST command asks permission to restart.

Otherwise, if the planned start times have not been changed, you can manually restart primary space management or secondary space management.

If you are in a multiple DFSMSHsm-host environment and are using secondary host promotion, and the primary DFSMSHsm host or the DFSMSHsm host that runs secondary space management fails or goes into emergency mode, another DFSMSHsm host is promoted to take over the unique functions of the failed DFSMSHsm host. The unique functions performed by the primary DFSMSHsm host include:

- Space check needed for event-driven migration and recall placement of non-SMS data for interval migration.
- Automatic CDS backup.
- Automatic movement of backup versions from ML1 to tape.
- Automatic backup of migrated data sets.
- Expiration of dump copies.
- Deletion of excess dump VTOC copy data sets.

When the original primary DFSMSHsm host or secondary space management DFSMSHsm host restarts, and is not in emergency mode, it takes over those unique functions once more. However, if the promoted host is in the process of performing some of the functions it took over for the original host, the promoted host continues with those functions to completion. The original host does not attempt to perform its unique functions until the beginning of the next window.

By default, if you manually shut down the primary DFSMSHsm host or secondary space management DFSMSHsm host, secondary host promotion is not invoked, with the exception of the STOP DUMP command. You can override the default and cause secondary host promotion to be invoked when you manually shut down a primary DFSMSHsm host or secondary space management DFSMSHsm host by using the STOP PROMOTE command.

Automatic restart of event-driven migration

If the interruption was a system outage, a cancel of DFSMSHsm, or a stop of DFSMSHsm, the start of DFSMSHsm causes DFSMSHsm to check space on primary volumes eligible for automatic space management. If any such volume is over its threshold, DFSMSHsm then restarts event-driven migration if all of the following conditions exist:

- Primary space management or secondary space management is not held.

- DFSMShsm is not in emergency mode.
- The operator gives permission if the SETSYS REQUEST command asks permission to restart.

Manual restart of primary space management or secondary space management

You can restart primary space management or secondary space management from where it was interrupted if it did not complete and if the current time is still within the processing window. The planned start time is when primary space management or secondary space management last planned to start from the beginning.

When the current time is in the processing window

If primary space management or secondary space management does not restart automatically and the current time is still in the processing window, use one of the following commands to restart primary space management or secondary space management:

- If primary space management or secondary space management was held, issue the RELEASE MIGRATION(AUTO) command.
- If DFSMShsm was in emergency mode, issue the SETSYS NOEMERGENCY command.

When the current time is not in the processing window

- Primary space management:

If the current time is after the ending time, redefine an expanded processing window using the SETSYS PRIMARYSPMGMTSTART(*old-planned,new-end*) command. In the command, *old-planned* is the previous planned start time; *new-end* is a value greater than the current time. This command causes primary space management to restart. If you enter a value for a planned start time that is earlier than the previous planned start time, DFSMShsm also considers the request to be for an expansion of the existing processing window.

If you enter a value for the planned start time that is later than the previous planned start time, DFSMShsm considers the request to be for a new processing window, and the restart is not possible. When primary space management next runs, it will start from the beginning.

- Secondary space management:

If the current time is after the ending time, redefine an expanded processing window using the SETSYS SECONDARYSPMGMTSTART(*old-planned,new-end*) command. In the command, *old-planned* is the previous planned start time; *new-end* is a value greater than the current time. This command causes secondary space management to restart.

If you enter a value for the planned start time that is different than the previous planned start time, DFSMShsm considers the request to be for a new processing window, and a resumption from the point of interruption is not done. When secondary space management next runs, it will start from the beginning.

Where primary space management or secondary space management restarts

- Primary space management:

DFSMShsm records which volumes were processed during primary space management, so it will not attempt to process them again when it is restarted.

- Secondary space management:

If your installation is not using secondary host promotion, and secondary space management functions restart (without changing the *planned-start time*), they resume their work at approximately the point of interruption. They complete the work that would have run if the interruption had not occurred, or they continue until the ending time has been reached. If your installation is using secondary host promotion,

and the promotion occurs during a secondary space management window, and the window is not past the latest time to start (as defined on the primary host), then the promoted host takes over from approximately where the original host left off.

Note: Secondary space management consists of two level functions, migration level cleanup and level 1 to level 2 migration. A restart occurs in whichever was interrupted.

Defining a new primary space management or secondary space management processing window

As you increase the number of volumes that DFSMSHsm space manages, you might want to change the primary space management or secondary space management processing window, or you might want to add more tasks. This topic describes how to define a new processing window. For more information about adding more tasks, see [“Specifying the maximum automatic volume space management tasks and subtasks” on page 49](#) and [“Specifying the maximum automatic secondary space management tasks” on page 50](#).

You can define a new processing window whether primary space management or secondary space management is running or not.

When primary space management or secondary space management is running

You can change the primary space management or secondary space management processing window at any time, even when primary space management or secondary space management is currently running on a particular DFSMSHsm host. Primary space management or secondary space management will run when the new planned start time occurs if the currently running primary space management or secondary space management completes before the planned start time of your newly defined processing window. If the currently running primary space management or secondary space management runs past the new planned start time, primary space management or secondary space management will not start automatically until the next time the planned start time occurs.

When primary space management or secondary space management is not running

If you define a planned start time that is later than the time that primary space management or secondary space management last completed, primary space management or secondary space management starts at the planned start time.

If you define a planned start time that is earlier than the time that primary space management or secondary space management last completed, primary space management or secondary space management does not start until the next time the planned start time occurs on a scheduled day.

Allowing another host to take over secondary space management functions

In a sysplex, you can enable another host to take over secondary space management functions for either a primary or a secondary host that has failed. This function, called secondary host promotion, ensures continuous availability of DFSMSHsm functions.

Related reading

For more information about secondary host promotion key definitions and how to enable secondary host promotion from the SETSYS command, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Chapter 6. Availability management of SMS-managed storage

As described in [“Overview of DFSMSHsm functions”](#) on page 9, DFSMSHsm manages SMS-managed storage on a data set basis. That is, the specification for how to treat each data set on a volume is contained in the management class for that data set. The ACS routine can determine the storage group to which a data set is recovered.

This topic contains the following information:

- What the example system shows
- The tasks necessary to do availability management for SMS-managed storage
- Dump and recover tasks

What the example system shows

In the example system (see [Figure 4](#) on page 21) all of the availability management functions are demonstrated. The automatic and command functions described in this topic are also performed for non-SMS-managed storage. The example system shows the following three functions:

Automatic incremental backup

Every day except Sunday in the example system, all changed data sets except those in certain management classes are backed up. Backup starts between 3 a.m. and 4 a.m. with no new volumes started after 5 a.m. Backup is to tape. See [Figure 32](#) on page 193 for the complete set of commands that DFSMSHsm uses to process the incremental backup.

Host 1 backs up the control data sets at the beginning of backup (see Chapter 12, [“Maintaining DFSMSHsm control data sets,”](#) on page 493). It also moves data sets that have been backed up by command from ML1 volumes to backup volumes and backs up migrated data sets that are in need of backup (see [“Determining eligible data sets for backup”](#) on page 68).

Automatic full-volume dump

In the example system a full-volume dump is performed once a week on Sunday, starting between 3 a.m. and 4 a.m. and starting no new volumes after 5 a.m. All DFSMSHsm-managed volumes are dumped every week. In addition, volumes in storage groups 1 and 2 are dumped to an extra copy for off-site storage every 14 days; volumes in storage groups 3 and 4 are dumped to an extra copy for off-site storage every 28 days. See [Figure 26](#) on page 177 for the complete set of commands that DFSMSHsm uses to process a full-volume dump.

Command availability management

In the example system, commands cause availability management functions to occur creating the following conditions:

- One data set to be backed up.
- All the changed data sets on a volume to be backed up.
- A volume to be dumped.
- A backed-up data set to be recovered.
- A volume to be restored from a dump and forward recovered from later incremental backup versions. *Forward recovery* is a process of updating a restored volume by applying later changes as indicated by the catalog and the incremental backup versions.

- A volume to be restored from a dump copy.
- A volume to be recovered from backup versions.
- A specific data set to be restored from a dump volume.
- All expired backup versions to be deleted.
- A fast replication backup version to be recovered from a copy pool.

Related reading

For more information about the controls for non-SMS-managed storage, see [Chapter 7, “Availability management of non-SMS-managed storage,”](#) on page 303.

Tasks for automatic availability management

Now that the functions that the example system can accomplish have been described, the means to make the example system accomplish these functions are described. The tasks for controlling automatic availability management of SMS-managed storage require adding DFSMSHsm commands to the ARCCMDxx member and specifying attributes in the SMS storage classes and management classes. It is assumed that the storage classes and management classes have already been defined. The attribute descriptions explain the attributes to be added to the previously defined storage groups and management classes. Similarly, the descriptions of DFSMSHsm commands relate to commands to be added to the ARCCMDxx member of SYS1.PARMLIB.

Two groups of tasks are performed for availability management: dump tasks and backup tasks. Each of these groups of tasks is described separately.

Dump tasks

The dump tasks that are described consist of:

- Specifying which volumes to dump and the dump classes to use
- Specifying when automatic dump processing starts
- Specifying the DFSMSdss DASD I/O buffering technique to use for dump
- Specifying the maximum number of dump tasks
- Specifying the days on which dump occurs
- Specifying the characteristics of dump classes
- Defining dump volumes to DFSMSHsm

Specifying which volumes to dump and the dump classes to use

DFSMSHsm must be told what volumes to dump and what dump classes to use for the volumes to be dumped. This specification comes from the storage group or copy pool.

Storage group

The AUTO DUMP attribute specifies whether DFSMSHsm is to perform automatic dumps on the volumes assigned to the storage group. In the example system, the AUTO DUMP attribute for all storage groups is AUTO DUMP = Y.

The DUMP CLASS attribute determines two controls for DFSMSHsm: (1) the dump classes to which the volumes associated with the storage group are dumped and (2) how many dump copies are made when the dump occurs. The number of dump classes specified for any storage group determines the *maximum* number of copies that are made for any one dump. You can specify up to five dump classes for each storage group. Parameters in the dump class determine whether a dump copy is made for that particular dump class on that particular day.

The space management example (Figure 5 on page 34) shows how host 1 can be limited to doing space management only for storage group 1. Dump for any of the storage groups is not being limited to a single host. However, should you want to limit the processing to a single host, you can use the DUMP SYSTEM/SYSPLEX NAME attributes in the same way as the MIGRATE SYSTEM/SYSPLEX NAME attribute. Figure 24 on page 167 shows the dump management attributes that have been added to the storage groups in the example system.

```

NAME ==> SG1
  AUTO DUMP    ==> Y
  DUMP CLASS   ==> ONEWEEK
  DUMP CLASS   ==> TWOWEEK
  DUMP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG2
  AUTO DUMP    ==> Y
  DUMP CLASS   ==> ONEWEEK
  DUMP CLASS   ==> TWOWEEK
  DUMP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG3
  AUTO DUMP    ==> Y
  DUMP CLASS   ==> ONEWEEK
  DUMP CLASS   ==> FOURWEEK
  DUMP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG4
  AUTO DUMP    ==> Y
  DUMP CLASS   ==> ONEWEEK
  DUMP CLASS   ==> FOURWEEK
  DUMP SYSTEM/SYSPLEX NAME ==>

```

Figure 24. Storage Group Dump Attributes for Availability Management

For more information about storage groups and their function in SMS, see [z/OS DFSMSdfp Storage Administration](#).

Copy pool

The following fields in the COPY POOL DEFINE and COPY POOL ATLER panels provide DFSMSHsm with the required information to know what volumes to dump and what dump classes to use for the volumes to be dumped.

Field

Description and use

Auto Dump

The **Auto Dump** field specifies whether to do automatic dump processing for volumes in the copy pool. Type Y to enable automatic dump processing for the copy pool. If a generation 0 DASD version exists and the dump state of that version is either PARTIAL, FAILED, or NONE, automatic dump processes the version.

Dump Class

The **Dump Class** fields define the dump class name for fast replication dump to use. These fields are used by the DFSMSHsm dump function to know how to manage both automatic dump and FRBACKUP dumps of the copy pool. Up to 5 dump classes can be specified for the copy pool; therefore, up to 5 copies can be created per dump version.

Dump Sys/Sys Group Name

The **Dump Sys/Sys Group Name** field specifies the system to which this copy pool has affinity.

Number of DASD Fast Replication Backup Versions with Background Copy

The **Number of DASD Fast Replication Backup Versions with Background Copy** field specifies the number of backup version (0 to 85, or blank). Up to 85 DASD versions can be specified with a default of 2. If 0 is specified, it implies a NOCOPY environment. In a NOCOPY environment, unless the copy pool is defined with fast reverse restore capability, you cannot recover the backup version from DASD – the only purpose of the backup version is to be available as a source for dump processing.

Note: Expiration is performed at the dump class level for copy pool volumes. The last dump copy is never deleted during expiration processing or version roll-off. It is deleted only when specifically done so by command. Expiration processing of copy pool dump versions might extend the overall automatic dump window because of the additional CDS records that need to be accessed and any overlap with existing non-copy pool dump expiration.

For more information about the copy pool construct, see [“Using the copy pool construct for fast replication setup”](#) on page 257.

```

Panel  Utilities  Scroll  Help
-----
DGTDCPP1          COPY POOL [DEFINE|ALTER]          Page 1 of 5
Command ===>

SCDS Name . . . : IBMUSER.SCDs
Copy Pool Name  : COPYPL1

To DEFINE/ALTER Copy Pool, Specify:
Description ==> -----
=====
Auto Dump . . . N (Y or N)          Dump Sys/Sys Group Name . . . -----
Dump Class . . -----              Dump Class . . -----
Dump Class . . -----              Dump Class . . -----
Dump Class . . -----

Number of DASD Fast Replication Backup
Versions with Background Copy . . . . . 2 (0 to 85 or blank)
FRBACKUP to PPRC Primary Volumes allowed . . -- (NO, PN, PP, PR or blank)
FRRECOV to PPRC Primary Volumes allowed . . -- (NO, PN, PP, PR or blank)

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 25. COPY POOL DEFINE and COPY POOL ALTER panels, Page 1

Specifying when automatic dump processing starts

In the example system, automatic dump processing is started between 3 a.m. and 4 a.m. and no new volumes start after 5 a.m. The AUTODUMPSTART parameter of the SETSYS command controls the start of automatic dump processing. In the example system, the following SETSYS command is added to the ARCCMDxx member in each of the hosts:

```
SETSYS AUTODUMPSTART(0300 0400 0500)
```

Before automatic dump can start, you must have specified that backup be enabled in that particular host.

Related reading

For more information about how and when to run automatic backup, see [“Specifying whether to perform volume backup and the kind of device to use”](#) on page 189.

Specifying the DFSMSdss DASD I/O buffering technique to use for dump

With a single START I/O instruction, DFSMSdss can read 1, 2, or 5 tracks at a time, or a complete cylinder from the DASD being dumped. The DUMPIO parameter of the SETSYS command identifies which DFSMSdss DASD I/O buffering technique to use and the number of tracks to read.

The command is:

```
SETSYS DUMPIO(n,m)
```

where *n* indicates the DFSMSdss DASD I/O buffering technique for physical volume dump (using the BACKVOL command with the DUMP parameter or during automatic dump), and *m* indicates the value used for DFSMSdss logical dump (DFSMSdss specified as data mover on the SETSYS command).

The values used for n and m are:

- 1** DFSMSDss reads 1 track at a time
- 2** DFSMSDss reads 2 tracks at a time
- 3** DFSMSDss reads 5 tracks at a time
- 4** DFSMSDss reads 1 cylinder at a time

Note: If you specify DUMPIO without a value for m , then m defaults to the value specified for n .

If you do not specify the DUMPIO parameter on any SETSYS command, the DFSMSHsm default for n is 1 and for m is 4.

In the example system, a buffer of 5 tracks is used, so the SETSYS command that is added to the ARCCMDxx member in each of the hosts is:

```
SETSYS DUMPIO(3,1)
```

Specifying concurrent copy for volume dumps

By default, concurrent copy for volume dumps is disabled. To enable concurrent copy for volumes dumps, DFSMSHsm will request that DFSMSDss use concurrent copy, use the VOLUMEDUMP parameter of the SETSYS command.

If you are using concurrent copy for volume dumps and want to disable it, issue the following command:

```
SETSYS VOLUMEDUMP(NOCC)
```

Note: You can use STANDARD in place of NOCC as they are aliases.

For more information about using the VOLUMEDUMP parameter of the SETSYS command, see Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.

Specifying the maximum number of dump tasks

DFSMSHsm can run up to 64 dump tasks concurrently in each host. Because DFSMSHsm can make up to five dump copies concurrently for each dump task, the number of dump tasks does not limit the number of dump copies. In deciding how many dump tasks to specify, consider how many tape units are available to receive the dump copies.

The MAXDUMPTASKS parameter of the SETSYS command controls how many dump tasks run concurrently. In the example system, the use of three concurrent dump tasks is demonstrated. The following SETSYS command is added to the ARCCMDxx member in each host in the example system:

```
SETSYS MAXDUMPTASKS(3)
```

Because it is specified that each storage group have two dump classes, six tape units must be available for each host on the days when both dump classes are scheduled.

If any dump class involved specifies a STACK value of more than one, DFSMSHsm writes multiple dump copies sequentially as each tape is mounted for that dump class. That can reduce the number of tape drives (the number of dump tasks) needed concurrently, but the reduced parallelism can lengthen the total time needed for the automatic dump window specified by SETSYS AUTODUMPSTART.

The STACK value on the dump class (or BACKVOL command) might end up overriding the available MAXDUMPTASKS. Thus, available dump tasks to process dump copies might be under-utilized when the number dump copies from a dump operation divided by the STACK value is less than MAXDUMPTASKS.

The MINSTACK value in the dump class or the value on the BACKVOL command is used to determine how many dump tasks are to be utilized. In some cases, by specifying the minimum stack value, more available MAXDUMPTASKS is utilized as compared to specifying just the maximum stack value, STACK, but might also end up having fewer dump copies on the tape volume. When the dump tasks are selected, they are assigned the number of volumes specified with the MINSTACK keyword. When the maximum number of dump tasks are started, each dump task is then assigned additional volumes in sets equal to the MINSTACK value until the MAXSTACK value are reached.

See Table 18 on page 170 for an illustration about how tasks are selected and assigned volumes.

Table 18. Example of Selecting Tasks and Assigning Volumes

Scenario	Without MINSTACK	With MINSTACK(20)
<ul style="list-style-type: none"> • Volumes to be dumped: 300 • Dump tasks: 5 • MAXSTACK(100) specified for the dump class 	300 volumes are stacked onto 3 tapes (100 volumes on each tape), using 3 dump tasks.	<p>300 volumes are stacked onto 5 tapes (60 volumes on each tape), using 5 dump tasks.</p> <ol style="list-style-type: none"> 1. First 100 volumes are dumped. <ul style="list-style-type: none"> • 20 volumes are stacked onto 5 tapes, using all 5 dump tasks. 2. As the MAXSTACK value (100) is not reached, each of the dump tasks requests more volumes. Processing continues with sets of 20 volumes (the MINSTACK value), until the 300 volumes is processed.

When the SETSYS COMMONQUEUE(DUMP(CONNECT(*basename*))) command is issued, the preference by the system is to fully utilize the available dump tasks, as defined by MAXDUMPTASKS, on all the dump group members in the HSMplex. Tuning the MINSTACK and STACK values might increase dump task parallelism, and might shorten the total time needed for the dump operations to complete. Note that if this command is *not* specified, only local dump tasks where the command was issued are utilized.

Specifying the day on which a dump occurs

The DUMPCYCLE parameter of the DEFINE command specifies when the host does dump processing. You can also specify the day to start the cycle so that the cycle does not change with each reinitialization of DFSMSHsm. In the example system, the following command is added to the ARCCMDxx member in each host:

```
DEFINE DUMPCYCLE(NNNNNNYNNNNNNYNNNNNNYNNNNNNY -  
CYCLESTARTDATE(06/01/03))
```

The command establishes a 28-day cycle that begins on a Monday. Automatic dump processing occurs on Sunday. DFSMSHsm uses the dump cycle to determine the days on which automatic dump occurs. DFSMSHsm performs automatic dump only on days designated with a Y in the dump cycle. The dump cycle that you define applies to *all dump classes that you define*. Thus, the preceding example dump cycle applies to all of the dump classes that are defined in the following examples.

DFSMSHsm stores the date you specify with `CYCLESTARTDATE` as the date the dump cycle began. If you do not specify a `CYCLESTARTDATE` parameter with the `DEFINE` command but a cycle start date is already stored and the dump cycle length has not changed, DFSMSHsm uses the cycle start date that is already stored. If no cycle start date is stored and you do not specify one, DFSMSHsm uses the day that the `DEFINE DUMPCYCLE` command is issued as the cycle start date.

Rule: The date that is specified for the cycle start date cannot be a date in the future. The date must always fall on or before the date that the DEFINE command is issued.

Note:

1. When you redefine the dump class, be careful if you use the DAY parameter to define the days that a given dump class is active. The day that is specified with the DAY parameter should be a Y day in the DUMPCYCLE. If the day that is specified is an N day in the DUMPCYCLE, automatic dump processing does not target the dump class.
2. Dump cleanups run on N days in the dump cycle. Cleanup functions include expiration of expired dump copies and deletion of excess dump VTOC copy data sets and inactive empty containers. Expiration of expired dump copies and deletion of excess dump VTOC copy data sets occur on both 'Y' and 'N' days of the dump cycle. Empty dump container deletion only occurs on an 'N' day in the dump cycle.

Specifying the characteristics of dump classes

The dump classes to which a volume is dumped determine how the particular dump copies are made and used. You can control the following factors with the dump class:

- Whether the dump tapes are automatically reused when the dumped data is no longer valid
- Whether the dump tapes can be used to restore individual data sets
- The day of the dump cycle on which data will be dumped to the dump class
- What to do with the newly created dump tapes
- How often to dump data to the dump class
- Whether to reset the data-set-changed indicator
- When the data on the dump volumes becomes invalid
- The kind of tape unit to use for the dump tapes
- The number of generations for which copies of the VTOC of dumped volumes will be kept
- The expiration date to use in the tape header label
- The maximum number of dump copies to be stacked on a dump tape assigned to this dump class
- Whether to compress and encrypt the dump data
- Whether the dump class is required or optional

The DUMPCLASS parameter of the DEFINE command provides the control for the dump class.

There are three dump classes in the example system: ONEWEEK, TWOWEEK, and FOURWEEK. The copies in the ONEWEEK class are kept at the computer site and are used to restore complete volumes or individual data sets as necessary. The copies in the TWOWEEK and FOURWEEK classes are sent to another site to be available for returning the business to production should a disaster occur at the computer site. Because the DUMPCLASS parameter has many subparameters, the effect of each subparameter is described and the DUMPCLASS parameter is built as you proceed.

Controlling automatic reuse of dump tapes

The AUTOREUSE subparameter of the DUMPCLASS parameter controls whether dump tapes that have a retention period specified for them in the dump class are reused when the data expires. AUTOREUSE specifies that the tapes are returned to the DFSMSHsm tape pool when the dump copy is invalidated. NOAUTOREUSE specifies that the tapes are marked as unavailable for selection by DFSMSHsm when the dump copy is invalidated.

Because the volumes in dump class ONEWEEK are retained at the computer site, they can be reused when the data on them is invalidated. Therefore, begin building the DEFINE DUMPCLASS command to be added to the ARCCMDxx member in each host for this dump class as:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE)
```

Because the volumes in dump classes TWOWEEK and FOURWEEK are sent to another site, they cannot be automatically returned to use by the system. Therefore, add the following DEFINE DUMPCLASS commands to the ARCCMDxx member in each host for these dump classes:


```
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE)  
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE)
```

The AUTOREUSE subparameter is really of use only when you specify that empty tapes be returned to a DFSMSHsm pool of empty tapes that are known to DFSMSHsm. If tapes are returned to the scratch pool and if you are not selecting specific tapes for subsequent output tape selection, this subparameter is of no use because it makes tapes reusable only if they are known to DFSMSHsm.

In the example system, scratch tapes are used and empty tapes are released to the scratch pool. Therefore, although the specification of AUTOREUSE for dump class ONEWEEK was shown, it has no effect because volumes are not being defined to DFSMSHsm. However, the reason for specifying NOAUTOREUSE in dump classes TWOWEEK and FOURWEEK is valid. NOAUTOREUSE must be specified for tapes that are taken away from the computer system site.

Controlling restoring of individual data sets from dump tapes

You can control whether DFSMSHsm can use copies of data sets on dump tapes to recover individual data sets. If you allow restoring of individual data sets from dump copies in a particular dump class, DFSMSHsm selects the latest copy of the data set, whether it resides on a dump volume or a backup volume. If you do not allow restoring of individual data sets from dump copies in a particular dump class, DFSMSHsm cannot select the data set from a dump copy in that class.

Because it is intended to use the later of the copies on either the dump volumes in dump class ONEWEEK or the backup volumes, change the DUMPCLASS parameter to the following in the ARCCMDxx member in each host:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE DATASETRESTORE)
```

Because the volumes in dump classes TWOWEEK and FOURWEEK will be sent to another site, you do not want to choose them for restoring data sets. Therefore, change the DUMPCLASS parameter for these two classes in the ARCCMDxx member in each host by issuing the following commands:

```
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -  
NODATASETRESTORE)  
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -  
NODATASETRESTORE)
```

Note: The DATASETRESTORE keyword is not applicable to copy pool processing.

Specifying the day to dump data to this dump class

You can specify that you want data dumped to a particular dump class on a particular day in the dump cycle with the DAY subparameter. If you specify the DAY subparameter, the dump class is used only on that day. The day specified with the DAY subparameter must be a Y day in the dump cycle that you have defined with the DEFINE DUMPCYCLE command.

Because you want dump class ONEWEEK to be used four times during the dump cycle and you want dump class TWOWEEK to be used two times, do not specify the DAY subparameter for these two classes. The DAY subparameter does not allow you to use a dump class more than one time in a dump cycle. However, as dump class FOURWEEK is used once every 28 days, you can change its DEFINE DUMPCLASS command as follows:

```
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -  
NODATASETRESTORE DAY(28))
```

Specifying how often to dump data to the dump classes

You can control how often volumes are dumped to a dump class by using the FREQUENCY subparameter. This subparameter of the DUMPCLASS parameter works with the DUMPCYCLE parameter to determine when volumes are dumped. For a volume to be dumped, the dump must occur on a Y day in the dump

cycle and the number of days specified with the FREQUENCY subparameter must have passed. Use the FREQUENCY subparameter, rather than the DAY subparameter, when you want to use the dump class more than one time in the dump cycle.

In the example system, you want the volumes being dumped in dump class ONEWEEK to be dumped every seven days and the volumes in dump class TWOWEEK to be dumped every 14 days. (You also want the volumes being dumped to dump class FOURWEEK to be dumped every 28 days, but have taken care of this requirement with the DAYS subparameter.) To control dumping to dump classes ONEWEEK and TWOWEEK, you can change the DEFINE DUMPCLASS commands as follows:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -
DATASETRESTORE FREQUENCY(7))
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE FREQUENCY(14))
```

Specifying what to do with the dump volumes

As each DASD volume is dumped, DFSMSHsm notifies the operator that the volume is dumped and to which tape volumes that it is dumped. By using the DISPOSITION subparameter of the DUMPCLASS parameter, you can add to this message instructions about what to do with the dump volumes.

Because you are using the volumes from dump class ONEWEEK at the computer site, you have no particular instructions for the operator concerning these volumes. Therefore, do not add the DISPOSITION subparameter to the DEFINE DUMPCLASS command for this dump class. However, because you are sending the volumes from dump classes TWOWEEK and FOURWEEK to another site, you can change the DEFINE DUMPCLASS command for these dump classes as follows:

```
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE FREQUENCY(14) -
DISPOSITION('SEND OFFSITE BIWEEKLY'))
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
NODATASETRESTORE DAY(28) -
DISPOSITION('SEND OFFSITE MONTHLY'))
```

Specifying whether to reset the data-set-changed indicator

You can specify whether to reset the data-set-changed indicator on changed data sets when dump processing is performed to a specific dump class. If the dump operation resets the data-set-changed indicator, the data sets are not backed up by the next incremental backup operation unless the data set is changed after the dump occurs. If volumes are being dumped to multiple dump classes and you indicate in any of the dump classes that the indicator is to be reset, the indicator is reset whenever volumes are dumped to that dump class. If the dump copy is the most-recent backup copy of a data set and the data-set-changed indicator is reset, the most-recent backup copy can be lost if the dump copy expires automatically.

The RESET subparameter of the DUMPCLASS parameter controls whether the data-set-changed indicator is reset. If you intend to use incremental backup copies to update volumes restored from dump copies, you should use the NORESET option of this subparameter. You are going to use the incremental backup copies to update restored volumes; therefore, you can change the DEFINE DUMPCLASS command as follows:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -
DATASETRESTORE FREQUENCY(7) NORESET)
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -
NODATASETRESTORE FREQUENCY(14) -
DISPOSITION('SEND OFFSITE BIWEEKLY') -
NORESET)
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -
NODATASETRESTORE DAY(28) -
DISPOSITION('SEND OFFSITE MONTHLY') -
NORESET)
```

Note: You cannot specify RESET if the dump class is used for copy pool processing.

You can specify whether the data-set-changed indicator in the VTOC entry, DS1DSCHA, is reset for all data sets that are restored during full volume recover processing, using the RECOVERRESET subparameter of the DUMPCLASS parameter. The data-set-changed indicator indicates whether or not the data set has changed since its last backup. RECOVERRESET(YES), the default, species that the data-set-changed indicator is reset (set to OFF) for all data sets on the volume being recovered. RECOVERRESET(NO) species that the data-set-changed indicator is not reset. This option indicates that the data-set-changed indicator setting retains the value the data sets had when dumped. RECOVERRESET(DUMP) specifies that if the RESET keyword was specified in the dump class at the time the dump was created, the data-set-changed indicator is reset for all data on the volume being recovered, unless RESET was not specified, NORESET was specified, or the dump copy was created prior to z/OS V2R1 DFSMSHsm. Note that DUMP is not applicable for fast replication recovery because RESET cannot be specified in dump classes associated with the FRBACKUP function. For dump classes used during fast replication recovery processing, specify either YES or NO.

Specifying when the data on dump volumes becomes invalid

The data on dump volumes is of value until the data on the original volume has changed significantly. In the meantime, you are making new dump copies periodically. After some period, you want to indicate to DFSMSHsm that the old dump copies are no longer of value. You can do this with the RETENTIONPERIOD subparameter of the DUMPCLASS parameter.

Use the RETENTIONPERIOD subparameter to specify a number of days for which DFSMSHsm considers data on the dump volumes as valid. After the period specified by RETENTIONPERIOD has elapsed, the volumes become eligible to be reused, either automatically or manually.

Because you are making dump copies weekly for use at your site, consider all data on volumes dumped to dump class ONEWEEK to be invalid after three weeks. Similarly, consider data on volumes dumped to dump classes TWOWEEK and FOURWEEK to be invalid after three cycles. If you want to retain the dump copies until the full 100 generations have been made, you can specify NOLIMIT as the value for the retention period. Change the DEFINE DUMPCLASS commands as follows:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -  
  DATASETRESTORE FREQUENCY(7) NORESET -  
  RETENTIONPERIOD(21))  
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -  
  NODATASETRESTORE FREQUENCY(14) -  
  DISPOSITION('SEND OFFSITE BIWEEKLY') -  
  NORESET RETENTIONPERIOD(42))  
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -  
  NODATASETRESTORE DAY(28) -  
  DISPOSITION('SEND OFFSITE MONTHLY') -  
  NORESET RETENTIONPERIOD(84))
```

Specifying a tape expiration date

Some tape management programs require a tape expiration date in the tape header label. If you specify EXPIRATION or EXPIRATIONINCLUDE as tape security parameters, DFSMSHsm places a tape expiration date of 99365 in the tape header label. You use the TAPEEXPIRATIONDATE subparameter to cause DFSMSHsm to use a unique expiration date for the volumes rather than 99365. The example system does not use a tape expiration date.

Specifying the kind of tape units to use for dump volumes

The UNIT subparameter of the DUMPCLASS parameter limits the kind of tape unit to use for dump volumes when the dump function is not using a system-managed library.

Related reading

For more information about the UNIT subparameter of the DUMPCLASS parameter, see [Chapter 40, “DEFINE command: Defining control structures for use by DFSMSHsm,”](#) on page 809.

Specifying the maximum number of dump copies to stack on a dump volume

The STACK subparameter of the DUMPCLASS parameter directs DFSMSHsm to store one or more dump copies (each in its own file) on a dump tape, while that tape is initially mounted after selection during automatic dump.

Although the STACK value is intended to help you make better use of the newer, higher-capacity tape cartridges, DFSMSHsm does not limit its use to certain unit types, and does not reduce the amount of stacking to prevent dump copies from spanning to additional dump tapes. However, if a dump class specifies a STACK value of more than one, and that value results in a given sequence of dump copies spanning to one or more additional dump tapes, DFSMSHsm writes a message alerting you to that fact; you may want to reduce the STACK value.

Since the number of volumes to be dumped is not likely to be an integral multiple of the STACK value, the last dump tape assigned to a dump class with STACK value of M is likely to contain fewer than M dump copies.

DFSMSHsm does not try to extend existing dump volumes during initial selection, but selects only empty volumes as targets of full-volume dumps. Therefore, after a dump tape is dismounted, it can be taken offsite or whatever action is specified by the dump class DISPOSITION parameter.

All the dump copies stacked on a dump tape expire on the same day, but some copies may become invalid (rolled off) before that time.

Specifying the number of VTOC copies to keep

As DFSMSHsm makes a generation of dump copies, it makes a copy of the VTOC of the level 0 volume being dumped for that generation and associates it with the dump copies that are made. DFSMSHsm uses the VTOC copy when a dump copy is used to restore a data set. You can specify how many copies of the VTOC are kept. DFSMSHsm does not create a VTOC copy when an ML1 volume is dumped.

The example system demonstrates how VTOC copies are kept. It has been stated that storage groups 1 and 2 are using dump classes ONEWEEK and TWOWEEK and that storage groups 3 and 4 are using ONEWEEK and FOURWEEK. Every seven days a new generation is made for all volumes with a copy for dump class ONEWEEK. In addition, every 14 days the volumes in storage groups 1 and 2 are dumped to dump class TWOWEEK so that for those generations, two copies of the volumes in storage groups 1 and 2 are made. However, only one VTOC copy is made for each source volume in a generation even though two copies of the volume are made. [Table 19 on page 175](#) shows the copies in the various generations for one dump cycle.

In this example, the generation-numbering scheme is not the one used by DFSMSHsm but is a simplified scheme for instructional purposes.

Table 19. Generations of Example Dumps

Day	Generation	ONEWEEK	TWOWEEK	Discard VTOC Copy
7	1	X		None
14	2	X	X	None
21	3	X		1
28	4	X	X	None

Now, assuming that you have specified that you want two copies of the VTOC saved for each of dump classes ONEWEEK and TWOWEEK, complete the following actions:

1. When generation 1 is made, save one VTOC copy for dump class ONEWEEK.
2. When generation 2 is made, save one VTOC copy for both dump classes ONEWEEK and TWOWEEK.
3. When generation 3 is made, save one VTOC copy for dump class ONEWEEK and discard the VTOC copy from generation 1 because it exceeds the two VTOCs specified for dump class ONEWEEK.

4. When generation 4 is made, make one VTOC copy for dump classes ONEWEEK, TWOWEEK, and FOURWEEK; however, the VTOC copy from generation 2 cannot be deleted even though it exceeds the two VTOCs specified for dump class ONEWEEK, because it is still needed to satisfy the two VTOCs specified for dump class TWOWEEK. At this time, you have VTOC copies from generations 2, 3, and 4. The copies for generations 3 and 4 satisfy the two generations for dump class ONEWEEK. The copies for generations 2 and 4 satisfy the two generations for dump class TWOWEEK.

The VTOCCOPIES subparameter of the DUMPCLASS parameter controls the number of VTOC copies kept for the dump class. The dump VTOC copy data sets are written to ML1 volumes. The space required for these data sets varies with the size of the original VTOC, but it is less than that of the original VTOC because only part of each data set VTOC entry is written in the dump VTOC copy.

When no VTOC copy exists for a particular set of dump copies, DFSMSHsm can no longer use its automatic search capability to restore data sets from that dump (although it can restore entire volumes). Lack of a VTOC copy also prevents DFSMSHsm from providing listings of the data sets contained in that dump. *Authorized users* can restore data sets from a specified dump volume even though the VTOC copy does not exist.

In the example system, save two VTOC copies for dump classes ONEWEEK and TWOWEEK and no VTOC copy for dump class FOURWEEK. The DUMPCLASS parameters for the DEFINE commands are completed as follows:

```
DEFINE DUMPCLASS(ONEWEEK AUTOREUSE -  
  DATASETRESTORE FREQUENCY(7) NORESET -  
  RETENTIONPERIOD(21) UNIT(ACL) -  
  VTOCCOPIES(2))  
DEFINE DUMPCLASS(TWOWEEK NOAUTOREUSE -  
  NODATASETRESTORE FREQUENCY(14) -  
  DISPOSITION('SEND OFFSITE BIWEEKLY') -  
  NORESET RETENTIONPERIOD(42) -  
  UNIT(ACL) VTOCCOPIES(2))  
DEFINE DUMPCLASS(FOURWEEK NOAUTOREUSE -  
  NODATASETRESTORE DAY(28) -  
  DISPOSITION('SEND OFFSITE MONTHLY') -  
  NORESET RETENTIONPERIOD(84) -  
  UNIT(ACL) VTOCCOPIES(0))
```

Specifying whether to compress and encrypt the dump data

The ENCRYPT subparameter of the DUMPCLASS parameter directs DFSMSHsm to encrypt data dumped to the dump class. DFSMSHsm invokes DFSMSdss during the dump process with the encryption settings specified in the dump classes. For information about how to determine which encryption options to set in the dump class, see Chapter 6, "Managing Availability with DFSMSdss" in *z/OS DFSMSdss Storage Administration*.

The HWCOMPRESS subparameter of the DUMPCLASS parameter indicates whether DFSMSdss is to use hardware assisted compression for dump data before it is encrypted. Because encrypted data cannot be compressed, specify HWCOMPRESS(YES) to prevent a significant increase in the number of tapes required for the dump copies. HWCOMPRESS(YES) increases the CPU time consumed for the dump function.

The set of dump classes that constitute a dump generation (that is, the set of dump classes specified on the BACKVOL command or in the storage group definition) must have the same encryption and HWCOMPRESS settings, or the dump operation fails.

Specifying whether the dump class is required or optional

Using REQUIRED for a dump class ensures that dump class will be dumped to. Copy pools with dump classes that are defined with FASTREPLICATIONDUMP(REQUIRED) must have all REQUIRED dump classes successfully completed in order for the dump version to be considered successful. If an attempt to delete or roll off the DASD version of which a REQUIRED dump class has not been completed, the command fails. You can complete an incomplete dump copy by using the FRBACKUP DUMPPONLY function to retry the volumes that failed. To force a new DASD copy to be created to replace the existing

- Specifying the maximum number of backup versions to keep
- Specifying how long to keep backup versions
- Specifying who can back up data sets by command
- Specifying whether to perform automatic backup
- Specifying the start time for automatic backup
- Specifying whether to perform volume backups and the kind of device to use
- Specifying the first qualifier for names of backup versions
- Specifying the maximum number of volume backup tasks in each DFSMSHsm host
- Specifying the data set backup by command environment
- Specifying whether to back up RACF discrete data set profiles
- Specifying whether DFSMSHsm performs spill processing
- Specifying whether to back up only changed data sets
- Specifying how DFSMSHsm should respond to a data set in use
- Specifying the days on which automatic backup occurs
- Defining backup volumes to DFSMSHsm
- Specifying duplexing for backup tapes

There is also a discussion of the example system management classes in [“Example management classes”](#) on page 186.

Specifying which storage group has automatic backup performed

The AUTO BACKUP attribute of the storage group specifies whether DFSMSHsm performs automatic incremental backup on the volumes assigned to the storage group. For the example system, the AUTO BACKUP attribute for all storage groups is AUTO BACKUP = Y.

As with the space management example, backup for any of the storage groups is not limited to a single host. However, should you want to limit the processing to a single host, you can use the BACKUP SYSTEM/SYSPLEX NAME attribute in the same way as the MIGRATE SYSTEM/SYSPLEX NAME attribute was used. A sysplex name can also be used as the BACKUP SYSTEM/SYSPLEX NAME.

[Figure 27 on page 178](#) shows the backup attributes added to the storage groups for the example system.

```
NAME ==> SG1
  AUTO BACKUP ==> Y
  BACKUP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG2
  AUTO BACKUP ==> Y
  BACKUP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG3
  AUTO BACKUP ==> Y
  BACKUP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG4
  AUTO BACKUP ==> Y
  BACKUP SYSTEM/SYSPLEX NAME==>
```

Figure 27. Storage Group Attributes for Backup

Specifying how to manage groups of data sets

During backup processing, DFSMSHsm processes individual SMS-managed data sets to comply with the attributes specified in the management classes to which the data sets are associated. Management class backup attributes let you specify:

- Whether to use concurrent copy

- How often the data sets can be backed up
- How to guarantee frequency of backup
- How many backup versions (maximum) to keep
- How long to keep backup versions
- Who can back up data sets by command
- Whether to perform automatic backup

Related reading

For more information about management classes and their function in SMS, see [z/OS DFSMSdss Storage Administration](#).

Specifying a storage group priority for autobackup processing

The storage group processing priority of the volumes assigned to an SMS storage group may be specified within the storage group definition. The default value is 50. DFSMSHsm will process all of the SMS volumes in the order of the processing priority, with 100 being the highest and 1 being the lowest. This processing priority can help ensure that storage groups that need to be processed first during automatic backup are done so in order.

Specifying whether to use concurrent copy

Concurrent copy is a system-managed backup process. DFSMS provides a BACKUP COPY TECHNIQUE attribute for the SMS management class that informs DFSMSHsm whether to use the concurrent copy facility when backing up data sets during automatic, volume, or command data set backup processing. The BACKUP COPY TECHNIQUE attribute has seven parameters:

- CONCURRENT PREFERRED
- CONCURRENT REQUIRED
- STANDARD
- VIRTUAL PREFERRED
- VIRTUAL REQUIRED
- CACHE PREFERRED
- CACHE REQUIRED

These parameters function as follows:

- CONCURRENT PREFERRED (P) - specifies that the concurrent copy facility is preferred. If a concurrent copy session can be established, the data sets are backed up using the concurrent copy facility. Otherwise, the data sets are backed up by DFSMSdss logical dump without concurrent copy.
- CONCURRENT REQUIRED (R) - specifies that the concurrent copy facility is required. If a concurrent copy session cannot be established, the data sets will not be backed up.
- STANDARD (S) - specifies that the concurrent copy facility is not to be used, even though it may be available for the data sets. Normal DFSMSdss logical dump backup processing is used without concurrent copy.
- VIRTUAL PREFERRED (VP) - specifies that the virtual concurrent copy is preferred. If a virtual concurrent copy session cannot be established, the data sets are backed up by DFSMSdss logical dump without concurrent copy.
- VIRTUAL REQUIRED (VR) - specifies that the virtual concurrent copy is required. If a virtual concurrent copy session cannot be established, the data sets will not be backed up.
- CACHE PREFERRED (CP) - specifies that the cache-based concurrent copy is preferred. If a cache-based concurrent copy session cannot be established, the data sets are backed up by DFSMSdss logical dump without concurrent copy.

- CACHE REQUIRED (CR) - specifies that the cache-based concurrent copy is required. If a cache-based concurrent session cannot be established, the data sets will not be backed up.

When DFSMSHsm is performing either automatic, volume, or command data set backup processing, the management class associated with a data set specifies that a concurrent copy BACKUP COPY TECHNIQUE is to be used, and DFSMSdss is the data mover, then DFSMSHsm invokes DFSMSdss using the appropriate CONCURRENT parameter.

Note:

1. BACKUP COPY TECHNIQUE applies only to data sets residing on level 0 volumes. Migrated data sets are backed up using DFSMSdss logic dump.
2. Concurrent copy does not apply to catalogs and partitioned data sets.

Specifying frequency of backup

The backup attribute BACKUP FREQUENCY governs how frequently the data sets can be backed up automatically. Backup frequency is a management class attribute. This attribute specifies the number of days that must have elapsed since the last backup before the data set can be backed up again automatically, even though the data set may have changed and the volume it resides on is undergoing backup processing. For example, the data set will not be backed up because the frequency specification has not been met if the following conditions are true:

- The management class has a backup frequency of 2
- The data set was backed up yesterday
- The data set has been changed
- The volume is being processed for backup

However, if the backup frequency is 1 or if the data set had been backed up the day before yesterday, the data set is backed up.

The default for BACKUP FREQUENCY is 1.

Note: Normally, DFSMSHsm processes volumes only if they were not already processed within the previous 14 hours. A nightly backup starts over at the beginning. To have DFSMSHsm start the backup with volumes that were not backed up in the previous cycle, set the backup to a larger number of hours. For a patch command example, refer to a discussion on the patches to be added to the ARCCMDxx parmlib member in the [z/OS DFSMSHsm Diagnosis](#).

Guaranteeing frequency of backup

The maximum number of days between backups can be guaranteed by the storage group attribute GUARANTEED BACKUP FREQUENCY. This attribute provides an additional criterion to create a backup version and guarantees that a backup version of the data set is created after the specified number of days has elapsed, whether the data set has been changed. The backup version is created either during DFSMSHsm's automatic backup or command volume backup incremental processing.

Note: There is no interaction between GUARANTEED BACKUP FREQUENCY and command backup of a single data set or of a total volume.

If the data set has not changed and the backup version is created because the GUARANTEED BACKUP FREQUENCY has been met, then that backup version replaces the most recent backup version (replacement will not be performed for (H)BACKDS and BACKVOL TOTAL commands). If the data set has changed, backup versions are kept as described in the [“Specifying the maximum number of backup versions to keep”](#) on page 181 topic that follows.

GUARANTEED BACKUP FREQUENCY (which is a maximum) takes precedence over the management class attribute BACKUP FREQUENCY (which is a minimum), such as in the following: If the BACKUP FREQUENCY is set to six days and GUARANTEED BACKUP FREQUENCY is set to four days, the data set is backed up after four days, whether the data set has been changed. If the attribute values are reversed, the data set is backed up after four days, if changed, or after six days if not changed.

GUARANTEED BACKUP FREQUENCY provides a way to reduce the number of tape mounts needed to recover a volume when a few data sets are not being changed. Volume dumps copy all data, even if nearly all of it is already backed up by incremental backup. Having all the data needed for a recovery on a small number of tapes can reduce the number of tape mounts needed to recover a volume entirely from incremental backups.

If the data set's storage group attribute of AUTO BACKUP is NO, then no data sets are backed up by the automatic backup process, regardless of the setting of GUARANTEED BACKUP FREQUENCY. However, if an SMS volume is backed up by the BACKVOL command, GUARANTEED BACKUP FREQUENCY is honored for data sets that have not been changed.

Specifying the maximum number of backup versions to keep

Each time that DFSMSHsm makes a backup of a data set, the copy is known as a version. (Depending on the record length that is used to define the BCDS, DFSMSHsm can keep up to 29, or up to 100, versions of any data set.) The number of backup versions determines how far back you can go to recover a data set. You can specify the number of backup versions to keep by using the two NUMBER OF BACKUP VERSIONS attributes in the management class.

The NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) attribute controls the maximum number of backup versions you accumulate for data sets that are currently on either a level 0 volume or a migration volume. You will most likely want to keep multiple backup versions of these data sets. After you have as many versions of a data set as you specify with this attribute, each new backup version that is made of that data set causes the oldest backup version that exists to be discarded. The default for NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) is 2.

The NUMBER OF BACKUP VERSIONS (DATA SET DELETED) attribute controls the maximum number of backup versions you keep for a data set that has been deleted. DFSMSHsm does not automatically delete excess backup versions for deleted data sets at the time the data set is deleted. You must issue a DFSMSHsm EXPIREBV command that finds and deletes the excess backup versions of deleted data sets. The default for NUMBER OF BACKUP VERSIONS (DATA SET DELETED) is 1.

Note: When you delete a data set, control of the number of backup versions to be kept changes from the NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) attribute to the NUMBER OF BACKUP VERSIONS (DATA SET DELETED) attribute.

Related reading

For more information about how record length affects the number of backup versions, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Relationship of frequency and maximum number of backup versions

Proper use of the FREQUENCY and NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) attributes can give added assurance that data sets can be recovered. Assume that you specify a frequency of one day and a maximum number of versions of three. If the data set changes daily, the user has only two days to detect any errors that occur in the data set before the oldest backup version contains the error. If the error is not detected in two days, the user cannot use a backup version to recover the data set.

Now assume that you specify a frequency of two days and a maximum number of versions of three. If the data set changes daily, the user has six days to detect the error before the last backup version contains the error. However, each successive backup version will contain two days' worth of daily changes, and this may not provide enough granularity for your needs.

You can also control the number of days that a user has in which to detect errors by increasing the number of backup versions retained.

Specifying how long to keep backup versions

You can control how long to keep extra backup versions for data sets, how long to keep the only backup version of data sets that have been deleted, and how long to keep specific backup versions. (DFSMSHsm

does not delete the only backup version of a data set that exists unless you specifically delete the backup version.) The attributes that control the number of days to keep extra backup versions are overridden by the attributes that control the number of versions to keep. That is, if you have more backup versions than are specified by the appropriate NUMBER OF BACKUP VERSIONS attribute, the excess versions are deleted by the EXPIREBV command even though they may not have reached the age specified by the RETAIN DAYS attributes. The retention period for a specific backup version can be specified with the RETAIN DAYS keyword on the data set backup command. The value specified on the RETAIN DAYS keyword overrides all other retention values.

The RETAIN DAYS EXTRA BACKUP VERSIONS attribute controls the number of days that you keep backup versions that are older than the most recent one. You can specify values from 1 to 9999 or NOLIMIT for this attribute. If you use the NOLIMIT value, you can delete extra backup versions based on the following criteria:

- The value you assign to the NUMBER OF BACKUP VERSIONS attribute.
- The issuance of a DFSMSHsm command to delete specific backup versions.
- The value assigned to the RETAIN DAYS ONLY BACKUP VERSION attribute.

Note: All remaining backup versions are deleted after the retention period specified if you assign a value other than NOLIMIT.

The RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED) attribute controls how long the last backup version is retained after the data set has been deleted. You can specify values from 1 to 9999 or NOLIMIT. If you use the NOLIMIT value, the last remaining backup version can be deleted only by the value of the NUMBER OF BACKUP VERSIONS attribute being zero, or by a DFSMSHsm command to delete the specific backup version. The default for RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED) is 60. The number of days is counted from the time that DFSMSHsm detects that the data set has been deleted. DFSMSHsm detects that the data set has been deleted when the EXPIREBV command is processed.

When RETAIN DAYS ONLY BACKUP VERSION is set to other than NOLIMIT (1 to 9999), and the RETAIN DAYS EXTRA BACKUP VERSION is set to NOLIMIT, the resulting effect is to keep the number of backup versions specified by the NUMBER OF BACKUP VERSIONS (DATA SET DELETE) attribute for the amount of time specified by the RETAIN DAYS ONLY BACKUP VERSION attribute. After the retention period, all remaining backup versions are deleted. If NOLIMIT is specified for both RETAIN DAYS EXTRA BACKUP VERSION and RETAIN DAYS ONLY BACKUP VERSION, the backup versions are not deleted by EXPIREBV.

Relationship of the NUMBER OF BACKUP VERSIONS and RETAIN DAYS attributes

Consider a management class that specifies:

NUMBER OF BACKUP VERSIONS (DATA SET EXISTS)	= 4
NUMBER OF BACKUP VERSIONS (DATA SET DELETED)	= 2
RETAIN DAYS EXTRA BACKUP VERSIONS	= 20
RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED)	= 60

Now consider a single data set associated with this management class:

- As long as no backup version becomes more than 20 days old and the data set exists, no more than four backup versions are retained because of the action of NUMBER OF BACKUP VERSIONS (DATA SET EXISTS). Backup versions in excess of four are deleted even though they may not yet be 20 days old.
- If backup versions are no longer made (for example, because the data set is not being changed), the next processing of the EXPIREBV command deletes any backup version that is more than 20 days old except for the most recent backup version. (Remember, DFSMSHsm *always* requires you to take specific action to delete the last backup version of an existing data set.) The deletion of the oldest backup version is controlled by RETAIN DAYS EXTRA BACKUP VERSIONS. EXPIREBV treats the newest EXTRA backup copy as if it had been created on the same day as the newest BACKUP VERSION. Therefore, the newest EXTRA backup copy is not deleted until the number of days specified in RETAIN DAYS EXTRA BACKUP VERSIONS has passed, starting from the creation date of the newest backup copy.

Assume that the data set is deleted while four backup versions exist that are less than 20 days old:

- The next processing of the EXPIREBV command deletes the two oldest backup versions because of the action of NUMBER OF BACKUP VERSIONS (DATA SET DELETED). Deleting the data set caused a change in the controlling attribute.
- Since there are only two backup versions left, the oldest backup version is also the newest EXTRA backup version. EXPIREBV treats the newest EXTRA backup copy as if it had been created on the same day as the newest BACKUP VERSION. Therefore, the newest EXTRA backup copy is not deleted until the number of days specified in RETAIN DAYS EXTRA BACKUP VERSIONS has passed, starting from the creation date of the newest backup copy.
- When 60 days have passed since a DFSMSHsm EXPIREBV command has detected that the data set was deleted, the next processing of the EXPIREBV command deletes the last backup version through the action of RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED).

Specifying the number of days to retain a backup copy of a data set

You can keep an individual backup copy for either shorter or longer than the normal period of time by specifying the RETAIN DAYS keyword on the BACKDS command or specifying a RETAIN DAYS value in the backup exit (ARCBDEXT). RETAIN DAYS controls the minimum number of days that a backup copy of a data set is maintained. DFSMSHsm uses the RETAIN DAYS value to determine when a backup copy is to be rolled-off and during EXPIREBV processing.

The value of RETAIN DAYS can be in the range of 0–50000, which corresponds to the maximum of 136 years. If you specify 99999, the data set backup version is treated as never expire. For command processing, any value greater than 50000 (and other than 99999) causes a failure with an error message ARC1605I. When a value is specified via the ARCBDEXT exit, any value greater than 50000 (other than 99999) will be changed to 50000. A decimal value of 99999 will indicate that the backup copy should never expire. Any value less than 0 will be changed to 0. A retain days value of 0 indicates that:

- the backup version expires when the next backup copy is created,
- the backup version might expire within the same day that it was created if EXPIREBV processing takes place,
- the backup version is kept as an active copy before roll-off occurs,
- the backup version is not managed as a retained copy.

Note: You can use the RETAIN DAYS keyword only with cataloged data sets. If you specify RETAIN DAYS with an uncataloged data set, BACKDS processing fails with ARC1378I error message.

When you enter the EXPIREBV command, DFSMSHsm checks the retention days for each active backup copy for each data set, starting with the oldest backup version and ending with the third newest version. If the version has a specified retention days value, DFSMSHsm calculates the age of the version, compares the age to the value of the retention days, and expires the version if it has met its RETAIN DAYS. The second-newest version is treated as if it had been created on the same day as the newest backup version, and is not expired unless the number of retention days specified by RETAIN DAYS have passed since the creation of the newest backup version. EXPIREBV does not process the newest backup version until it meets both the management class retention values, and the RETAIN DAYS value.

Note: For non-SMS-managed data sets, the RETAIN DAYS value takes precedence over any of the EXPIREBV parameters.

During the EXPIREBV processing, DFSMSHsm checks the retention days for each retained backup copy. The retained copy is identified as an expired version if it has met its retention period. The EXPIREBV DISPLAY command displays the backup versions that have met their RETAIN DAYS value. The EXPIREBV EXECUTE command deletes the backup versions that have met their RETAIN DAYS value.

Related reading

For more information about active and retained backup copies, see [“Creating retained backup copies” on page 184](#).

Creating retained backup copies

DFSMSHsm maintains backup copies as active backup copies and retained backup copies. Active copies are the backup copies that have not yet rolled off. Retained copies are the backup copies that have rolled off from the active copies, but have not yet reached their retention periods. DFSMSHsm can maintain a maximum of 100 active copies. DFSMSHsm can maintain more than enough retained copies for each data set to meet all expected requirements.

Roll-off processing occurs when the number of existing active copies exceeds the maximum number of active backup versions, as defined by the management class attribute Number of backup versions (data exists) for SMS-managed data sets, and the value of the SETSYS VERSIONS command, for non SMS-managed data sets. When a new backup version causes the maximum number of active copies to be exceeded, DFSMSHsm checks all of the active copies to see if any active copy has retained days specified, and if the retained days value is met. The active backup copies are then rolled off in the following order:

1. Roll-off processing removes all backup versions that have expired RETAIN DAYS values.
2. If at least one active backup copy has retain days specified, but none of the active backup copies have met their retain days:
 - a. The oldest active backup copy is rolled-off if it does not have a RETAIN DAYS value specified.
 - b. If the oldest active backup copy has RETAIN DAYS value specified, then the oldest copy is replaced by the new backup copy, and the replaced active copy is maintained as a retained backup copy.
3. If none of the active backup copies have RETAIN DAYS values specified, the excess versions are deleted, starting with the oldest version. The maximum number of active backup copies is 100.

When an active backup copy is rolled off from the active copies, but has not yet reached its retention period, DFSMSHsm maintains it as a retained backup copy. DFSMSHsm can maintain more than enough retained copies for each data set to meet all expected requirements. Generation and version numbers are not associated with retained copies and cannot be used to recover them. You can recover retained copies only by specifying the date, and optionally time, that the version was created.

Example 1

The following example illustrates roll-off processing when several backup versions of a data set have met their retention periods. Assume, that:

1. the maximum number of active backup versions is 3,
2. the data set is cataloged,
3. 3 data set backup versions are already created.

Table 20. The state of the data set backup versions just before the fourth data set backup version is created on October 31st.

Version number	Date created	RETAIN DAYS value	RETAIN DAYS specified at backup time	RETAIN DAYS met indicator on Oct 31st
Version 3	Oct 30th	1	Yes	Met
Version 2	Oct 29th	2	Yes	Met
Version 1	Oct 28th	0	No	N/A

Table 21. The state of data set backup versions after roll-off processing completes. The second and the third versions rolled off

Version number	Date created	RETAIN DAYS value	RETAIN DAYS specified at backup time	RETAIN DAYS met indicator on Oct 31st
Version 4	Oct 31st	31	Yes	Not met

Table 21. The state of data set backup versions after roll-off processing completes. The second and the third versions rolled off (continued)

Version number	Date created	RETAIN DAYS value	RETAIN DAYS specified at backup time	RETAIN DAYS met indicator on Oct 31st
Version 1	Oct 28th	0	No	N/A

Example 2

The following example illustrates roll-off processing for active copies with retention periods that have not been met. Assume that:

- the maximum number of active backup versions is 3,
- the data set is cataloged,
- 3 data set backup versions are already created.

Table 22. The state of data set backup versions just before the fourth data set backup version is created on October 31st.

Version number	Date created	RETAIN DAYS value	RETAIN DAYS specified at backup time	RETAIN DAYS met indicator on Oct 31st
Version 3	Oct 30th	400	Yes	Not met
Version 2	Oct 29th	400	Yes	Not met
Version 1	Oct 28th	400	Yes	Not met

Table 23. The state of data sets after roll-off processing is complete. The oldest version rolled-off and became a retained backup copy.

Version number	Date created	RETAIN DAYS value	RETAIN DAYS specified at backup time	RETAIN DAYS met indicator on Oct 31st
Version 4	Oct 31st	400	Yes	Not met
Version 3	Oct 30th	400	Yes	Not met
Version 2	Oct 29th	400	Yes	Not met

Table 24. Retained backup copies after roll-off processing completes.

Version number	Date created	RETAIN DAYS value	RETAIN DAYS specified at backup time	RETAIN DAYS met indicator on Oct 31st
N/A	Oct 28th	400	Yes	Not met

Specifying who can back up data sets by command

The ADMIN OR USER COMMAND BACKUP attribute controls who can issue commands to back up data sets associated with the management class. You can specify values of BOTH, ADMIN, or NONE. If you specify BOTH, both DFSMSHsm-authorized and non-DFSMSHsm-authorized users can issue commands to back up data sets associated with the management class. If you specify ADMIN, only DFSMSHsm-authorized users can issue commands to back up data sets associated with the management class. If you specify NONE, data sets associated with that management class cannot be backed up either automatically or by command. The default for the ADMIN OR USER COMMAND BACKUP attribute is BOTH.

Specifying whether to perform automatic backup

The AUTO BACKUP attribute controls whether data sets that belong to the management class can be backed up by automatic backup processing. The values you can specify are Y (yes) and N (no). The default for AUTO BACKUP is Y. However, if the ADMIN OR USER COMMAND BACKUP attribute has the value of NONE, data sets associated with that management class are not backed up regardless of the value of the AUTO BACKUP attribute.

Example management classes

In the example system there are four management classes: standard, GDS, large, and no migration.

Standard management class

For data sets using the standard management class (STANDMC), the following actions occur:

- The data sets are automatically backed up no more frequently than every three days.
- Three versions of the data sets are retained while the data set exists.
- One version of the data set is retained for 60 days after the data set is deleted.
- Extra versions are retained for 20 days after they are created.
- Any users can issue commands to back up data sets.
- Automatic backup processing backs up the data sets.
- Concurrent copy is not used.

Figure 28 on page 186 shows the management class attributes for STANDMC.

```

COMMAND ===>
MANAGEMENT CLASS DEFINE
Page 3 of 5

SCDS NAME:          SYS1.SMS.SCDSA
MANAGEMENT CLASS NAME: STANDMC

TO DEFINE MANAGEMENT CLASS, SPECIFY:

BACKUP ATTRIBUTES
  BACKUP FREQUENCY          ===> 3      (0 to 9999 or blank)
  NUMBER OF BACKUP VERS     ===> 3      (1 to 100 or blank)
  (DATA SET EXISTS)
  NUMBER OF BACKUP VERS     ===> 1      (0 to 100 or blank)
  (DATA SET DELETED)
  RETAIN DAYS ONLY BACKUP VER ===> 60    (1 to 9999, NOLIMIT or blank)
  (DATA SET DELETED)
  RETAIN DAYS EXTRA BACKUP VERS ===> 20  (1 to 9999, NOLIMIT or blank)
  ADMIN OR USER COMMAND BACKUP ===> BOTH (BOTH, ADMIN, or NONE)
  AUTO BACKUP               ===> Y      (Y or N)
  BACKUP COPY TECHNIQUE     ===> S      (P,R,S,VP,VR,CP or CR)

```

Figure 28. Backup Attributes for STANDMC

GDS management class

Data sets in generation data groups use the GDS management class (GDSMC) for the following results:

- Changed data sets are automatically backed up every day.
- One backup version is retained while the data set exists.
- One backup version is retained for 14 days after the data set is deleted.
- No extra backup versions ever exist.
- Only DFSMSHsm-authorized users can issue commands to back up data sets.
- Automatic backup processing backs up data sets.
- Concurrent copy is not used.

Figure 29 on page 187 shows the management class attributes for GDSMC.

COMMAND ===> MANAGEMENT CLASS DEFINE Page 3 of 5

SCDS NAME: **SYS1.SMS.SCDSA**
 MANAGEMENT CLASS NAME: **GDSMC**

TO DEFINE MANAGEMENT CLASS, SPECIFY:

BACKUP ATTRIBUTES		
BACKUP FREQUENCY	===> 1	(0 to 9999 or blank)
NUMBER OF BACKUP VERS (DATA SET EXISTS)	===> 1	(1 to 100 or blank)
NUMBER OF BACKUP VERS (DATA SET DELETED)	===> 1	(0 to 100 or blank)
RETAIN DAYS ONLY BACKUP VER (DATA SET DELETED)	===> 14	(1 to 9999, NOLIMIT or blank)
RETAIN DAYS EXTRA BACKUP VERS	===>	(1 to 9999, NOLIMIT or blank)
ADMIN OR USER COMMAND BACKUP	===> ADMIN	(BOTH, ADMIN, or NONE)
AUTO BACKUP	===> Y	(Y or N)
BACKUP COPY TECHNIQUE	===> S	(P,R,S,VP,VR,CP or CR)

Figure 29. Backup Attributes for GDSMC

Large management class

Data sets that use this management class (LARGEMC) receive the following processing:

- The data sets are automatically backed up every other day (except Sunday).
- Four backup versions are retained while the data set exists.
- One version is retained for 60 days after the data set is deleted.
- Backup versions other than the most current are retained for 20 days after they are created.
- Only DFSMSshm-authorized users can issue commands to back up the data sets.
- Automatic backup processing backs up the data sets.
- Concurrent copy is not used.

Figure 30 on page 187 shows the management class attributes for LARGEMC.

COMMAND ===> MANAGEMENT CLASS DEFINE Page 3 of 5

SCDS NAME: **SYS1.SMS.SCDSA**
 MANAGEMENT CLASS NAME: **LARGEMC**

TO DEFINE MANAGEMENT CLASS, SPECIFY:

BACKUP ATTRIBUTES		
BACKUP FREQUENCY	===> 2	(0 to 9999 or blank)
NUMBER OF BACKUP VERS (DATA SET EXISTS)	===> 4	(1 to 100 or blank)
NUMBER OF BACKUP VERS (DATA SET DELETED)	===> 1	(0 to 100 or blank)
RETAIN DAYS ONLY BACKUP VER (DATA SET DELETED)	===> 60	(1 to 9999, NOLIMIT or blank)
RETAIN DAYS EXTRA BACKUP VERS	===> 20	(1 to 9999, NOLIMIT or blank)
ADMIN OR USER COMMAND BACKUP	===> ADMIN	(BOTH, ADMIN, or NONE)
AUTO BACKUP	===> Y	(Y or N)
BACKUP COPY TECHNIQUE	===> S	(P,R,S,VP,VR,CP or CR)

Figure 30. Backup Attributes for LARGEMC

No migration management class

For data sets that have value as long as they are used but lose value quickly when they are not used, the no migration (NOMIGMC) management class has been defined. Data sets that use this management class receive the following processing:

- The data sets are not backed up automatically.

- Both DFSMSHsm-authorized and non-DFSMSHsm-authorized users can issue commands to back up data sets.
- If a data set is backed up by command, one backup version is retained for 30 days after the data set is deleted.
- Concurrent copy is not used.

Figure 31 on page 188 shows the attributes for NOMIGMC.

MANAGEMENT CLASS DEFINE Page 3 of 5

COMMAND ==>

SCDS NAME: **SYS1.SMS.SCDSA**
 MANAGEMENT CLASS NAME: **NOMIGMC**

TO DEFINE MANAGEMENT CLASS, SPECIFY:

BACKUP ATTRIBUTES		
BACKUP FREQUENCY	==> 1	(0 to 9999 or blank)
NUMBER OF BACKUP VERS	==> 1	(1 to 100 or blank)
(DATA SET EXISTS)		
NUMBER OF BACKUP VERS	==> 1	(0 to 100 or blank)
(DATA SET DELETED)		
RETAIN DAYS ONLY BACKUP VER	==> 30	(1 to 9999, NOLIMIT or blank)
(DATA SET DELETED)		
RETAIN DAYS EXTRA BACKUP VERS	==>	(1 to 9999, NOLIMIT or blank)
ADMIN OR USER COMMAND BACKUP	==> BOTH	(BOTH, ADMIN, or NONE)
AUTO BACKUP	==> N	(Y or N)
BACKUP COPY TECHNIQUE	==> S	(P,R,S,VP,VR,CP or CR)

Figure 31. Backup Attributes for NOMIGMC

Specifying the start time for automatic backup

In the example system automatic backup processing starts at 3 a.m., and no later than 4 a.m.. New volumes are not started after 5 a.m. However, at the beginning of automatic backup, the primary host backs up the control data sets. The enqueue on the control data sets and reserve of volumes prevents automatic space management, automatic backup, and automatic dump from running simultaneously while the control data sets are being backed up. Whichever process starts first (space management, dump, or backup) delays the others from starting.

The AUTOBACKUPSTART parameter of the SETSYS command controls the time of day to start automatic backup processing. The SETSYS command that is added to the ARCCMDxx member in hosts 1, 2, and 3 for the example system is:

```
SETSYS AUTOBACKUPSTART(0300 0400 0500)
```

When DFSMSHsm backs up the control data sets, it issues an exclusive enqueue on the control data sets and, in a multiple DFSMS-host environment, reserves the volumes that contain the journal and the control data sets. The enqueue on data sets and reserve of volumes prevents automatic space management, automatic backup, and automatic dump from running simultaneously while the control data sets are being backed up. Whichever process starts first (space management, dump, or backup) delays the other from starting. You should consider this delay when you plan the start of automatic space management, the start of automatic backup, and the start of automatic dump.

The use of the concurrent copy function with DFSMSdss will significantly reduce the duration of complex-wide serialization of the DFSMSHsm CDS resources when performing CDS version backup. This reduction increases the availability of DFSMSHsm functions. To achieve maximum DFSMSHsm availability, you should place the CDS data sets on SMS-managed devices that support the concurrent copy function, and you should use DFSMSdss to back up the CDSs. If DFSMSdss backs up to tape, each CDS resource data set is copied to an individual tape drive (in parallel).

Specifying whether to perform volume backup and the kind of device to use

DFSMSHsm can use either tape or DASD as target volumes for backup. In the example system, all backup is to tape outside of a system-managed tape library. The BACKUP parameter of the SETSYS command controls whether a host is allowed to do backup and dump processing and limits the kind of device to use as target volumes. In the example system, the following SETSYS command is added to the ARCCMDxx member in each host:

```
SETSYS BACKUP(TAPE(ACL))
```

The BACKUP parameter specifies to DFSMSHsm that the host is allowed to perform backup and dump processing. The TAPE(ACL) subparameter specifies that backup versions are to be written to tapes mounted on the devices associated with the ACL esoteric name.

The SETSYS command has a DENSITY parameter that is related to mounting a scratch backup tape for the backup unit. However, when the tape unit specified is a cartridge-type device, DFSMSHsm ignores the DENSITY parameter.

To prevent the host from performing backup, you would use the NOBACKUP parameter, which prevents both automatic and command backup processing from occurring. The NOBACKUP parameter also prevents automatic and command dump from occurring.

Related reading

For information about how to direct volume backup to a tape library, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Specifying the first qualifier for names of backup versions

The backup prefix is used in the ACS filters that prevent DFSMSHsm-owned data sets from being allocated on SMS-managed volumes. In addition, the use of the prefix allows you to use RACF generic profile protection for all your backup versions.

When DFSMSHsm backs up data sets, it leaves the original data set name in the catalog. For use in its own records of the data sets, DFSMSHsm generates its own names for the backup versions. The data set name that DFSMSHsm generates is of the form:

```
prefix.BACK.Tssmmhh.user1.user2.Xydd
```

If patch-enabled, the following dataset name will be generated:

```
prefix.BACK.Tcccchh.user1.user2.Xydd
```

For more information, refer to [Changing DFSMSHsm backup and migration generated data set names to reduce contention for similar names and eliminating a possible performance degradation in z/OS DFSMSHsm Implementation and Customization Guide](#).

The BACKUPPREFIX parameter of the SETSYS command specifies the *prefix* (high-level qualifier) of the generated name. If you do not specify a backup prefix, DFSMSHsm uses the UID you specified in the startup procedure. In the example system the following command is added to the ARCCMDxx member in each host:

```
SETSYS BACKUPPREFIX(HSM)
```

Related reading

- For more information about the ACS filters, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

- For more information about the data set name, see [“Generating names of backup versions of data sets” on page 204.](#)

Specifying the maximum number of volume backup tasks in each host

DFSMSHsm can have up to 15 volume backup tasks running concurrently in each host. The MAXBACKUPTASKS parameter of the SETSYS command controls the number of concurrent volume backup tasks that will run. To decide how many concurrent volume backup tasks DFSMSHsm should run, consider how many tape units are available if you use tape backup volumes or how many backup DASD units are available. If you are using the duplex tape option for backups, you will need two tape units for every backup task. Also, consider the system work load when volume backup is running. The default for this parameter is 2.

Note: In a multiple DFSMSHsm-host environment with concurrent multitasking, it is beneficial to run all tasks on a single host. For example, having eight tasks running automatic backup from one host gives you better performance than splitting the tasks and running four on one host and four on another host.

In the example system, the maximum backup tasks are specified as:

```
SETSYS MAXBACKUPTASKS(3)
```

Specifying the data set backup by command environment

The data set backup by command function has a number of options that can control your data set backup operation. These options include the number of tasks, the output device type, and tape management.

Related reading

For more information about how to tailor the command data set backup environment, see [“Using the data set or UNIX file backup by command capabilities” on page 224.](#)

Specifying whether to back up RACF discrete data set profiles

If you are using RACF discrete profiles rather than generic profiles, you can have DFSMSHsm back up the profile when it backs up the data set. The PROFILEBACKUP parameter of the SETSYS command controls whether discrete profiles are backed up. Because the example system is in a RACF always-call environment and uses generic profiles, choose the NOPROFILEBACKUP option of this parameter. In the example system, the following command is added to the ARCCMDxx member in each host:

```
SETSYS NOPROFILEBACKUP
```

Specifying whether DFSMSHsm performs spill processing

When DFSMSHsm performs backup to DASD volumes, the DASD volumes can become full. If this occurs, DFSMSHsm can perform the spill process, which moves older backup versions of data sets to other volumes, known as *spill volumes*. Spill volumes are typically tape volumes.

Because the example system uses tape for backup volumes, no spill processing is possible. Therefore, the following command is added to the ARCCMDxx member in each host:

```
SETSYS NOSPILL
```

Specifying whether to back up only changed data sets

When DFSMSHsm is first installed, you might have many data sets on your volumes with the data-set-changed indicator in the data set VTOC entry off (such as data sets for which the bit has been turned off by some other program product). Because DFSMSHsm uses the data-set-changed indicator to decide if a data set needs to be backed up for incremental backup, these data sets would not be backed

up. DFSMSHsm provides an INCREMENTALBACKUP parameter for the SETSYS command that allows you to specify whether to back up such data sets. If you specify the ORIGINAL subparameter of the INCREMENTALBACKUP parameter, DFSMSHsm searches its records for each data set that does not have the data-set-changed indicator turned on. If DFSMSHsm has a backup copy of the data set, the data set is not backed up again. If DFSMSHsm does not have a backup copy of the data set, a backup version is made.

Of course, searching the records each time a data set is found with the data-set-changed indicator turned off takes a lot of time. You may want to use the ORIGINAL subparameter for the first few backup cycles, then change to the CHANGEDONLY subparameter. When you use the CHANGEDONLY subparameter, DFSMSHsm backs up only those data sets that have the data-set-changed indicator turned on.

In the example system, it is presumed that all old data sets have been backed up or have their data-set-changed indicators set to on, so the following command is added to the ARCCMDxx member in each host:

```
SETSYS INCREMENTALBACKUP(CHANGEDONLY)
```

Specifying how DFSMSHsm should respond to a data set in use

To ensure integrity and consistency of the data in a backup version, DFSMSHsm tries to serialize on system resources to indicate whether a data set is in use. This prevents the data set from changing while DFSMSHsm is backing it up. However, accepting the philosophy that something is better than nothing, you may still want backup copies of data sets that are in use at the time of backup, so long as you can identify (if you should need to recover from such a backup) that a given backup was created while the data set was possibly being changed.

If DFSMSHsm makes an unserialized backup, after first trying and failing to serialize on system resources, the data-set-changed indicator remains on; the data set is still eligible to be backed up. If DFSMSHsm is configured *not* to serialize on system resources on the first attempt to back up a data set, the data-set-changed indicator is turned off. The instruction for no serialization is an option of the ARCHBACK user macro and the ARCBDEXT installation exit. However, the ARCINBAK inline backup program always specifies this option, because it expects the JCL to perform its own serialization.

DFSMSHsm recognizes two situations involving backup and a data set being in use. First, there are data sets that you do not expect to be open for update, but that may be in use when backup is attempted. For this situation, the SETSYS BACKUP parameter has an INUSE subparameter to control how DFSMSHsm can try a second time, after a specified delay, to serialize on a data set before backing it up, and whether successful serialization for such a retry is required or merely preferable. The backups that the INUSE subparameter allows may be placed temporarily on ML1 DASD.

In the example system it is presumed that you want DFSMSHsm to attempt such a retry if needed, after a delay of 10 minutes, and to back up the data set even if it is still (or again) in use on the second try. The following command is added to the example system:

```
SETSYS BACKUP(INUSE(RETRY(Y) DELAY(10) +  
SERIALIZATION(PREFERRED)))
```

For the second situation, there are data sets (typically a database) that you expect to be open for update most of the time. You can use the data set backup installation exit ARCBDEXT for additional control, to direct DFSMSHsm either not to serialize before making a backup or to back up the data set even if serialization fails. The SETSYS INUSE subparameters can also apply in this situation, but the exit can override them for a given data set.

Related reading

For more information about the installation exit, see [z/OS DFSMS Installation Exits](#).

Specifying the days on which automatic backup occurs

The BACKUP parameter of the DEFINE command specifies a backup cycle that specifies the days on which the host does automatic backup processing. You can also specify the day to start the cycle so that the cycle does not change with each re-initialization of DFSMSHsm. In the example system, the following command is added to the ARCCMDxx member in each host:

```
DEFINE BACKUP(YYYYYYN CYCLESTARTDATE(1995/01/06))
```

The command establishes a 7-day cycle that begins on a Monday. Backup processing is done on every day except Sunday.

DFSMSHsm stores the date you specify with CYCLESTARTDATE as the date the backup cycle began. If you do not specify a CYCLESTARTDATE parameter with the DEFINE command but a cycle start date is already stored and the backup cycle length has not changed, DFSMSHsm uses the cycle start date that is already stored. If no cycle start date is stored and you do not specify one, DFSMSHsm uses the day that the DEFINE BACKUP command is issued as the cycle start date.

If you are backing up only to DASD, you should also specify how many DASD backup volumes you want DFSMSHsm to use for each day in the backup cycle.

Rule: The date specified for the cycle start date cannot be a date in the future. The date must always fall on or before the date that the DEFINE command is issued.

Related reading

For more information about defining the desired number of DASD backup volumes, see [“BACKUP: Specifying the backup cycle”](#) on page 814.

Defining backup volumes to DFSMSHsm

If you are using SELECTVOLUME(SPECIFIC) and TAPEDELETION(HSMTAPE), you may want to identify your backup volumes to DFSMSHsm. Use the ADDVOL command to identify the volume as a backup volume and to specify the following:

- The kind of unit for the volume
- Whether the volume is a daily backup volume or a spill volume

The following provides a sample ADDVOL command for adding a DASD daily backup volume on a unit 3390:

```
ADDVOL BAK001 UNIT(3390) BACKUP(DAILY(5))
```

This command makes BAK001 a daily backup volume that is used on day five of the backup cycle.

Figure 32 on page 193 shows the complete set of DFSMSHsm commands that the example system uses to control automatic backup processing for SMS-managed storage.


```

/* SPECIFYING THE TIME AUTOMATIC BACKUP IS TO START */
/* (PROCESSING UNITS 1, 2, AND 3) */
SETSYS AUTOBACKUPSTART(0300 0400 0500)
/* SPECIFYING THE TIME AUTOMATIC BACKUP IS TO START */

/* SPECIFYING TO DIRECT BACKUP TO TAPE, AND TO RETRY */
/* BACKUP WHEN A DATA SET IS IN USE. */
SETSYS BACKUP(TAPE(ACL) INUSE(RETRY(Y) DELAY(10) +
                SERIALIZATION(PREFERRED)))

/* SPECIFYING FIRST QUALIFIER FOR NAMES OF BACKED UP */
/* DATA SETS */
SETSYS BACKUPPREFIX(HSM)

/* SPECIFYING MAXIMUM NUMBER OF BACKUP TASKS */
SETSYS MAXBACKUPTASKS(3)

/* SPECIFYING BACKUP OF RACF DISCRETE DATA SET */
/*PROFILES */
SETSYS NOPROFILEBACKUP

/* SPECIFYING SPILL PROCESSING */
SETSYS NOSPILL

/* SPECIFYING BACKUP OF ONLY CHANGED DATA SETS */
SETSYS INCREMENTALBACKUP(CHANGEDONLY)

/* SPECIFYING DAYS FOR AUTOMATIC BACKUP TO OCCUR */
DEFINE BACKUP(YYYYYYN CYCLESTARTDATE(1995/01/06))

```

Figure 32. DFSMShsm Automatic Backup Commands for SMS-Managed Storage

Specifying duplexing for backup tapes

The duplex tape option creates two identical tapes in parallel, with one tape designated as the original and one the alternate. To create duplex backup tapes, specify:

```
SETSYS DUPLEX(BACKUP(Y))
```

If duplexing is not desired for your backup tapes, specify:

```
SETSYS DUPLEX(BACKUP(N))
```

The DFSMShsm default is no duplexing.

Automatic availability management operation

Previously, it has been pointed out how the commands to control DFSMShsm are structured for the example system. In the text that follows, the example system is used to describe what DFSMShsm does in the process of automatic availability management. Dump and backup are shown as two separate processes that contribute to total availability management.

The example system uses cartridge-type tape units in single-file mode for backup volumes. Also, the example system always selects scratch volumes because there are no volumes identified to DFSMShsm as dump or backup tape volumes and all partial tapes are marked full when they are demounted. The example system defers requests for tape mounts until OPEN processing for the data set occurs. These parameter selections minimize interference with other DFSMShsm functions when a tape mount is requested.

Dump processing

Figure 2 on page 17 shows the example system for dump processing. For this process, the dump volumes are mounted on the cartridge-type devices that are designated as 3490 with the automatic cartridge loader feature. Because dump processing moves all of the data sets on the volumes that are processed, any data set names on the volumes in each group are not shown.

Assume that it is 3 AM on 2003/02/02. This is day 28 of the dump cycle that was established with the DEFINE command. Unless a specific action is taken to prevent it, all hosts that are active begin automatic dump processing for the following reasons:

- BACKUP in the SETSYS command is specified.
- AUTODUMPSTART(0300 0400 0500) in the SETSYS command is specified.
- It is a Y day in the dump cycle.
- NOREQUEST in the SETSYS command is specified.

For SMS active hosts, at the beginning of automatic dump, DFSMSHsm obtains volume lists from storage groups that are defined to automatic dump enabled copy pools and storage groups that are pool type, automatic dump enabled, and have a status of ENABLED, QUIESCED NEW, or DISABLED NEW. If a storage group is both automatic dump enabled and defined in an automatic dump enabled copy pool, it is processed twice. If a storage group is in more than one copy pool, it can be processed multiple times. Volume priority for dump processing is determined by copy pool association, dump class stacking settings, system affinity, and SMS association. If SMS is not installed or not active, DFSMSHsm processes only non-SMS managed DASD volumes.

When DFSMSHsm is processing copy pools, volumes are considered candidates if the frequency requirement for the dump class to which the volume belongs has been met.

When DFSMSHsm processes volumes within automatic dump enabled storage groups, those volumes are only considered candidates if:

- The volume has not been processed in any host by automatic dump in the last 14 hours. Any dump that is created during copy pool processing does not apply to this 14 hour window.
- The frequency requirement that is specified for the dump class to which the volume belongs has been met.

A copy pool can be dumped at automatically scheduled intervals with Automatic Dump function. If Auto Dump = Y is specified in the copy pool construct, automatic dump checks generation 0 in the copy pool. If a DASD copy generation 0 exists and has not been previously successfully dumped, automatic dump dumps it. Automatic dump uses the Dump Class specified in the copy pool construct. If the GEN(0) version is partially complete, the failed volumes in the dump classes previously attempted are retried. The automatic dump Y|N settings of the source volumes' storage groups are ignored by automatic dump when it is processing a copy pool. The 14- hours-between-dumps limitation is also ignored by automatic dump when it is processing a copy pool.

DFSMSHsm first determines whether there are volumes with dumps that can be stacked. Those volumes associated with the same set of eligible dump classes, of which at least one has a STACK value of more than one, are considered "stackable" together. Within those volumes, the order is:

1. Affinity copy pool volumes
2. Non-affinity copy pool volumes
3. Affinity SMS volumes
4. Non-affinity SMS volumes
5. Non-SMS volumes

Volumes associated with copy pools are not stacked with non-copy pool volumes. Outside of copy pool associated volumes, dumps of SMS-managed and non-SMS-managed volumes can exist on the same dump tape. Affinity and non-affinity volumes cannot be stacked on the same tape. For more information about dump stacking, see [“DUMPCCLASS\(STACK/MAXSTACK\) and DUMPCCLASS\(MINSTACK\): Specifying the maximum and minimum number of dump copies that are stacked for this dump class” on page 825.](#)

After all volumes eligible for stacking have been dumped, the remaining volumes are dumped, with DFSMSHsm selecting the order of processing as listed above.

After DFSMSHsm has attempted to process all the candidate volumes eligible for automatic dump, it retries the selection of any of these volumes found in use by another DFSMSHsm function.

Dump processing is performed in four phases in the primary host and in one phase in the non-primary hosts. These phases are:

- Delete expired dump copies (primary host only).
- Dump DFSMSHsm-managed volumes for which automatic dump is requested.
- Delete and uncatalog excess dump VTOC copy data sets (primary host only).
- Delete DFSMSHsm inactive empty dump containers (primary host only).

Note: Empty dump container deletion occurs only on an 'N' day in the dump cycle, as specified in the [“DUMPCYCLE: Specifying the automatic dump cycle”](#) on page 829 command.

If a secondary host is made eligible to take over for a primary host that has failed, it can perform the expiration of dump copies, and deletion of excess dump VTOC copy data sets.

Deleting expired dump copies

In the first phase of automatic dump on the primary host, the expired dump copies are deleted. Dump copies become eligible to be automatically deleted when they expire or when the maximum number of dump generations is reached and a new generation of dump copies is to be made. Dump copies are automatically deleted after the number of days that are specified in the RETENTIONPERIOD subparameter of the DUMPCLASS parameter of the DEFINE command elapsed. This phase of automatic dump causes the status of the dump tapes containing the deleted copies to be changed from expired to available if they remain under DFSMSHsm control.

Automatic dump expiration is performed at the dump class level for both copy pool volumes and noncopy pool volumes residing on tape and in cloud storage. DFSMSHsm does not automatically delete the last and only remaining copy of a source volume. The dump volumes that make up this dump copy remain in EXPIRED status. To determine which dump tapes are currently in EXPIRED status, issue the LIST DUMPVOLUME command with the SELECT(EXPIRED) option. To delete the dump volumes in this copy, issue the DELVOL command with the LASTCOPY option for one of the dump volumes. The COPYPOOLCOPY keyword must also be specified if the dump volumes belong to a copy pool version. All volumes in the copy are then deleted. You can delete the last dump class for a copy pool version using the FRDELETE command. Automatic dump expiration of copy pool copies might extend the automatic dump window because of the additional CDS records that need to be accessed and any overlap with existing non-copy pool dump expiration.

To list cloud dump copies with creation and expiration information, issue the LIST PRIMARYOVLUME(*srcvol*) ALLDUMPS BCDS command. To delete the last non-copy pool dump copy residing in cloud storage, issue DDELETE VOLUME(*srcvol*) LASTCOPY DATE(*yyyy/mm/dd*) TIME(*hhmmss*) command.

Deleting empty dump containers

Auto dump processing on a primary host scans for inactive empty DFSMSHsm dump containers and deletes them. A dump container within the current creation period is considered active and, therefore, is not deleted.

Defining what to do with tapes marked unavailable during DUMP processing

When dump processing encounters an I/O error on the output tape, DFSMSHsm marks the tape unavailable. The tape is left under DFSMSHsm's control to prevent re-release of the unavailable tape. This prevents DFSMSHsm from selecting the tape for scratch, and prevents other system functions from encountering the unavailable tape before reinitialization.

Use the following procedure to remove the tape from DFSMSHsm's control:

- Issue the LIST DUMPVOLUME SELECT(UNAVAILABLE) command to get a list of the tapes marked unavailable.
- Run the relabel utility that is provided by your tape management system, such as EDGINERS from DFSMSrmm. If no tape management system is available, use the IEHINITT utility.

- Issue DELVOL *volser* DUMP(PURGE) for each volume.

Dumping DFSMSHsm-managed volumes

Before beginning this phase, each host obtains from SMS a list of the volumes in copy pools and storage groups that have the AUTODUMP attribute = Y. It also obtains a list of the dump classes to which the volumes in each copy pool and storage group are to be dumped. DFSMSHsm then starts automatic dump tasks in each host, until each host has as many dump tasks as are specified by the MAXDUMPTASKS parameter of the SETSYS command. Volumes in copy pools are processed first.

The following process is followed first for volumes with dumps that can be stacked, then for the remaining volumes: If the maximum number of dump tasks is reached and there are still more volumes to process, DFSMSHsm waits for any already-active volume dump task to complete dumping its current volume. When a volume dump task completes dumping its current volume, it begins dumping the next DFSMSHsm-managed volume with the automatic dump attribute. This process continues until all DFSMSHsm-managed volumes have been dumped, or the quiesce time is passed.

When processing each volume, DFSMSHsm checks the FREQUENCY and DAY subparameters specified in the dump classes to which the volume is assigned. If the DAY corresponds with the day of the dump cycle, and the time since the volume was last dumped is not less than the FREQUENCY specified, the volume is dumped to that dump class. The determination of frequency of dumps includes dumps made by command as well as automatic dumps.

As an example of the effect of the DAY and FREQUENCY subparameters, it has been stated that you are beginning this description of processing for day 28 in a specific dump cycle. Consider volume SG2002 in [Figure 2 on page 17](#). Storage Group 2 specifies that the volumes be dumped to dump classes ONEWEEK and TWOWEEK. Now assume that someone had issued a command to dump SG2002 to dump class ONEWEEK on day 16 of the dump cycle. SG2002 would not have been dumped on day 21, as was specified by ONEWEEK for that day, because less than seven days had elapsed since it was last dumped. It is now day 28. Dump class ONEWEEK specifies that you can dump the volume to it every seven days, so that SG2002 is dumped to dump class ONEWEEK.

Now consider volume SG4001. Storage Group 4 is dumped to dump classes ONEWEEK and FOURWEEK. Dump class FOURWEEK specifies that volumes can be dumped to it only on day 28 in the dump cycle. Therefore, SG4001 is dumped to dump class FOURWEEK because it is day 28 in the dump cycle.

If the maximum number of dump generations has been reached and another dump is to be performed, the oldest generation is deleted. If the RETENTIONPERIOD has the attribute of NOLIMIT, a dump copy is automatically deleted only when it is a member of a dump generation that exceeds the maximum of 100.

Copy pool dump processing is given priority in this phase. Dump processing proceeds as follows:

- Copy pool volumes with affinity to the processing host that are to be stacked
- Copy pool volumes with no affinity that are to be stacked
- Non-copy pool volumes with affinity to the processing host that are to be stacked
- Non-copy pool volumes with no affinity that are to be stacked
- Copy pool volumes with affinity that are not to be stacked
- Copy pool volumes with no affinity that are not to be stacked
- Non-copy pool volumes with affinity that are not to be stacked
- Non-copy pool volumes with no affinity that are not to be stacked
- Non-SMS volumes

During dump processing, if a dump class is not eligible because of FREQUENCY, it is not dumped regardless of its REQUIRED/OPTIONAL status. When a copy pool dump is being resumed, FREQUENCY is *not* considered. When a dump class is not added to a copy pool version because it has not met the FREQUENCY criteria, it can be added outside of the automatic dump window with the FRBACKUP DUMPONLY command

Automatic dump attempts to dump every volume in a copy pool. If any of the volumes fail, the dump version is marked as a partial success. If automatic dump terminates early, it issues message ARC1818I indicating '*n* volumes of copy pool *cpname* did not complete the dump process.' This message is issued once for each copy pool that has not completed automatic dump processing. Automatic dump continues to check for the end of the automatic dump window between each volume.

Since a copy pool is composed of one or more storage groups each having their own Auto Dump Y|N settings, it is suggested that these individual storage groups all have Auto Dump=N defined, unless there is a specific need for dumping volumes at both the copy pool and the storage group levels.

Volume serialization

DFSMSHsm always allocates the source and dump volumes for the full-volume dump before it invokes DFSMSdss. DFSMSdss performs only volume level serialization during full-volume dump processing. Because DFSMSHsm does not perform any data set serialization during full-volume dump processing, activity against a volume being dumped should be kept at an absolute minimum during the full-volume dump process, or errors will occur.

Related reading

For more information about DFSMSdss full-volume dump, see [z/OS DFSMSdss Storage Administration](#).

Names of dump copy data sets

When DFSMSHsm initially allocates a dump volume for output during a volume dump operation, it generates a unique data set name and supplies this name on the dynamic allocation request. (If dump copies are being stacked on a dump volume, each dump copy is in its own file and has its own unique data set name. For each dump copy, DFSMSHsm stores the file sequence number and beginning file block ID in the BCDS records for use during restore.) Because DFSMSdss writes a full-volume dump copy as one data set, this data set name is the same for all tape volumes that are a part of the same dump copy. This name is supplied by DFSMSHsm on the dynamic allocation of a dump copy during a restore operation.

The format of the dump copy data set name is:

```
prefix.DMP.dclass.Vvolser.Dyyddd.Tssmmhh
```

where:

prefix

Prefix you specify with the SETSYS BACKUPPREFIX command.

DMP

V

D

T

Constants.

dclass

Dump class name.

volser

Source volume serial number.

yy

Last two digits of the year of the dump date stamp.

ddd

Julian day of the dump date stamp.

ssmmhh

Time in seconds, minutes, and hours.

Related reading

For more information about how the DFSMSHsm unique data set name is used, see [“Determining the dump copy data set name” on page 458](#).

Write dump VTOC copy data set

If you request it, DFSMSHsm makes a dump VTOC copy data set for each dump generation that is made for a volume. When DFSMSHsm makes a dump VTOC copy data set, it also catalogs the data set. Only a single dump VTOC copy is made for each generation, no matter how many copies are in the generation.

The dump VTOC copy data sets are physical sequential data sets written on ML1 volumes. They contain an entry for each data set VTOC entry that is in the source volume's VTOC when the volume is dumped. These data sets are created just before DFSMSHsm invokes DFSMSdss. VTOC copy data sets are not made when DFSMSHsm dumps a DFSMSHsm-owned volume.

The data set name of the dump VTOC copy is:

```
prefix.DUMPVTOC.Tssmmhh.Vvolser.Dydddd
```

where:

prefix

Prefix you specified with the BACKUPPREFIX parameter of the SETSYS command.

DUMPVTOC

T

V

D

Constants.

ssmmhh

Time in seconds, minutes, and hours.

volser

Source volume serial number.

yyddd

Year and day.

With the following sample job, you can selectively copy all dump VTOC copy data sets to tape. These tapes can be taken off site as a group for use during the recovery and restore process.

```
//DFDSSDUMP JOB 'ACCOUNT #' , 'NAME' , MSGLEVEL=(1,1)
//          EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=A
//TAPE2    DD UNIT=3490 , VOL=SER=DPVTOC , LABEL=( , BLP)
//SYSIN     DD *
            DUMP DATASET (INCLUDE(prefix.DUMPVTOC.**))
            OUTDDNAME (TAPE2)
/*
```

Selecting tape dump volumes

For information about selecting tape dump volumes, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Deleting and uncataloging excess dump VTOC copy data sets

When a volume dump of a level 0 volume is directed to a dump class that has a VTOCCOPIES value greater than zero, a VTOC copy data set is created. Only one VTOC copy data set is created for each dump generation. If more dump copies of a given volume associated with the same dump class are kept than the specified limit of VTOCCOPIES, the VTOC copies created for the older dumps are excess. If the VTOC copy is in excess for all other dump copies in the same generation, the VTOC copy is deleted and uncataloged during this phase of automatic dump.

Automatic backup processing

Assume that it is 3 a.m. in the morning. Because this is a Monday, it is a Y day in the backup cycle. The combination of the Y day in the backup cycle, the BACKUP parameter of the SETSYS command, and the AUTOBACKUPSTART parameter of the SETSYS command cause automatic backup to start in host 1.

Automatic backup is performed in four phases:

1. Backing up the control data sets
2. Moving backup versions
3. Backing up migrated data sets
4. Backing up DFSMSHsm-managed volumes

In multiple DFSMSHsm-host configurations, only the primary host performs the first three phases. The first three phases are performed as successive phases of a single task.

If the primary host fails in a multiple DFSMSHsm-host environment where the secondary host promotion function is being used, the promoted host can take over the first three phases for the failed primary host.

At the beginning of backup in each of the hosts, DFSMSHsm requests a list of all defined management classes. If SMS does not return the list, DFSMSHsm considers SMS inactive and issues a message that backup cannot be performed for SMS-managed volumes. If SMS is active, DFSMSHsm obtains a list of the volumes associated with storage groups that the host is allowed to process, and that have the following characteristics: the AUTO BACKUP ==> Y attribute, are type POOL, and have a status of ENABLED, QUIESCED NEW, or DISABLED NEW. Volumes in storage groups that have a host specified by the BACKUP SYSTEM/SYSPLEX NAME have a host affinity. If SMS is not installed and active, DFSMSHsm processes only non-SMS-managed DASD volumes. For purposes of the illustration, it is assumed that SMS is installed and active.

DFSMSHsm selects the following order of processing as a method of controlling input path contention, subject to prioritization for SMS volumes restricted to processing by one host:

1. Affinity SMS volumes
2. Retry of in-use affinity SMS volumes
3. Non-SMS and nonaffinity SMS volumes

After DFSMSHsm has attempted to process all candidate volumes eligible for automatic backup, DFSMSHsm retries the selection of any of these volumes found in-use by another DFSMSHsm function.

Related reading

- For more information about secondary host promotion, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about retrieving a list of management classes and a list exception, see [“DFSMSHsm retrieval of management classes” on page 464](#).

Phase 1: Backing up the control data sets

The first phase of automatic backup is the backing up of the DFSMSHsm control data sets (CDS). This process is called CDS version backup, and it is controlled by the CDSVERSIONBACKUP parameter of the SETSYS command.

Before backing up the control data sets and the journal data set, DFSMSHsm issues an exclusive enqueue against the control data sets. In a multiple DFSMSHsm-host environment, if the control data sets are *not* being accessed with RLS, DFSMSHsm reserves the volumes containing the journal and the control data sets. If the control data sets *are* being accessed using RLS, DFSMSHsm does an exclusive enqueue. If XCF is available in the multi-host environment, the CDS backup host will notify other hosts through XCF that the enqueue is needed.

The use of the concurrent copy function will reduce the duration of complex-wide serialization of the DFSMSHsm CDS resources when performing CDS version backup and, when SETSYS JOURNAL(RECOVERY) is specified, non-intrusive journal backup is used to further reduce the duration (see [“Using non-intrusive journal backup”](#) on page 496). This time reduction increases the availability of DFSMSHsm functions and non-intrusive journal backup allows DFSMSHsm activity to continue during the journal backup. To achieve the highest level of DFSMSHsm availability, you should place the control data sets on SMS-managed devices that support the concurrent copy function, and you should use DFSMSdss to back up the control data sets. If DFSMSdss backs up to tape, each CDS (or each cluster if multi-cluster control data sets are used) is copied to an individual tape drive (in parallel). Use of concurrent copy is required in order to use non-intrusive journal backup.

If CDS backup is invoked while the control data sets are accessed in RLS mode, DFSMSdss must be the datamover. If the backup is directed to tape, the PARALLEL parameter of the SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY) must be specified. Automatic backup overrides the settings if they are not correct and issues error message ARC0793I. CDS backup fails.

Using the CDSVERSIONBACKUP parameter, you can specify how many versions of the CDS backups be kept active. Each time CDS version backup runs, DFSMSHsm assigns a new version number to the newly-created backup data sets and uses that version number as the final qualifier of the CDS backup data set names.

When backing up control data sets to tape, DFSMSHsm always creates and catalogs the new version and expires the oldest version, so that only the latest versions are available. When backing up the control data sets to DASD, the oldest existing version is reused to record the new backup, and when the backup is complete, that oldest version is renamed and recataloged as the newest version.

DFSMSHsm optionally invokes the CDS backup installation exit (ARCCBEXT) that allows you to access or validate the backup versions after DFSMSHsm creates them.

When the CDS backup is complete, DFSMSHsm releases the volumes containing the journal data set and each control data set, and removes the exclusive enqueue against the control data sets, allowing other DFSMSHsm functions to proceed.

Related reading

- For more information about non-intrusive journal backup, see [“Using non-intrusive journal backup”](#) on page 496.
- For more information about the ARCCBEXT installation exit, see [z/OS DFSMS Installation Exits](#).

Phase 2: Moving backup versions

Before moving backup versions, DFSMSHsm checks to see if any internal TAPECOPY processing is needed for backup volumes, and if so, schedules internal TAPECOPY MWEs. TAPECOPY processing could be needed because of duplexing errors.

Backup versions that are created by data set backup commands may be stored temporarily on ML1 volumes. After DFSMSHsm backs up the control data sets, the primary host moves any backup versions created by data set backup commands that are temporarily residing on ML1 volumes to daily backup volumes.

The primary host is the only host that moves backup versions. Backup versions are moved only once a day and normally during automatic backup. However, if automatic backup is not scheduled to run on a particular day, you can use the ML1BACKUPVERSIONS parameter of the FREEVOL command to move backup versions. Automatic backup is not scheduled to run if you have specified SETSYS AUTOBACKUPSTART(0) or have specified N for this day in the backup cycle.

Phase 3: Backing up migrated data sets

A migrated data set might not have a backup version because it has migrated after it was created or changed but before it was backed up. Such a condition could occur if a MIGRATE command were issued

for the data set shortly after it was created or changed. The migration could be to either an ML1 volume or a DASD ML2 volume.

A data set needs to be backed up when all of the following apply:

- The data-set-changed indicator is on.
- The value of the AUTO BACKUP attribute in the management class is Y.
- The value of the ADMIN OR USER COMMAND BACKUP attribute in the management class is not NONE.
- The value of the AUTO BACKUP attribute in the storage group is Y.

If DFSMSHsm recalls the data set before it is backed up, DFSMSHsm does not back up the data set from the migration volume. Instead, DFSMSHsm backs up the data set when it backs up the DFSMSHsm-managed volume that now contains the data set.

If you specify the BACKDS command to back up a migrated data set, DFSMSHsm does not back up the data set again during this phase.

The primary host is the only host that backs up migrated data sets. Migrated data sets are backed up once a day and normally during automatic backup. However, if automatic backup is not scheduled to run on a particular day, you can use the BACKUP parameter of the RELEASE command to back up migrated data sets.

In backing up migrated data sets, DFSMSHsm reads the required records in the BCDS. As each record is read, DFSMSHsm passes control to the data set backup installation exit (ARCBDEXT) if the installation exit is enabled. If ARCBDEXT does not prevent backup, DFSMSHsm checks (1) the BACKUP FREQUENCY attribute from the management class associated with the data set when the data set migrated, and (2) the recorded last backup date. If the number of days since the last backup date equals or exceeds the BACKUP FREQUENCY attribute value, DFSMSHsm backs up the data set. If DFSMSHsm cannot determine a last backup date and no backup version exists, it backs up the data set. If the number of days since the last backup date is less than the BACKUP FREQUENCY attribute value, DFSMSHsm retains the information in the BCDS and does not back up the data set. If the exit returns a RETAINDAYS value for the data set, it is stored in the MCC and the MCB records.

Related reading

For more information about the ARCBDEXT installation exit, see *z/OS DFSMS Installation Exits*.

Phase 4: Backing up SMS-managed volumes from storage groups with the automatic backup attribute

Before backup processing begins for SMS-managed volumes, DFSMSHsm requests from SMS a list of management classes defined to the configuration in which DFSMSHsm is running. Successful return of the list of management classes indicates that SMS is installed and active. If DFSMSHsm does not receive the list of management classes, it issues a message to indicate that backup cannot be performed for SMS-managed volumes.

If SMS is active, DFSMSHsm retrieves the list of volumes associated with storage groups that have the AUTO BACKUP attribute value of Y. After it retrieves the list of volumes, DFSMSHsm starts the backup tasks up to the maximum specified by the MAXBACKUPTASKS parameter of the SETSYS command in the particular host. If the maximum number of backup tasks is reached and still more volumes must be processed, DFSMSHsm waits for any already-active volume backup task to complete backing up its current volume. When a volume backup task completes backing up its current volume, it begins backing up the next SMS-managed volume with the automatic backup attribute. This process continues until all SMS-managed volumes have been backed up or the quiesce time has passed.

Backup processing examines each data set on the volume individually. Backup processing for each data set:

- Checks for unmatched data sets and issues a message
- Deletes temporary data sets

- Creates a VTOC copy data set entry for the volume
- Backs up the eligible data sets
- Resets the data-set-changed indicator and the last backup date

Processing unmatched data sets

If the catalog entry, the data set VTOC entry, or both, are missing, the data set is an unmatched data set. For an unmatched data set, the space management and backup exit (ARCSAEXT) is invoked if the exit is enabled. If the exit indicates that further eligibility testing be performed for this data set, DFSMSHsm issues a message that this data set is unmatched. If the exit indicates that further eligibility testing not be performed, DFSMSHsm issues no message. In either case, processing continues on the next data set.

Deleting temporary data sets

If for some reason (such as a system outage) SMS-managed temporary data sets are not deleted at end of job, DFSMSHsm attempts to delete them during backup. In a JES3 environment, an SMS-managed temporary data set must be two days old before it can be deleted. These data sets must be as old or older than the integrity age, which is normally zero days, but can be one or two days.

Related reading

For more information about data set integrity age, see [“Integrity age for space management”](#) on page 56.

Create a VTOC copy data set entry for the volume

As DFSMSHsm examines each data set, it creates a VTOC copy data set entry from the information contained in the data set’s VTOC entry. The VTOC copy data set is used in recovering a complete volume. The VTOC copy data set is stored on an ML1 volume.

For SMS-managed volumes, DFSMSHsm records all the data sets on the volume except unmatched data sets and temporary data sets.

DFSMSHsm keeps the two most recent versions of the VTOC copy data sets for the volumes it backs up. If an excess VTOC copy results when DFSMSHsm backs up a volume, it deletes the oldest VTOC copy data set.

The VTOC copy data set is used to guide volume recovery. When a VTOC copy data set is written, DFSMSHsm catalogs the data set.

For each volume that is backed up, DFSMSHsm generates a unique name for the backup VTOC copy data set. The data set name of the VTOC copy is:

```
prefix.VTOC.Tssmmhh.Vvolser.Dyyddd
```

where:

prefix

Prefix you specified with the BACKUPPREFIX parameter of the SETSYS command, if entered.
Otherwise, the DFSMSHsm default UID.

VTOC

T
V
D

Constants.

ssmmhh

Time in seconds, minutes, and hours.

volser

Source volume serial number.

yyddd

Year and day.

With the following sample job, you can selectively copy all backup VTOC copy data sets to tape. These tapes can be taken off site as a group for use during the recovery/restore process.

```
//DFDSSDMP  JOB 'ACCOUT #', 'NAME', MSGLEVEL=(1,1)
//          EXEC PGM=ADRDSSU
//SYSPRINT  DD SYSOUT=A
//TAPE1     DD UNIT=3490, VOL=SER=BUVTOC, LABEL=(, BLP)
//SYSIN     DD *
            DUMP DATASET (INCLUDE(prefix.VTOC.**)) OUTDDNAME (TAPE1)
/*
```

Backing up eligible data sets

DFSMSHsm loops through the entries for the data sets on the volume and backs up all those data sets that are eligible. An eligible data set is one that:

- Has the management class ADMIN OR USER COMMAND BACKUP attribute value of ADMIN or BOTH
- Has the management class AUTO BACKUP attribute value of Y
- Has eligibility, as determined by the data-set-changed indicator, as follows:
 - If the data-set-changed indicator is off and SETSYS INCREMENTALBACKUP(ORIGINAL) is specified, DFSMSHsm backs up the data set if no prior backup version exists.
 - If the data-set-changed indicator is on and DFSMSHsm can determine a last backup date, DFSMSHsm backs up the data set if the number of days since the last backup date equals or exceeds the value specified with the management class BACKUP FREQUENCY attribute.
 - If the data-set-changed indicator is on but DFSMSHsm cannot determine a last backup date (indicating a new data set), DFSMSHsm backs up the data set.
 - If the data-set-changed indicator is on and the data set resides on a volume in a storage group that is defined with the GUARANTEED BACKUP FREQUENCY attribute, DFSMSHsm backs up the data set if the number of days since the last backup date equals or exceeds the value that is specified by GUARANTEED BACKUP FREQUENCY days.

If the data set is eligible for backup and the exit is enabled, DFSMSHsm passes control to the data set backup exit (ARCBDEXT).

For those data sets that are backed up, DFSMSHsm invalidates the oldest excess versions that exceed the limit specified with the NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) management class attribute.

Related reading

- For more information about the ARCBDEXT installation exit, see [z/OS DFSMS Installation Exits](#).
- For more information about data mover support for DFSMSHsm availability management for backup data sets, see the table about availability management and data mover support in [z/OS DFSMSHsm Implementation and Customization Guide](#).

Resetting the data-set-changed indicator and last backup date

After each data set is backed up, if the backup was serialized, the data-set-changed indicator is turned off in the data set's VTOC entry on the source volume. The data-set-changed indicator is also turned off for data sets that are not serialized, if DFSMSHsm has been configured not to serialize on the first attempt to back up.

Note: For non-VSAM multivolume data sets, the data-set-changed indicator is reset on each volume that the data set is cataloged to.

No serialization is an option of the ARCHBACK user macro and the ARCBDEXT installation exit. However, the ARCINBAK inline backup program always specifies this option, because it expects the JCL to perform

its own serialization. In addition, the current date is written in the catalog entry and in the last backup date field in its own MCB record.

Backing up VSAM data sets

DFSMSHsm backup functions that exist for non-VSAM data sets also exist for VSAM data sets cataloged in the integrated catalog facility catalog. DFSMSHsm backs up VSAM data sets on level 0 volumes by using the Access Method Services EXPORT command or DFSMSdss as a data mover.

The backup of a VSAM data set is performed only on a complete VSAM sphere. The base cluster and all alternate index clusters defined on the base cluster are backed up as part of the data set and are not specified separately. Paths are included automatically in the data set backup.

Note: When a VSAM compressed data set is backed up, DFSMSHsm suspends any software compaction requested and records in the BCDS that the data set is striped, compressed, or both.

DFSMSHsm can back up multiple-volume, SMS-managed, VSAM data sets when it processes the first volume of the base data component. Normally, DFSMSHsm prevents update access to any VSAM data set that is currently being backed up. This is done to prevent possible inconsistency of the backup copy.

Tip: You can override this protection for VSAM data sets with the SETSYS INUSE RETRY command or the ARCBDEX exit.

Some installations, however, would prefer that their jobs not fail when they attempt to open a VSAM data set that is currently being backed up. There are two methods to accomplish this, one which includes the risk of having a corrupted backup copy, and one which excludes that risk:

- One method is to use the DFSMSHsm data set backup installation exit, ARCBDEX. This exit can be coded to change the way DFSMSHsm performs serialization on VSAM data sets, allowing batch jobs to open the data sets while the data sets are being backed up. This technique, however, exposes installations to the possibility of creating unserialized backup copies.
- The other method is to add an IEFBR14 step to any batch job that accesses any VSAM data set for updating. The only function of the added step is to access the data set with a DISP of MOD or OLD. This step should be placed immediately before the step that accesses the data set for updating. The IEFBR14 step attempts to get an exclusive ENQ on the data set, which causes the job to wait until DFSMSHsm has finished processing the data set before the subsequent step accesses the data set. This method may require more time to implement than the ARCBDEX solution, but it does eliminate the possibility of having corrupted backup copies.

Related reading

- For more information about the ARCBDEX installation exit, see [z/OS DFSMS Installation Exits](#).
- For more information and an example of how to use the ARCBDEX installation exit in the SYS1.SAMPLIB file, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Generating names of backup versions of data sets

For each data set that is backed up, DFSMSHsm generates a name for the backup version. The name also applies to the name of the backup version of the integrated catalog facility catalog and to DFSMSHsm backup profiles. The name is in the form:

```
prefix.BACK.Tssmmhh.user1.user2.Xydd
```

If patch-enabled, the following dataset name will be generated:

```
prefix.BACK.Tcccchh.user1.user2.Xydd
```

where:

prefix

Prefix you specify with the SETSYS BACKUPPREFIX command.

BACK**T**

Constants.

Note: If two data sets with the same first two high-level qualifiers are backed up at the same second on the same day, DFSMSHsm assigns a letter other than T to one of these data sets.

ssmmhh

Time in seconds, minutes, and hours.

If the format is Tcccchh, then cccchh is the time in hundredths of seconds from the beginning of the hour converted to four alphabetic characters, and hh is hours.

The four characters cccc may be converted to time as follows:

1. Create a variable for each character of the string cccc, using *p1* through *p4*, from left to right. Then assign a value 0 thru 24 for each character A thru Y in the sequence. (Note that 25 characters are used, for A through Y, but not Z.)
2. Find the total hundredths of seconds from the beginning of the hour using this formula:

```
hunds_seconds_begin_hour = p1+(p2*25)+(p3*625)+(p4*625*25)
```

3. Find the minutes, seconds, and hundredths using these formulae (% = integer division):

```
minutes = hunds_seconds_begin_hour%(60*100)
seconds = (hunds_seconds_begin_hour-minutes*60*100)%100
hundredths = hunds_seconds_begin_hour-minutes*60*100-seconds*100
```

cccchh

The time in hundredths of seconds from the beginning of the hour converted to four alphabetic characters (cccc) and hours (hh).

The four characters cccc may be converted to time as follows:

1. Create a variable for each character of the string cccc. Assume these variables are *p1* thru *p4*, from left to right. Assign a value 0 thru 24 for each character A thru Y, in sequence (25 characters are used A - Y, but not Z).
2. Find the total seconds from the beginning of the hour:

```
hunds_seconds_begin_hour = p1+(p2*25)+(p3*625)+(p4*625*25)
```

3. Find minutes, seconds, and hundredths (% = integer division):

```
minutes = hunds_seconds_begin_hour%(60*100)
seconds = (hunds_seconds_begin_hour-minutes*60*100)%100
hundredths = hunds_seconds_begin_hour-minutes*60*100-seconds*100
```

For more information, refer to [Changing DFSMSHsm backup and migration generated data set names to reduce contention for similar names and eliminating a possible performance degradation in z/OS DFSMSHsm Implementation and Customization Guide](#).

user1**user2**

First two qualifiers of the data set name.

Xydd

Decade, year, and day of the year. Valid character values for X are:

G

1970

H

1980

I

1990

J	2000
A	2010
B	2020
C	2030
D	2040
E	2050
F	2060

For example, J3032 represents February 1, 2003.

If you need to access a single-file format tape, you must specify a valid data set name for the single file that comprises up to 254 tape volumes. The backed up data set name for each set of cartridge-type tape volumes in single-file format is:

```
prefix.BACKTAPE.DATASET
```

The tape data set name for each copy of a cartridge-type tape volume created by the TAPECOPY command or the duplex tape option is:

```
prefix.COPY.BACKTAPE.DATASET
```

Only the last 17 characters (.BACKTAPE.DATASET) are recorded in the tape label. To access the single-file format data set, you must supply any valid first qualifier along with the .BACKTAPE.DATASET qualifiers.

Backing up RACF discrete profiles for cataloged data sets

When DFSMSHsm backs up a cataloged RACF-indicated data set, it optionally creates a RACF profile, called the backup profile. The original data set's profile is a model for the backup profile, but the volume serial number is MIGRAT, which means the backup profile does not correspond to an existing data set. This eliminates the need for further updates to the backup profile when DFSMSHsm moves the backup versions from an ML1 volume or backup volume to a daily or spill backup volume. After the backup profile is created, it does not have to be changed. DFSMSHsm uses the backup profile for authorization checking during recovery if the discrete profile for the original data set no longer exists or if the authorization check against the original profile fails.

DFSMSHsm also uses the backup profile for authorization checking during HALTERDS and HBDELETE processing if the authorization check against the original profile fails.

You use the PROFILEBACKUP or NOPROFILEBACKUP parameter of the SETSYS command to specify whether DFSMSHsm creates backup profiles. If you specify PROFILEBACKUP, the backup prefix defined to DFSMSHsm or the default backup prefix of HSM must be a user ID or group defined to RACF.

DFSMSHsm maintains only one backup profile for each cataloged, RACF-indicated, VSAM or non-VSAM data set, no matter how many backup versions of the data set are maintained. The backup profile is always associated with the latest backup version of a data set that was RACF-indicated and cataloged at the time of backup. If you delete the latest backup version of a cataloged, RACF-indicated data set through the BDELETE or HBDELETE command, the backup profile is transferred to the next latest backup version of the data set that was made from a RACF-indicated, cataloged data set. DFSMSHsm does not create backup profiles when it backs up uncataloged data sets.

If the DFSMSHsm-started procedure is not associated with a RACF user ID, the user and group names in the backup profile will be “*”.

DFSMSHsm backs up cataloged, RACF-indicated data sets even if a backup profile is not created or transferred.

Profile backup for migrated data sets

If a migrated data set is being backed up, DFSMSHsm uses the RACF profile of the data set as a model for creating the backup profile.

Deleting backup profiles

When DFSMSHsm creates a new backup profile for a data set, it deletes all previously created backup profiles for the data set.

When backup versions of cataloged, RACF-indicated data sets become obsolete, DFSMSHsm deletes their associated backup profiles even if you specified the NOPROFILEBACKUP parameter of the SETSYS command after the backup profiles were created.

Note: Depending on the release of RACF you are using, you can distribute the location of backed up RACF profiles throughout the RACF data set.

Related reading

For more information about distributing the location of backed up RACF profiles, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Parameters used in creating backup profiles

DFSMSHsm's access control environment element (ACEE) is used when backup profiles are created. If the user ID associated with the DFSMSHsm startup procedure has profile modeling active or is in a group for which profile modeling is active, any access list associated with the model profile is added to the backup profiles. If a profile is recovered that uses the backup profile as a model, the additional user IDs in the access list are attached to the recovered profile.

Selecting daily backup volumes

When DFSMSHsm selects a backup volume, it considers whether this is initial volume selection or subsequent volume selection.

If an error occurs during the selection process for a tape and the error is associated with a particular volume, DFSMSHsm considers the tape backup volume unavailable for backup processing and selects another backup volume. Each time DFSMSHsm is started, all nonfull backup volumes are considered available for backup processing.

The error handling and tape selection process differs when using the duplex tape option for your backup tapes.

Unless they are superseded by the BACKUPDEVICECATEGORY parameter of the BACKVOL or ADDVOL commands, the subparameters of the BACKUP parameter of the SETSYS command control the type of backup volume selected for output during backup processing:

- SETSYS BACKUP(ANY) specifies that any type of backup volume (DASD) or TAPE) defined to DFSMSHsm with the ADVOL command can be selected for output during backup processing. The backup volume has to be nonfull, unallocated, daily backup assigned to the current day in the backup cycle. You must ADDVOL DASD or TAPE volumes to DFSMSHsm for volume selection. If none are defined or available, DFSMSHsm fails the request.
- SETSYS BACKUP(DASD) specifies that only DASD backup volumes are selected for output during backup processing.
- SETSYS BACKUP(TAPE) specifies that only tape backup volumes are selected for output during backup processing. The automatic class selection (ACS) routines determine whether the backup function is directed to a system-managed tape library.

- SETSYS BACKUP(TAPE(*unittype*)) specifies that only tape backup volumes are selected for output during backup processing. The ACS routines determine whether the backup function is directed to a system-managed tape library.

When the backup function is not using a system-managed tape library, UNITTYPE specifies that only tape backup volumes capable of being mounted and written to on the specified type of unit are selected for output during backup processing.

DFSMSHsm backs up users' data sets to daily backup volumes during volume backup. Daily backup volumes either are assigned by the storage administrator to a day in the backup cycle before DFSMSHsm uses them or are assigned when DFSMSHsm selects a volume added as a backup volume.

During backup of a level 0 volume, backup of migrated data sets, or movement of backup versions, DFSMSHsm searches for a target volume as follows:

- It finds a volume already associated with the current day in the backup cycle that is neither full nor in use.
- It reuses a previously full volume.
- It assigns an additional volume.
- It waits for another task to finish using an acceptable volume.

DFSMSHsm stops the search as soon as it finds an acceptable volume. If it does not find an acceptable volume, DFSMSHsm stops processing the requested backup function.

DFSMSHsm determines which daily backup volume to use by considering:

- The status of volumes previously used on this day in the backup cycle. This includes whether this is initial volume selection or subsequent volume selection.
- The backup attributes of the volume being backed up.
- The directions the storage administrator gives to DFSMSHsm.

If an error occurs during the selection process and the error is associated with a particular volume, DFSMSHsm considers the volume unavailable for backup processing and selects another daily backup volume. Each time DFSMSHsm is started, the target volumes previously considered unavailable for backup processing are reinstated as available volumes.

When DFSMSHsm initially selects a daily backup volume, it tries to find one that meets the following characteristics:

- The volume is already assigned to the current day in the backup cycle.
- The volume has the appropriate backup device category.
- The volume is not full.
- The volume is not being used by another task in this or another host.

After a volume backup task has obtained a daily backup volume, it continues to use that volume until the volume is no longer needed, the volume becomes full, or certain error conditions occur.

Now consider the processing that occurs when you specify the backup device categories of DASD, TAPE, or NONE.

Related reading

- For more information about the duplex tape option, see [“Duplex tape option” on page 436](#).
- For more information about how to assign ACS routines and their affect on tape backup selection, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about tape backup selection, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For detailed information on how DFSMSHsm selects volumes when SETSYS BACKUP (ANY) is specified and what happens if no daily backup volumes are defined or available, see [“Selecting DASD and tape daily backup volumes” on page 212](#).

Selecting DASD daily backup volumes

For SMS-managed volumes, you specify a backup device category for the entire system with the BACKUP parameter of the SETSYS command. Suppose that you specified SETSYS BACKUP(DASD). In this environment, DFSMSHsm uses DASD unless a command backup for a volume specifies another device category. DFSMSHsm uses DASD daily backup volumes even if the storage administrator has specified the ADDVOL command to add tape daily backup volumes.

Selection of DASD daily backup volumes depends on the daily limit you specify for the backup cycle. The daily limit determines at what point volume backup stops assigning more DASD daily backup volumes to this day in the backup cycle and instead tries to reuse the full DASD daily backup volumes already assigned to this day in the backup cycle. The daily limit has little or no value when backup output is to tape.

Until the daily limit is reached, DFSMSHsm attempts to associate another backup volume with the day in the cycle. DFSMSHsm then reuses as many of these daily backup volumes as necessary during this day in the backup cycle as long as enough space is available on the DASD daily backup volumes. You use the BACKUP parameter of the DEFINE command to define the daily limit.

If you choose a daily limit that is too small, your system will have poorer performance because it spends too much time doing cleanup for the amount of space made usable. The daily limit is too small if cleanup does not make the daily backup volumes reusable.

If you choose a daily limit that is too large, too many volumes can be assigned to a particular day in the backup cycle; therefore, available space cannot be used because it is restricted to certain days in the backup cycle. The daily limit is too large if DFSMSHsm uses only a small percentage of the daily backup volumes assigned to the day in the backup cycle on a day when a normal amount of data is backed up.

To reuse DASD daily backup volumes that were previously full, DFSMSHsm uses spill processing or cleanup processing. You specify spill processing with the SPILL parameter of the SETSYS command; only cleanup processing is done if you specify the NOSPILL parameter.

DFSMSHsm reuses DASD daily backup volumes when the following conditions occur:

- Another volume is needed.
- The number of volumes assigned to this day in the backup cycle has already reached the daily limit or there were no unassigned DASD daily backup volumes to assign to this day in the backup cycle.
- All volumes assigned to this day in the backup cycle are either being used or were marked full after being used earlier during this day in the backup cycle.

If the reuse process has tried today to use all volumes not currently in use, an additional volume is assigned from the set of unassigned DASD backup volumes, even though the number of assigned volumes exceeds the daily limit. If no unassigned DASD backup volume exists but at least one acceptable volume is being used, DFSMSHsm waits for the task to finish using the acceptable volume.

Related reading

- For more information about spill processing, see [“Backup volume spill process” on page 215](#).
- For more information about cleanup processing, see [“Cleanup processing” on page 218](#).

Selecting initial DASD daily backup volumes

The processing that occurs during initial selection of the DASD daily backup volumes depends on whether the daily limit has not been reached or the daily limit has been reached or exceeded.

Daily limit not reached

If all DASD daily backup volumes are full or in use before the daily limit is reached, DFSMSHsm does the following processing:

1. It assigns another DASD daily backup volume. When DFSMSHsm assigns another DASD daily backup volume, it tries to select volumes in the following order:

- a. Unassigned DASD daily backup volume. DFSMSHsm then assigns it to the current day in the backup cycle.
- b. Unassigned DASD backup volume. DFSMSHsm then assigns it as a daily backup volume and to the current day in the backup cycle.
2. It reuses a full DASD daily backup volume. When no unassigned DASD backup volumes are available, DFSMSHsm either spills or cleans up the full, unallocated, DASD daily backup volumes. DFSMSHsm uses the first volume that falls below the threshold of occupancy that you specified with the THRESHOLD parameter of the ADDVOL command.
3. It waits for a DASD daily backup volume that is in use to become available. If the spill or cleanup process does not result in a usable DASD daily backup volume and another volume backup task is currently using a DASD daily backup volume, this volume backup task waits for a DASD daily backup volume to become available.

In a multiple DFSMSHsm-host environment, before a volume backup task enters the wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched, and DFSMSHsm tries to select another DASD daily backup volume. If the DASD daily backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMSHsm tries to select the DASD daily backup volume again.

4. It stops backup processing. If DFSMSHsm cannot find a DASD daily backup volume and no other volume backup task is using a DASD daily backup volume, the current volume backup task ends because no DASD daily backup volume is available.

Daily limit reached or exceeded

If no daily backup volume is available and the number of daily backup volumes assigned to the current day in the backup cycle is greater than or equal to the daily limit, DFSMSHsm does the following processing:

1. It reuses a full DASD daily backup volume. When the daily limit has been reached or exceeded, DFSMSHsm either spills or cleans up the full, unallocated, DASD daily backup volumes. DFSMSHsm uses the first volume that falls below the threshold of occupancy that you specified with the THRESHOLD parameter of the ADDVOL command.
2. It assigns another DASD daily backup volume. If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMSHsm assigns another daily backup volume. A DASD daily backup volume is not usable if all volumes currently assigned to this day in the backup cycle are being used or have been cleaned up today. DFSMSHsm tries to select volumes in the following order:
 - a. Unassigned DASD daily backup volume. DFSMSHsm then assigns it to the current day in the backup cycle.
 - b. Unassigned DASD backup volume. DFSMSHsm then assigns it as a daily backup volume and to the current day in the backup cycle.
3. It waits for a DASD daily backup volume that is in use to become available. If the spill or cleanup process does not result in a usable DASD daily backup volume and another volume backup task is currently using a DASD daily backup volume, this volume backup task waits for a DASD daily backup volume to become available.

In a multiple DFSMSHsm-host environment, before a volume backup task enters the wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched, and DFSMSHsm tries to select another DASD daily backup volume. If the DASD daily backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMSHsm tries to select the DASD daily backup volume again.

4. It stops backup processing. If DFSMSHsm cannot find a DASD daily backup volume and no other volume backup task is using a DASD daily backup volume, the current volume backup task ends because no DASD daily backup volume is available.

Selecting subsequent DASD daily backup volumes

DFSMSHsm uses the same procedures during subsequent DASD daily backup volume selection as are described in [“Selecting initial DASD daily backup volumes”](#) on page 209. However, after a volume backup task has obtained a daily backup volume, it continues to use that volume until the volume is no longer needed, the volume becomes full, or certain error conditions occur.

If a DASD daily backup volume does not contain enough available space for the backup version of the data set DFSMSHsm is backing up or contains enough total available space but the space is too fragmented, DFSMSHsm does not back up the data set. The space is too fragmented if DFSMSHsm cannot allocate the backup version in five or fewer extents. When the allocation of the backup copy of a data set fails because the DASD daily backup volume does not have enough space for the backup version, DFSMSHsm does one of the following:

- If the percent of occupied space on the DASD daily backup volume is less than the specified threshold of occupancy, DFSMSHsm fails backup of the data set. Volume processing continues with the next eligible data set on the source volume.
- If the percent of occupied space on the DASD daily backup volume is greater than or equal to the specified threshold of occupancy, DFSMSHsm considers the DASD daily backup volume to be full. When the current DASD daily backup volume becomes full, DFSMSHsm releases it, selects another daily backup volume, and tries to back up the data set again. If DFSMSHsm cannot back up the data set a second time because the second daily backup volume does not have enough space, DFSMSHsm fails backup of the data set. However, DFSMSHsm does not consider the second DASD daily backup volume to be full. Volume backup processing continues with the next eligible data set on the source volume.

Selecting tape daily backup volumes

In the example system, the SETSYS BACKUP(TAPE) command was used to specify that all backups are to tape. In this environment, DFSMSHsm uses only tape daily backup volumes unless a volume backup command specifies backup to DASD.

If you want DFSMSHsm to use tape as a daily backup volume and you have not specified restrictions limiting selection to tape, you must define at least one tape for each of the appropriate days in the backup cycle by using the ADDVOL command. You do not have to define tapes as daily backup volumes with the ADDVOL command if you have restricted daily backup volumes to tape by:

- Specifying BDC(TAPE) on a BACKVOL command
- Specifying BACKUP(TAPE) on a SETSYS command

If you use cartridge-type tape backup volumes, DFSMSHsm writes data on the tape in a single-file format.

DFSMSHsm considers the following factors when it selects a tape backup volume:

- Whether this is initial or subsequent volume selection
- For subsequent volume selection, the type of volume selection you specified with the SELECTVOLUME parameter of the SETSYS command: specific or scratch.

Related reading

For more information about the selection of tape volumes that are in a system-managed tape library, refer to the [z/OS DFSMSHsm Implementation and Customization Guide](#).

Selecting initial tape daily backup volumes

If you have no partially filled or empty tape daily backup volumes already assigned to the day in the backup cycle, DFSMSHsm does the following processing:

1. DFSMSHsm assigns another tape daily backup volume. When these volumes are assigned, DFSMSHsm tries to select volumes in the following order:
 - a. Unassigned tape daily backup volume. DFSMSHsm then assigns it to the current day in the backup cycle.

- b. Unassigned tape backup volume. DFSMSHsm then assigns it as a daily backup volume and to the current day in the backup cycle.

If no unassigned backup volumes are available, DFSMSHsm requests a scratch tape.

2. DFSMSHsm waits for a tape daily backup volume that is in use to become available. If no scratch tape is available and another volume backup task is currently using a tape daily backup volume, this volume backup task waits for a tape daily backup volume to become available.

In a multiple DFSMSHsm-host environment, before a volume backup task enters the wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched, and DFSMSHsm tries to select another tape daily backup volume. If the tape daily backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMSHsm tries to select the tape daily backup volume again.

3. DFSMSHsm stops backup processing. If DFSMSHsm cannot find a tape daily backup volume and no other volume backup task is using a tape daily backup volume, the current volume backup task ends because no tape daily backup volume is available.

Selecting subsequent tape daily backup volumes

After a volume backup task has obtained a tape daily backup volume, it continues to use that volume until the volume is no longer needed, the volume becomes full, or certain error conditions occur.

If end-of-volume is reached while DFSMSHsm is writing on a tape daily backup volume, DFSMSHsm considers the currently mounted tape daily backup volume to be full and selects another tape volume.

A backup version of a data set can span more than one tape backup volume. DFSMSHsm fails the backup of a data set if the data set spans more than 254 tape backup volumes.

When DFSMSHsm reaches end-of-volume while writing on a tape daily backup volume, the processing that occurs depends on whether you want DFSMSHsm to select a volume from tapes already known to it, or to select a scratch tape volume.

If you specified the `SELECTVOLUME(SPECIFIC)` or `SELECTVOLUME(BACKUP(SPECIFIC))` parameter of the `SETSYS` command, DFSMSHsm selects a tape backup volume that meets the following conditions:

- The volume is added to DFSMSHsm as a tape backup volume. Use the `ADDVOL` command to add the volume to DFSMSHsm.
- The volume is empty.
- The volume is not currently being used.
- The volume has the same device type and density as the previous volume.

If DFSMSHsm cannot find a volume that meets the criteria just specified or if you specified `SETSYS SELECTVOLUME(SCRATCH)` or `SELECTVOLUME(BACKUP(SCRATCH))`, DFSMSHsm requests that a scratch tape be mounted on the already allocated unit. When the operator mounts a scratch tape, DFSMSHsm assigns it as a tape daily backup volume and assigns it to the current day in the backup cycle.

After the operator mounts the volume, DFSMSHsm continues backing up the current data set to that volume.

Certain error conditions, such as an error in writing on the tape, cause DFSMSHsm to switch tape daily backup volumes. If an error occurs, DFSMSHsm performs the following actions:

- Marks the tape daily backup volume as full
- Releases the tape volume and deallocates the unit
- Performs initial selection to select another tape daily backup volume

Selecting DASD and tape daily backup volumes

If you specify `BACKUP(ANY)` on the `SETSYS` command, you must `ADDVOL DASD` or `Tape` daily backup volumes to DFSMSHsm for daily backup selection. DFSMSHsm selects the first non-full, unallocated, daily

backup volume assigned to the current day in the backup cycle, regardless of its device category. If you want DFSMSHsm to use tape as a daily backup volume and you do not specify restrictions limiting selection to tape, you must define at least one tape for each of the appropriate days in the backup cycle by using the ADDVOL command. DFSMSHsm does not select a scratch tape if no daily or unassigned, nonfull backup volumes are found already defined to DFSMSHsm. If DASD daily backup volumes are not being used in your environment, specify SETSYS BACKUP (TAPE) rather than SETSYS BACKUP (ANY). This allows flexibility of backing up to a scratch tape if an eligible daily backup volume is not found. You do not have to define tapes as daily backup volumes with the ADDVOL command if you restrict daily backup volumes to a tape by:

- Specifying BACKUP (TAPE) on a SETSYS command.
- Specifying BACKUPDEVICECATEGORY (TAPE) on a BACKVOL command.
- Specifying BACKUPDEVICECATEGORY (TAPE) on the ADDVOL command of the nonSMS volume being backed up. When daily backup volumes are restricted to tape in one of these ways, DFSMSHsm selects a scratch tape if an eligible daily backup volume is not found.

Note: If you specify BACKUP(ANY) on the SETSYS command, DFSMSHsm selects the first nonfull, unallocated, daily backup volume assigned to the current day in the backup cycle. If you want DFSMSHsm to use tape as a daily backup volume and you do not specify BACKUP(TAPE), you must define at least one tape for each of the appropriate days in the backup cycle by using the ADDVOL command. DFSMSHsm does not select a scratch tape if no daily or unassigned nonfull backup volumes are found already defined to DFSMSHsm. However, if you restrict daily backup volumes to tape by specifying BACKUP(TAPE) on a SETSYS command, you do not have to define tapes as daily backup volumes with the ADDVOL command. DFSMSHsm selects the first nonfull, unallocated, daily backup assigned to the current day in the backup cycle. If none are available, DFSMSHsm selects a scratch tape.

Selecting initial daily backup volumes (DASD or tape)

DFSMSHsm can select either DASD or tape as a daily backup volume. The process that occurs during initial DASD or tape daily backup volume selection depends on whether the daily limit has been reached or exceeded or has not been reached.

Daily limit reached or exceeded

If no daily backup volume is available and the number of daily backup volumes assigned to the current day in the backup cycle is greater than or equal to the daily limit, DFSMSHsm does the following processing:

1. It reuses a full DASD daily backup volume. When no unassigned backup volumes are available, DFSMSHsm either spills or cleans up the full, unallocated, DASD daily backup volumes. If you specified SETSYS SPILL, DFSMSHsm spills the volumes. If you specified SETSYS NOSPILL, DFSMSHsm only cleans up the volumes. DFSMSHsm uses the first volume that falls below the threshold of occupancy (which you specified with the THRESHOLD parameter of the ADDVOL command). A DASD daily backup volume is not usable if all volumes currently assigned to this day in the backup cycle are either being used or are full and have been cleaned up today.

If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMSHsm requests a scratch tape. After a scratch tape is mounted, DFSMSHsm obtains the volume serial number and internally adds the volume as a daily backup volume assigned to the current day in the backup cycle.
2. It assigns another daily backup volume. If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMSHsm assigns another daily backup volume. When DFSMSHsm assigns another daily backup volume, it tries to select volumes in the following order:
 - a. Unassigned daily backup volume. DFSMSHsm then assigns it to the current day in the backup cycle. This daily backup volume can be tape or DASD.
 - b. Unassigned backup volume. DFSMSHsm then assigns it as a daily backup volume and to the current day in the backup cycle. This backup volume can be tape or DASD.

If no unassigned backup volume exists, DFSMSHsm requests a scratch tape. When no unassigned backup volumes are available, DFSMSHsm either spills or cleans up the full, unallocated, DASD daily backup volumes. If you specified SETSYS SPILL, DFSMSHsm spills the volumes. If you specified SETSYS NOSPILL, DFSMSHsm only cleans up the volumes. DFSMSHsm uses the first volume that falls below the threshold of occupancy (which you specified with the THRESHOLD parameter of the ADDVOL command). A DASD daily backup volume is not usable if all volumes currently assigned to this day in the backup cycle are either being used or are full and have been cleaned up today.

If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMSHsm requests a scratch tape. After a scratch tape is mounted, DFSMSHsm obtains the volume serial number and internally adds the volume as a daily backup volume assigned to the current day in the backup cycle.

3. It waits for a daily backup volume that is in use to become available. If DFSMSHsm cannot find a daily backup volume to use and another volume backup task is currently using an acceptable daily backup volume, this volume backup task waits for any type of daily backup volume to become available.
4. It stops backup processing. If DFSMSHsm cannot find the type of daily backup volume it needs and no other volume backup task is using that type of backup volume, the current volume backup task ends because no daily backup volume is available.

Daily limit not reached

If all daily backup volumes are full or in use before the daily limit for daily backup volumes is reached, DFSMSHsm does the following:

1. It assigns another daily backup volume. When DFSMSHsm assigns another daily backup volume, it tries to select volumes in the following order:
 - a. Unassigned daily backup volume. DFSMSHsm then assigns it to the current day in the backup cycle. This daily backup volume can be tape or DASD.
 - b. Unassigned backup volume. DFSMSHsm then assigns it as a daily backup volume and to the current day in the backup cycle. This backup volume can be tape or DASD.
2. It reuses a full DASD daily backup volume or requests a scratch tape. When no unassigned backup volumes are available, DFSMSHsm either spills or cleans up the full, unallocated, DASD daily backup volumes. If you specified SETSYS SPILL, DFSMSHsm spills the volumes. If you specified SETSYS NOSPILL, DFSMSHsm only cleans up the volumes. DFSMSHsm uses the first volume that falls below the threshold of occupancy (which you specified with the THRESHOLD parameter of the ADDVOL command). A DASD daily backup volume is not usable if all volumes currently assigned to this day in the backup cycle are either being used or are full and have been cleaned up today.

If the spill or cleanup process does not result in a usable DASD daily backup volume, DFSMSHsm requests a scratch tape. After a scratch tape is mounted, DFSMSHsm obtains the volume serial number and internally adds the volume as a daily backup volume assigned to the current day in the backup cycle.

3. It waits for a daily backup volume that is in use to become available. If DFSMSHsm cannot find a daily backup volume to use and another volume backup task is currently using an acceptable daily backup volume, this volume backup task waits for any type of daily backup volume to become available.

In a multiple DFSMSHsm-host environment, before a volume backup task enters a wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched and DFSMSHsm tries to select another daily backup volume. If the daily backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMSHsm tries to select the daily backup volume again.

4. It stops backup processing. If DFSMSHsm cannot find the type of daily backup volume it needs and no other volume backup task is using that type of backup volume, the current volume backup task ends because no daily backup volume is available.

Selecting subsequent daily backup volumes (DASD or tape)

DFSMSHsm uses the same procedures during subsequent daily backup volume selection as those described in [“Selecting initial daily backup volumes \(DASD or tape\)”](#) on page 213. However, after a volume backup task has obtained a daily backup volume, it continues to use that volume until the volume is no longer needed or becomes full, or until certain error conditions occur.

Selecting subsequent DASD daily backup volumes

If a DASD daily backup volume does not contain enough available space for the backup version of the data set DFSMSHsm is backing up, or contains enough total available space for the backup version but the space is too fragmented, DFSMSHsm follows the procedures described in [“Selecting subsequent DASD daily backup volumes”](#) on page 211.

If you use both DASD and tape backup volumes, DFSMSHsm can start backing up a volume to a DASD daily backup volume and continue with a tape daily backup volume. However, after DFSMSHsm has selected a tape daily backup volume to back up a volume, it uses only tape backup volumes when additional backup volumes are required for the backup of the current volume.

Selecting subsequent tape daily backup volumes

If end-of-volume is reached while DFSMSHsm is writing on a tape daily backup volume, DFSMSHsm follows the procedures described in [“Selecting subsequent tape daily backup volumes”](#) on page 212.

Backup volume spill process

Moving backup versions from DASD backup volumes to spill volumes is called *spill*. Spill is a process that applies *only* if you are using DASD devices to store your backup versions. If you use DASD as your daily backup volumes, your daily backup volumes eventually become filled with recent backup versions and older backup versions of the same data sets. The DASD daily backup volumes can also contain backup versions of deleted data sets and retired backup versions.

Note: Although both the spill and recycle process output to spill backup tapes, the spill process occurs only from DASD. Moving backup versions from tape backup volumes is part of recycle.

When adding DASD backup volumes to DFSMSHsm with an ADDVOL command, you can specify a maximum percentage of the capacity of the volume above which the volume is considered to be full after spill or cleanup processing. This percentage is called the threshold of occupancy. When a DASD daily backup volume becomes full (an allocation has been failed for lack of space), it is *spilled* by the spill process. If the spill process does not reduce the occupancy of the volume below the threshold of occupancy, the volume is still considered to be full.

The spill process moves all but the latest backup version of a data set from a DASD daily backup volume to a spill backup volume. DFSMSHsm owns the spill backup volume. During daily backup volume selection when all DASD daily backup volumes are full, DFSMSHsm uses the spill process to make space available on the full DASD daily backup volumes. The spill process produces a DASD daily backup volume usable during volume backup processing. The spill process moves the following from daily backup volumes to spill backup volumes:

- All but the latest backup version of each data set
- All retired backup versions (see [“Data set retirement”](#) on page 114)
- All backup versions of a deleted data set

If a backup version is not valid, DFSMSHsm does not move it. Instead, DFSMSHsm scratches the invalid backup version.

DFSMSHsm spills the full, unallocated, DASD daily backup volume that is assigned to the current day in the backup cycle and that has not been spilled in the longest period of time. If the volume spill process is successful, the DASD daily backup volume is used for the backup of the DFSMSHsm-managed volumes.

Otherwise, DFSMSHsm spills the full, unallocated, DASD daily backup volume that has not been spilled in the next longest period of time. This process is repeated until a usable DASD daily backup volume is produced or until every full, unallocated, DASD daily backup volume assigned to the current day in the backup cycle has been spilled on that day.

If cataloged and uncataloged data sets have the same name and each has backup versions, DFSMSHsm does not move either the latest backup version of the cataloged data set or the latest backup version of the uncataloged data set.

When a DASD daily backup volume is spilled, it is also cleaned up. [“Backup volume cleanup process” on page 219](#) contains detailed information about cleanup processing. When the full DASD daily backup volume to be spilled has been allocated, DFSMSHsm:

- Reads the volume table of contents (VTOC) of the DASD daily backup volume. As each data set VTOC entry is encountered, the name is checked to see if it is a backup version created by DFSMSHsm. If it is, a check is made to see if this backup version is the latest backup version for the cataloged or uncataloged data set. If it is not the latest backup version, is a retired version, or is a version of a deleted data set and it is a valid backup version, DFSMSHsm records that the backup version needs to be moved to a spill backup volume. If the data set on the backup volume either was not created by DFSMSHsm or is a VTOC or VCAT copy, it is not processed.
- Selects a spill backup volume if any backup versions exist that should be moved. After DFSMSHsm selects a spill backup volume, it begins to move the backup versions.
- Determines the percentage of allocated space on the daily backup volume after all the backup versions are moved to the spill backup volume. If the percentage of allocated space is less than or equal to the user-specified threshold of occupancy, the daily backup volume is no longer considered full and is used for the backup of the DFSMSHsm-managed volumes. If the percentage of allocated space is not less than or equal to the threshold of occupancy, the daily backup volume is considered to still be full and is not selected as a target for backup versions. You can specify the threshold of occupancy by using the THRESHOLD parameter of the ADDVOL command.

Selecting spill backup volumes

When DFSMSHsm selects a spill backup volume, it considers whether this is the initial spill volume selection or a subsequent spill volume selection.

If an error occurs during the selection process for a tape and the error is associated with a particular volume, DFSMSHsm considers the tape spill backup volume unavailable for spill backup processing and selects another spill backup volume. Each time DFSMSHsm is started, all non-full backup volumes are considered available for backup processing.

The subparameters of the SPILL parameter of the SETSYS command control the type of spill backup volume selected for output during spill processing. DFSMSHsm ignores the device category specified for backup when it performs spill processing:

- SETSYS SPILL(ANY) specifies that any type of spill backup volume can be selected for output during spill processing.
- SETSYS SPILL(DASD) specifies that only DASD spill backup volumes are selected for output during spill processing.
- SETSYS SPILL(TAPE) specifies that only tape spill volumes are selected for output during spill processing. The automatic class selection (ACS) routines determine whether the spill function is directed to a system-managed tape library.
- SETSYS SPILL(TAPE(*unittype*)) specifies that only tape spill volumes are selected for output during spill processing. The ACS routines determine whether the spill function is directed to a system-managed tape library.

When the spill function is not using a system-managed tape library, UNITTYPE specifies that only tape spill volumes capable of being mounted and written to on the specified type of unit are selected for output during spill processing.

Related reading

For more information about spill processing, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Initial spill backup volume selection (DASD and tape)

For each full, DASD daily backup volume being spilled, DFSMSHsm selects a spill backup volume to contain the backup versions to be moved. If you want DFSMSHsm to use tape as spill backup volumes and you have not specified restrictions limiting selection to tape, you must define at least one tape as a spill volume by using the ADDVOL command. You need not define a tape as a spill volume if you restrict spill backup volumes to tape by specifying SPILL(TAPE) on a SETSYS command.

DFSMSHsm selects the first non-full, unallocated spill backup volume to contain the backup versions to be moved. If a spill backup volume is not available, DFSMSHsm selects an unassigned backup volume and assigns it as a spill backup volume.

Unassigned backup volume exists

The type of unassigned backup volume selected depends on the type of volumes already assigned as spill backup volumes:

- If both tape and DASD spill backup volumes are defined, DFSMSHsm selects any type of unassigned backup volume
- If only DASD spill backup volumes are defined, DFSMSHsm selects only a DASD unassigned backup volume
- If only tape spill backup volumes are defined, DFSMSHsm selects only a tape unassigned backup volume

No unassigned backup volume exists

If no unassigned backup volume exists, the actions taken depend on the type of spill backup volumes already defined:

- If both tape and DASD spill backup volumes are defined, DFSMSHsm also uses the cleanup process in an attempt to produce a usable spill backup volume. However, if the cleanup process does not produce a usable spill backup volume, DFSMSHsm requests a scratch tape. If a scratch tape is not available and another volume backup task is using a tape spill backup volume, this volume backup task waits for any spill backup volume to become available.
- If only DASD spill backup volumes are defined, DFSMSHsm uses the cleanup process in an attempt to produce a usable spill backup volume. If the cleanup process is unsuccessful and another volume backup task is using a DASD spill backup volume, this volume backup task waits for a DASD spill backup volume to become available.
- If only tape spill backup volumes are defined, DFSMSHsm requests a scratch tape. If a scratch tape is not available and another volume backup task is using a spill backup volume, this volume backup task waits for a tape spill backup volume to become available.

Wait for a spill backup volume

In a multiple DFSMSHsm-host environment, before a volume backup task enters a wait state, a five-minute timer is set. When the timer expires, the volume backup task is dispatched, and DFSMSHsm tries to select another spill backup volume. If the spill backup volume is still not available, the volume backup task resets the timer and reenters a wait state. If another volume backup task dispatches the volume backup task in the same host before the timer expires, the timer is canceled, and DFSMSHsm tries to select the spill backup volume again.

No spill backup volume available

If no spill backup volume is available and another volume backup task is not using a spill backup volume, this backup volume spill operation ends because no spill backup volumes are available.

Subsequent spill backup volume selection (DASD and tape)

After a volume backup task has obtained a spill backup volume, the task continues to use that volume until the volume becomes full, the spill of the full DASD daily backup volume completes, or certain error conditions occur.

Subsequent DASD spill backup volume selection

If a DASD spill backup volume is being used and does not contain enough available space for the backup version currently being moved or contains enough total available space for the backup version but the space is too fragmented, DFSMSHsm fails the movement of the backup version. DFSMSHsm considers the space too fragmented if the allocation of the backup version cannot be done in five or fewer extents. When the new allocation for the backup version fails because of insufficient space for the backup version, DFSMSHsm performs the procedures described in [“Selecting subsequent DASD daily backup volumes” on page 211](#).

Note: In reading the procedures just referred to, substitute the term “spill” for the term “daily”.

Subsequent tape spill backup volume selection

If end-of-volume is reached while DFSMSHsm is writing on a tape spill backup volume, DFSMSHsm performs the procedures described in [“Selecting subsequent tape daily backup volumes” on page 212](#).

Note: In reading the procedures just referred to, substitute the term “spill” for the term “daily”.

If DFSMSHsm cannot find a tape volume meeting the preceding criteria or if you specified SETSYS SELECTVOLUME(SCRATCH), DFSMSHsm requests that a scratch tape be mounted on the already allocated unit. When the operator mounts a scratch tape, DFSMSHsm assigns it as a tape spill backup volume.

After the operator mounts the volume, DFSMSHsm continues moving backup versions to that volume.

Certain error conditions, such as an error in writing on the tape, cause DFSMSHsm to switch tape spill backup volumes. If an error occurs, DFSMSHsm considers the tape spill backup volume to be full, releases the volume, and selects another tape spill backup volume as described in [“Initial spill backup volume selection \(DASD and tape\)” on page 217](#), but only a tape volume is selected. After DFSMSHsm selects a tape spill backup volume for the spilling of a full DASD daily backup volume, it uses only tape spill backup volumes when additional spill backup volumes are required during the current spilling of the full DASD daily backup volume.

Cleanup processing

Cleanup processing is a subset of spill processing. When DFSMSHsm does spill processing, it also does cleanup processing. DFSMSHsm also does cleanup processing even when spill processing is not enabled.

DFSMSHsm uses the cleanup process to scratch unwanted data sets on the DASD backup volumes. These data sets are:

- Out-of-date VTOC copy data sets created by DFSMSHsm releases before Version 2 Release 3.0 and out-of-date VCAT copy data sets
- Unwanted backup versions

When DFSMSHsm scratches a VTOC copy data set, it also uncatalogs the data set.

During volume backup processing, DFSMSHsm creates a data set (the VTOC copy data set) that contains selected information from the VTOC of the volume that DFSMSHsm is backing up. This data set drives the recovery of the volume. DFSMSHsm maintains two copies of the VTOC copy data set. When DFSMSHsm backs up a volume, the newly created VTOC copy data set becomes the latest VTOC copy data set, and the data set that was previously the latest VTOC copy data set becomes the next latest VTOC copy data set. As a result, these data sets become continually out of date as DFSMSHsm keeps backing up the

volume. If DFSMSHsm encounters any of these out-of-date VTOC or VCAT copy data sets during volume cleanup processing, DFSMSHsm scratches them and uncatalogs the VTOC copy data set.

Unwanted backup versions are those whose backup version record (MCC) has been deleted from the BCDS. The following are processes that can cause these unwanted backup versions:

- New backup versions are created and the MCC record for an old backup version is deleted, but the old version is not scratched.
- Use of the BDELETE command. The MCC record for the old backup version is deleted, but the old version is not scratched.

DFSMSHsm does cleanup processing under three conditions:

- DFSMSHsm selects a DASD daily or spill backup volume.
- No daily backup volume is available. That is, all daily backup volumes assigned to that day in the backup cycle are full, in use, or unavailable.
- No spill backup volume is available. That is, all spill backup volumes are full, in use, or unavailable.

DFSMSHsm selects a DASD backup volume

When DFSMSHsm selects a non-full, unallocated, DASD daily or spill backup volume, DFSMSHsm cleans it up if the volume has not been cleaned up on that day. DFSMSHsm cleans up the volume even if the volume is not full.

No daily backup volume available

If you specified SETSYS NOSPILL, DFSMSHsm cleans up the full, unallocated, DASD daily backup volumes. DFSMSHsm starts with the full, unallocated, DASD daily backup volume that has not been cleaned up in the longest period of time. If the cleaned up DASD daily backup volume is usable, DFSMSHsm backs up the level 0 volume to this DASD daily backup volume. Otherwise, DFSMSHsm cleans up the full, unallocated, DASD daily backup volume that has not been cleaned up in the next longest period of time. This process continues until a usable DASD daily backup volume is produced or until every full, DASD daily backup volume has been cleaned up on that day. DFSMSHsm does not clean up a DASD backup volume more than once a day.

No spill backup volume available

DFSMSHsm cleans up the full, unallocated, DASD spill backup volume that has not been cleaned up in the longest period of time. If the cleaned up DASD spill backup volume is usable, DFSMSHsm spills the full, DASD daily backup volume to this DASD spill backup volume. Otherwise, DFSMSHsm cleans up the full, unallocated, DASD spill backup volume that has not been cleaned up in the next longest period of time. This process continues until a usable DASD spill backup volume is produced or until every full DASD spill backup volume has been cleaned up on that day. DFSMSHsm does not clean up a DASD spill backup volume more than once a day.

Backup volume cleanup process

After the DASD backup volume to be cleaned up is allocated, the following is performed:

1. The VTOC is read.
 - A check is performed to determine if the data set name corresponds to a VTOC or VCAT copy data set.
 - If the data set name corresponds to a VTOC or VCAT copy data set, a check is performed to determine if the data set is a current version on any volume. If the data set is a current version, the data set is scratched.
 - If the data set name does not correspond to a VTOC or VCAT copy data set, a check is performed to determine if the data set name corresponds to a DFSMSHsm created backup version.

- If the data set name corresponds to a DFSMSHsm created backup version, the backup version record (MCC) from the BCDS is read. If no backup version record exists, the backup version is scratched.
 - If the data set name does not correspond to a DFSMSHsm created backup version, the data is not processed.
2. After the VTOC is read and all unnecessary backup versions, VTOC copy data sets and VCAT copy data sets are scratched, and the percentage of allocated space on the DASD backup volume is determined.
- If a full DASD daily backup volume was cleaned up, and the allocated space is less than or equal to the user-specified allocated space threshold, the volume is no longer considered full and is used to back up DFSMSHsm-managed volumes.
 - If a full DASD spill backup volume was cleaned up, and the allocated space is less than or equal to the user-specified allocated space threshold, the volume is no longer considered full and is used to spill a full DASD daily backup volume.

Tip: The allocated space threshold can be specified using the THRESHOLD parameter of the ADDVOL command.

Note: When cleanup is performed, the backup volume record (BVR) for the volume is held with an exclusive enqueue while the cleanup process is performed. This enqueue can cause message ARC0712I to be issued during the cleanup process.

Releasing a backup volume

When a backup volume task finishes using a backup volume, it releases the backup volume by deallocating it. If the backup volume being released is not full and another volume backup task in the same host is waiting for the type of backup volume being released, the other volume backup task is notified that the backup volume is available. DFSMSHsm determines the amount of unallocated space remaining on the volume if the backup volume is a DASD volume. If the released backup volume is a virtual volume and you have specified the UNLOAD parameter of the SETSYS command, DFSMSHsm issues the MVS UNLOAD command to unload this volume.

Automatic backup example

Table 25 on page 220 shows the conditions that pertain to a group of data sets at the beginning of automatic backup. See “[Example management classes](#)” on page 186 for a description of the management classes for these data sets. The names of the data sets are not precisely as the system would generate them. They have been shortened and modified for convenience in using them for illustrations.

Note: The management class name has been used as the high-order qualifier for the data set names for your convenience in comparing the conditions of the data sets with the attributes specified in the management classes. This is purely for educational purposes. You are not required to use the management class name as any qualifier in the data set name for SMS-managed data sets.

Table 25. Data Set Characteristics for Availability Management Example

Data set name	Number of existing backups	Last backup (days)	Data set changed indicator	Data set was backed up
GDSMC.G3.V1	0	0	Yes	Yes
GDSMC.G3.V2	1	1	Yes	Yes
LARGMC.G4.E1	4	3	Yes	Yes
LARGMC.G4.E2	4	2	No	No
NOMIGMC.G4.G1	0	4	Yes	No
STANDMC.G4.H1	3	2	Yes	No

As DFSMSHsm processes each of the data sets, the following occurs:

- Data set GDSCMC.G3.V1 is backed up because the data-set-changed indicator is on and no backup version exists.
- Data set GDSCMC.G3.V2 is backed up because the data-set-changed indicator is on and the data set has met the BACKUP FREQUENCY specified in management class GDSCMC. However, because a backup version of the data set already exists and the new backup copy exceeds NUMBER OF BACKUPS (DATA SET EXISTS) specified in management class GDSCMC, DFSMSHsm invalidates the old backup version.
- Data set LARGEMC.G4.E1 is backed up for the same reason as for GDSCMC.G3.V2. Note, however, that the days since last backup exceeds the value specified for BACKUP FREQUENCY in management class LARGEMC. This happened because the data set was last backed up on Friday. (Remember, the example system is running on a Monday.) Whether the data set changed on Friday (after backup ran), Saturday, Sunday, or Monday, it could not be backed up until Monday. This data set also causes a roll-off of the existing backup version.
- Data set LARGEMC.G4.E2 is not backed up because the data-set-changed indicator is set off.
- Data set NOMIGMC.G4.G1 is not backed up even though the data-set-changed indicator is on and the last backup meets the BACKUP FREQUENCY attribute in management class NOMIGMC. The data set is not backed up because the AUTO BACKUP attribute value is N in management class NOMIGMC.
- Data set STANDMC.G4.H1 is not backed up because, even though the data-set-changed indicator is on, the last backup days value does not meet the BACKUP FREQUENCY specified in management class STANDMC.

Command availability management

Previous topics have described how you can automatically cause DFSMSHsm to dump and back up your data sets. However, you may want to dump a volume or back up data sets or volumes at a time other than when automatic availability management would cause the backup or dump. Also, when a data set or volume is damaged, either physically or logically, you want to access the dumped or backed up versions to replace the damaged data. Command availability management allows you to accomplish these functions. You can perform the following tasks using availability management commands:

- Dump a volume to either the dump classes specified in the command or the dump classes that are specified in the storage group.
- Dump the volumes in one or more storage groups.
- Back up an individual data set.
- Use the data set backup by command capabilities.
- Use the UNIX file backup by command capabilities.
- Back up all eligible changed data sets on a volume.
- Back up all eligible changed data sets in one or more storage groups.
- Back up all eligible changed UNIX files in a directory subtree.
- Back up all supported data sets on a volume.
- Back up all supported data sets in one or more storage groups.
- Back up all supported UNIX files in a directory subtree.
- Delete expired backup versions from the backup volumes.
- Delete specific backup versions from the backup volumes.
- Recover an individual data set from either a dump copy or an incremental backup version, whichever is the more recent.
- Recover an individual data set from an incremental backup version.
- Recover an individual data set from a dump copy.
- Recover an individual UNIX file.

- Restore a complete volume from a dump copy and update the data sets from the latest incremental backup versions.
- Restore a complete volume from a dump copy.
- Recover a volume by using only incremental backup versions.
- Recover all UNIX files in a directory subtree.

Dumping a volume

Issue one of the following commands to cause a command dump of one or more volumes:

```
BACKVOL VOLUMES(volser,...,volser) DUMP
BACKVOL VOLUMES(volser,...,volser) -
DUMP(DUMPCCLASS(class,class,class,class,class))
BACKVOL VOLUMES(volser,...,volser) -
DUMP(DUMPCCLASS(class, . . . ,class) -
RETENTIONPERIOD(days|*|NOLIMIT, . . . ,days|*|NOLIMIT))
```

For all of the commands, DFSMSHsm checks to ensure that SMS is installed and active in the system. If SMS is installed and active:

- For the first form of the command, the dump operates as explained in [“Dump processing” on page 193](#).
- For the second form of the command, dump processing is the same as is described in [“Dump processing” on page 193](#) except that the dump classes specified by the command are used instead of the dump classes specified by the storage group.
- For the third form of the command, dump processing is the same as is described in [“Dump processing” on page 193](#) except that the dump classes specified by the command are used instead of the dump classes specified by the storage group, and the retention periods specified in the command are used instead of the retention periods specified by the dump classes.

Dumping the volumes in one or more storage groups

To dump all the volumes in one or more storage groups, use the command:

```
BACKVOL STORAGEGROUP(sgm) DUMP
```

The processing for dumping each volume in a specified storage group is the same as explained in [“Dumping a volume” on page 222](#). As in that case, there is an additional available option for stacking dump copies as shown in the following example:

```
BACKVOL STORAGEGROUP(sgn) -
DUMP(DUMPCCLASS(class,class,...,class) STACK(nn,nn,...,nn))
```

You can specify a stacking value, as follows:

- In the STACK parameter on the BACKVOL command
- If the STACK parameter is not specified on the BACKVOL command, in the definition of the dump class specified on the BACKVOL
- If the DUMPCCLASS parameter is not specified, in the definition of a dump class associated with a specified storage group

If the stacking value is greater than 1, DFSMSHsm stacks up to *nn* dump copies on the dump volumes of the dump classes used.

Backing up an individual data set or UNIX file

You can use one of the following data set backup commands to back up a single data set or UNIX file:

BACKDS *dsname* | 'UNIX_filename' | 'UNIX_path+pattern'

HBACKDS *dsname/dsn_filter_name* | 'UNIX_filename' | 'UNIX_path+pattern'

hbackup UNIX_filename | UNIX_path | "UNIX_path+pattern"

Note: You can also use the ARCINBAK program and the ARCHBACK macro to submit data set or UNIX file backup requests. DFSMSHsm processes these requests according to descriptions in this topic.

In processing these commands, DFSMSHsm first checks the management class for the data set or the zFS data set containing the UNIX file, to determine the value of the ADMIN OR USER COMMAND BACKUP attribute. If the value of the attribute is BOTH, a DFSMSHsm-authorized user can use either of the commands, and a non-DFSMSHsm-authorized user can use the HBACKDS command to back up the data set or UNIX file, or the UNIX **hbackup** command to back up the UNIX file. If the value of the attribute is ADMIN, a DFSMSHsm-authorized user can use either of the commands to back up the data set or UNIX file, but a non-DFSMSHsm-authorized user cannot back up the data set. If the value of the attribute is NONE, the command backup cannot be done.

The following process occurs when an individual data set or UNIX file is backed up to DASD:

1. DFSMSHsm can create backup copies to either ML1 backup OVERFLOW volumes or NOOVERFLOW volumes. Use the SETSYS ML1OVERFLOW(DATASETSIZE(*dssize*)) command to specify the minimum size in K bytes (where K=1024) of the data set or UNIX file for which an ML1 OVERFLOW volume is preferred for the migration or backup copy.
2. For data sets, or UNIX files, smaller than 58K tracks (3,365,388,288 bytes for a UNIX file), DFSMSHsm allocates a basic sequential format data set for the backup copy.
3. For data sets larger than 58K tracks (3,365,388,288 bytes for a UNIX file), DFSMSHsm allocates a large format sequential data set for the backup copy. Basic or large format sequential data sets will prefer OVERFLOW or NOOVERFLOW volumes based on the SETSYS ML1OVERFLOW(DATASETSIZE(*dssize*)) value. If there is not enough free space on the NOOVERFLOW or OVERFLOW volume for a particular backup copy, DFSMSHsm tries to create the backup on an OVERFLOW or NOOVERFLOW volume respectively. If the data set, or UNIX file, is too large to fit on a single ML1 volume (OVERFLOW or NOOVERFLOW), then the migration or backup fails.
4. Build an MCM record in the BCDS to indicate that the backup copy must be moved to a backup volume.
5. Build a UFN record in the BCDS for the absolute path of the file. Additionally build UFN records for each parent directory that does not already have a UFN record in the BCDS. The first backup of a file can result in a significant amount of I/O to the BCDS, and thus take longer.
6. Set the data-set-changed indicator to off and update the last backup date fields in the catalog entry and the BCDS MCB record. For UNIX files, the **Last Reftime** field in the file stat area for the UNIX file is updated to the current date or time. (In the past, this field has held the last backup date.) It is used in a manner similar to the data-set-changed indicator so that DFSMSHsm can determine whether it needs to take another backup of the file.
7. Invalidate the oldest versions if the number of backup versions exceeds the value that is specified by the NUMBER OF BACKUPS (DATA SET EXISTS) management class attribute.

If the backup to DASD is unsuccessful, a message is issued.

The following process occurs when an individual data set or UNIX file is backed up to tape:

8. Copy the data set or UNIX file to a backup tape volume.
9. Set the data-set-changed indicator to off and update the last backup date fields in the catalog entry and the BCDS MCB record. For UNIX files, the **Last Reftime** field in the file stat area for the UNIX file is updated to the current date or time. (In the past, this field has held the last backup date.) It is used in a manner similar to the data-set-changed indicator so that DFSMSHsm can determine whether it needs to take another backup of the file.
10. Build a UFN record in the BCDS for the absolute path of the file. Additionally build UFN records for each parent directory that does not already have a UFN record in the BCDS. The first backup of a file can result in a significant amount of I/O to the BCDS, and thus take longer.

11. Invalidate the oldest versions if the number of backup versions exceeds the value that is specified by the NUMBER OF BACKUPS (DATA SET EXISTS) management class attribute.

If the backup to tape is unsuccessful, a message is issued.

DFSMSHsm does not back up an SMS-managed data set or UNIX file that is contained in an SMS-managed zFS if SMS is not installed and active in the system.

Note: Any files in an encrypted zFS are decrypted as they are written to the backup.

Using the data set or UNIX file backup by command capabilities

The data set back up by command function provides the following capabilities:

- Up to 64 data sets or UNIX files per host can be backed up at one time.
- Data sets or UNIX files can be backed up directly to DASD or to tape.
- If concurrent copy is specified, users are notified when the data set or UNIX file backup is either logically or physically complete. Concurrent copy supports non-SMS data sets or UNIX files contained in a non-SMS managed zFS, or if specified on the command, concurrent copy overrides the management class for an SMS data set or UNIX files contained in a non-SMS managed zFS.
- Users can unmount continuously mounted backup tapes.
- Users can tailor the times when DFSMSHsm unmounts a tape.
- Users can specify the number of days to retain a specific backup copy of a data set or UNIX file. See [“Specifying the number of days to retain a backup copy of a data set” on page 183.](#)
- Users can create a data set backup copy and store it under a different, specified data set name. See [“Data set backup using NEWNAME” on page 454.](#)
- Users can create a UNIX file backup copy as if it was originally backed up from a subset of the current path.
- Users can request removal of the UNIX file after a successful backup.
- Users can tailor how backup copies of in-use UNIX files are made using the INUSE keyword or -I option. See [“Backing up UNIX files in use at time of backup” on page 224.](#)

Because the data set or UNIX file backup function allows users to target their backups directly to tape, this might increase the number of required tape drives and mounts. The SETSYS DSBACKUP and the HOLD BACKUP(DSCOMMAND) commands manage the task levels of both tape and ML1 DASD.

Backing up UNIX files in use at time of backup

The INUSE keyword and -I option allows users to tailor how backup copies of in-use UNIX files are made. When the INUSE keyword or -I option is used, UNIX files that are in use at the time of backup will have the backup retried until exclusive serialization can be obtained or the max number of retries has been made. If after the max number of retries has been made and exclusive serialization cannot be obtained the request will either fail or a backup will be taken anyway (a fuzzy backup), depending on what was specified on the command. The INUSE keyword and -I option have subkeyword and suboptions that specify the number of retries, the amount of time to delay in between retries, and whether serialization is required (backup request will fail if exclusive serialization cannot be obtained) or preferred (a fuzzy backup will be taken if exclusive serialization cannot be obtained). The user will be notified if a fuzzy backup of the UNIX file was made.

Specifying data set backup tasks

The data set and UNIX file backup commands allow a user to target a specific output device type. However, if an output device type is not specified, the SETSYS DSBACKUP command controls whether individual backup requests go to DASD or tape. It also allows you to define the number of tasks that are available, up to 64 tasks, to the data set and UNIX file backup function. The SETSYS DSBACKUP command has three parameters: DASDSELECTIONSIZE, DASD, and TAPE.

The DASDSELECTIONSIZE parameter allows you to balance the workload between DASD and tape tasks for all WAIT requests that do not target an output device. This parameter is applicable if both tape and ML1 DASD are used for data set and UNIX file backup requests. Defaults for this parameter are 3000 KB for the *maximum* size and 250 KB for the *standard* size. For a discussion about assigning values for *maximum* and *standard* sizes, see [“Assigning DASDSELECTIONSIZE values”](#) on page 226.

The DASD parameter allows you to specify the number of concurrent data set or UNIX file backup tasks that can direct backups to ML1 DASD. The value *nn* for the TASKS subparameter ranges from 0 to 64 tasks. The command can take the following form:

```
SETSYS DSBACKUP(DASD(TASKS(nn)))
```

The TAPE parameter allows you to specify the maximum number of concurrent data set or UNIX file backup tasks that can direct backups to tape. The value *nn* for the TASKS subparameter ranges from 0 to 64 tasks. The command can take the following form:

```
SETSYS DSBACKUP(TAPE(TASKS(nn)))
```

Note:

1. The DASDSELECTIONSIZE parameter allows you to balance the workload between DASD and TAPE tasks for all WAIT requests that do not specify output media. By setting appropriate values, you ensure that a sufficient number of backups go to each media type to keep them busy. Ultimately, you must weigh the efficient use of data set or UNIX file backup tasks against media use and the need to perform a second movement of backup versions from ML1 DASD to backup tapes during automatic backup processing.
2. The sum of both DASD and tape tasks may not exceed 64.
3. If your installation directs a large amount of individual data set or UNIX file backups to tape, you may want to reassess the number of ML1 OVERFLOW volumes that you have defined to DFSMSHsm.
4. A backup to tape is a one-time process, but a backup to DASD means that the data first goes to DASD before it moves to tape. To avoid this additional data movement, it is almost always more efficient to back up to tape. (An environment with small data sets or UNIX files that arrive infrequently may be the exception.)

Additional TAPE subparameters are MAXIDLETASKS and DEMOUNTDELAY. The MAXIDLETASKS option allows you to tell DFSMSHsm the maximum number of idle tape tasks (0 to 64) that you will accept for each delay. The DEMOUNTDELAY option allows you to tell DFSMSHsm how much time to wait before it deallocates idle tape drives.

Related reading

For more information about the relationship between maximum and standard values associated with the DASDSELECTIONSIZE keyword and the SETSYS DSBACKUP command, see Chapter 65, [“SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,”](#) on page 1057.

Controlling target devices that are used by data set backup

Because DFSMSHsm can use both tape and DASD media types for data set or UNIX file backup requests, there is a need to balance existing workloads against available or allowed resources. In general, tapes take longer from initial selection to first write than DASD does, but tape may be faster in terms of data throughput. By tailoring the SETSYS DSBACKUP command to define the number of tasks that are available to the data set or UNIX file backup function, you can ensure that DFSMSHsm efficiently directs your backup to either tape or DASD, depending on the workload.

The DASDSELECTIONSIZE (*maximum standard*) parameter on the SETSYS DSBACKUP command allows DFSMSHsm to balance the workload between DASD and tape tasks for all WAIT requests that do not target a device type. DFSMSHsm uses a selection process that is based on data set or UNIX file sizes and the availability of tasks. If the data set or UNIX file size is greater than a specified threshold *maximum*,

then the data set or UNIX file is directed to tape. Any data set or UNIX file that is less than or equal to a specified threshold *standard* is directed to the first available task, either DASD or tape.

Note: If you specify the subparameter TASKS(0) for either DASD or tape, DFSMSHsm never directs nontargeted requests to that media.

There are three distinct data set or UNIX file size ranges:

- A small data set or UNIX file is any data set with a size less than the value of *standard* in DASDSELECTIONSIZE(*maximum standard*).
- A medium data set or UNIX file is any data set larger than a small data set but less than the value of *maximum* in DASDSELECTIONSIZE(*maximum standard*).
- A large data set or UNIX file is any data set that is greater than a medium data set.

Note:

1. There is a direct correlation between the number of defined tasks (tape and DASD) and the size of data sets or UNIX files that are directed to DASD. This correlation can result in double data movement during the automatic backup function. (See “Assigning DASDSELECTIONSIZE values” on page 226 for help with assigning size values to *maximum* and *standard*.) The automatic backup function can delay the start of the multitasked volume backup function; that is, volume backup will not begin until all backup versions on ML1 have been moved to tape. You can use the FREEVOL ML1BACKUPVERSIONS command to initiate movement of all backup versions off all ML1 volumes to daily backup volumes.
2. DFSMSHsm starts tape tasks immediately on receipt of WAIT requests (medium and large) and NOWAIT requests, up to the TAPE (DEMOUNTDELAY(MAXIDLETASKS)) value.
3. When both tape and DASD tasks are allowed, NOWAIT requests are only processed by tape tasks.
4. DFSMSHsm starts DASD tasks and tape tasks up to the allowable value when there is enough work for the tasks to do. New tasks are started that are based on the number of requests not currently being processed.

Assigning DASDSELECTIONSIZE values

Small, medium, and large data set or UNIX file sizes are established when you specify the SETSYS DSBACKUP DASDSELECTIONSIZE(*maximum standard*) command. The values assigned to *maximum* and *standard* essentially help balance the workload between DASD and tape tasks. Default sizes for this parameter are 3000 KB for *maximum* and 250 KB for *standard* sizes.

The following considerations might help you decide what to specify for the *maximum* and *standard* values that are associated with the DASDSELECTIONSIZE parameter:

- Increasing either the *maximum* or *standard* value:
 - Increases the number of data sets or UNIX files that may be directed to DASD and decreases the number of data sets or UNIX files that may be directed to tape
 - Uses more ML1 DASD space
 - Increases the workload for autobackup because more backup versions must be moved from ML1 DASD to tape
- Decreasing the *maximum* or *standard* value:
 - Decreases the number of data sets or UNIX files that can be directed to DASD and increases the number of data sets or UNIX file that can be directed to tape
 - Uses less ML1 DASD space
 - Reduces the workload for autobackup since less backup versions must be moved from ML1 DASD to tape

Note: The default is DASDSELECTIONSIZE(3000 250), where 3000 is the maximum in KB, and 250 is the standard in KB. The categories in the table are based on these default values. If the you change the DASDSELECTIONSIZE(maximum standard) settings, then these categories change.

Table 26. Determining Data Set Sizes and Their Task Targets

Default Criteria	Large	Medium	Small
Data Set Size	3001 KB and up	251 – 3000 KB	0 – 250 KB
WAIT Requests	Favor tape	First available	Favor disk
NOWAIT Requests	Favor tape	Favor tape	Favor tape

Related reading

For more information about how to determine the size of a data set and whether it will use tape or disk, see Chapter 65, “[SETSYS command: Establishing or changing the values of DFSMSHsm control parameters](#),” on page 1057.

Switching data set or UNIX file backup tapes

When you specify a value other than zero for SETSYS DSBACKUP(TAPE(DEMOUNTDELAY(MAXIDLETASKS(*n*)))), the data set or UNIX file backup function may allow output tapes to be continuously mounted for an indefinite amount of time. There may be times when you want or need to demount the tapes. You can use the SWITCHTAPES keyword and parameters to demount a continuously mounted tape. You may want to do so, for example, so that it or an alternate can be taken to another site for disaster backup, or so that you can install new tape devices. The SWITCHTAPES keyword is available on both the DEFINE and HOLD commands.

Note: If you specify the MARKFULL parameter of the DEFINE SWITCHTAPES(DSBACKUP) command, the tapes that are released from command data set or UNIX file backup are marked full. However, other partial tapes in the data set or UNIX file backup inventory (previous partial tapes) remain as selection candidates. These partial tapes may be immediately selected and mounted for output processing.

DEFINE SWITCHTAPES tells DFSMSHsm that, at specific times or events, you want the data set or UNIX file backup function to automatically demount and deallocate all currently mounted tapes and allocated drives. HOLD(BACKUP(DSCOMMAND)) tells DFSMSHsm that you want to immediately quiesce data set or UNIX file backup activity to tape, demount all mounted tapes, and deallocate all tape devices. Therefore, if you plan ahead, you can use the DEFINE command to switch tapes. For unplanned activity, use the HOLD command to switch tapes.

Note: A tape can remain continuously mounted for a long period of time because both new tape technologies and a steady stream of data set backup requests allow an increased amount of data on the tape. It can also be mounted for a long time because the MAXIDLETASKS and DEMOUNTDELAY parameters on the SETSYS DSBACKUP command specify that tapes can be continuously mounted while inactive.

The DEFINE SWITCHTAPES(DSBACKUP) command allows you to choose among the following parameters:

TIME

Indicates that you want to define a time when DFSMSHsm initiates the demount and deallocation of all mounted tapes, performed at a certain time of day.

AUTOBACKUPEND

Indicates that you want DFSMSHsm to initiate the demount and deallocation of all mounted tapes at the end of the autobackup cycle.

PARTIALTAPE

Indicates the method that DFSMSHsm uses to mark an output tape as full. This is an optional keyword. MARKFULL, REUSE, and SETSYS are all valid options. SETSYS indicates that you want DFSMSHsm to use the SETSYS PARTIALTAPE value.

The HOLD BACKUP(DSCOMMAND(SWITCHTAPES)) command tells DFSMSHsm to perform SWITCHTAPES processing now; that is, to demount and to deallocate drives. DFSMSHsm demounts active tapes at the end of the current data set, but immediately demounts idle tapes. The demounted tapes are marked

full or not, according to the existing PARTIALTAPE parameter of the DEFINE SWITCHTAPES command. No subsequent RELEASE command is necessary. After these tapes are demounted and the drives are deallocated, data set backup continues with newly selected tapes, if requests are present.

Related reading

For more information about the exact syntax of the DEFINE and HOLD commands, see [Chapter 40, “DEFINE command: Defining control structures for use by DFSMSHsm,”](#) on page 809 and the [Chapter 50, “HOLD command: Preventing DFSMSHsm functions from running,”](#) on page 895.

Taking a backup tape away from a data set backup task

Depending on how you have defined your data set or UNIX file backup by command environment and the arrival rates of data set or UNIX file backup requests, backup output tapes may be mounted for extended periods of time. Consequently, DFSMSHsm has a recover tape takeaway function. This function allows, for instance, the timely recovery of a data set or UNIX file when its backup was made in a previous job step.

For an explanation of backup tape contention possibilities and the scope of the recover tape-takeaway function, see [Chapter 10, “Other availability management considerations,”](#) on page 445.

Targeting data set backups to required device types

To target data set or UNIX file backups to a required target output device type, such as ML1 DASD or backup tape, use the following commands:

```
BACKDS dsname|'UNIX_filename' TARGET(DASD)
```

```
BACKDS dsname|'UNIX_filename' TARGET(TAPE)
```

TARGET DASD specifies that you want to back up the data set or UNIX file initially to ML1 DASD. TARGET TAPE specifies that you want to back up the data set or UNIX file directly to tape.

Users can specify the TARGET keyword on any of the data set backup commands. For example, the BACKDS *dsname* TARGET(DASD) NOWAIT command forces the backup copy to DASD. The TARGET keyword may override any device output or task that DFSMSHsm might have selected on its own.

Using concurrent copy for data set or Clone for UNIX file backup

The BACKDS command has the optional CC keyword, which allows you to specify concurrent copy for both SMS and non-SMS data sets, or UNIX File Snapshot for UNIX files.

```
BACKDS dsname CC(PREFERRED)
```

```
BACKDS dsname CC(REQUIRED)
```

```
BACKDS dsname CC(VIRTUALPREFERRED)
```

```
BACKDS dsname CC(VIRTUALREQUIRED)
```

```
BACKDS dsname CC(CACHEREQUIRED)
```

```
BACKDS dsname CC(CACHEPREFERRED)
```

REQUIRED indicates that concurrent copy must be used as the backup method. If virtual concurrent copy is unavailable cache-based concurrent copy is used. The data set backup should fail if both types of concurrent copy are not available. When processing a UNIX file, REQUIRED indicates that UNIX File SnapShot must be used as the backup method.

PREFERRED indicates that you want to use concurrent copy as the backup method, if it is available. If virtual concurrent copy is not available, cache-based concurrent copy is used. If both types of concurrent

copy are not available, the data set will be backed up as if this keyword were not specified. . When processing a UNIX file, PREFERRED indicates that you want to use UNIX File SnapShot as the backup method, if it is available. If UNIX File SnapShot is unable to be used, the file will be backed up as if this keyword were not specified.

VIRTUALREQUIRED indicates that virtual concurrent copy must be used as the backup method. The data set backup should fail if virtual concurrent copy is not available. When processing a UNIX file, this keyword is treated as if REQUIRED were specified.

VIRTUALPREFERRED indicates that you want to use virtual concurrent copy as the backup method, if it is available. If virtual concurrent copy is not available, the data set will be backed up as if this keyword were not specified. When processing a UNIX file, this keyword is treated as if PREFERRED were specified.

CACHEREQUIRED indicates that cache-based concurrent copy must be used as the backup method. The data set backup should fail if cache-based concurrent copy is not available. When processing a UNIX file, this keyword is treated as if REQUIRED were specified.

CACHEPREFERRED indicates that you want to use cache-based concurrent copy as the backup method, if it is available. If cache-based concurrent copy is not available, the data set will be backed up as if this keyword were not specified. When processing a UNIX file, this keyword is treated as if PREFERRED were specified.

You can also specify the STANDARD keyword, which indicates that you want to use a standard backup method (without using concurrent copy). The standard backup method may possibly override a management class option to use concurrent copy.

Note: For SMS data sets or UNIX files, DFSMSHsm uses the management class values if you do not specify the CC parameter. (For UNIX files, the management class value is inherited from the value of the zFS data set that contains the UNIX file.) For non-SMS data sets or UNIX files, DFSMSHsm does not use concurrent copy if you do not specify the CC parameter.

There are two additional options: PHYSICALEND (PE) and LOGICALEND (LE).

- PHYSICALEND (the default) indicates that control returns to the application only after the backup has physically completed.
- LOGICALEND indicates that control returns to the application when the concurrent copy initialization completes. If PREFERRED was specified and concurrent copy is unavailable for the data set or UNIX file, the backup continues as if this keyword had not been specified. The BACKDS command fails if you specify both STANDARD and LOGICALEND keywords.

DFSMSHsm uses the STGADMIN.ADR.DUMP.CNCURRNT facility class resource to limit concurrent copy use to those users who are authorized to use the CC option.

Authorizing the use of concurrent copy: By default, all users can use the CC option. However, if RACF indicates a lack of authority, DFSMSHsm fails the data set or UNIX file backup request if the concurrent copy request is specified as REQUIRED, VIRTUALREQUIRED or CACHEREQUIRED. If the concurrent copy request is not specified as REQUIRED, VIRTUALREQUIRED or CACHEREQUIRED and RACF indicates a lack of authority, DFSMSHsm continues to back up the data set or UNIX file as if the concurrent copy keyword were not specified on the backup command.

Note: UNIX File Snapshot can only be initiated from the owning system for files located in a zFS that is mounted NORWSHARE. If the zFS is mounted RWSHARE, UNIX file Snapshot can be used from any system in the sysplex.

Authorizing the use of concurrent copy

By default, all users can use the CC option. However, if RACF indicates a lack of authority, DFSMSHsm fails the data set backup request if the concurrent copy request is specified as REQUIRED, VIRTUALREQUIRED or CACHEREQUIRED. If the concurrent copy request is not specified as REQUIRED, VIRTUALREQUIRED or CACHEREQUIRED and RACF indicates a lack of authority, DFSMSHsm continues to back up the data set as if the concurrent copy keyword were not specified on the backup command.

Inline backup of an individual data set or UNIX file

Use the ARCINBAK program to back up individual data sets or UNIX files in a batch environment and to support the JCL back reference of data set or UNIX file names and relative GDG data set names. The processing for backing up individual data sets or UNIX files with ARCINBAK is the same as explained in [“Backing up an individual data set or UNIX file” on page 222](#).

For information about how to use inline backup, see [“Inline backup” on page 447](#).

Backing up all data sets on one or more volumes

To back up all the data sets on a volume, use the command:

BACKVOL VOLUMES(*volser*,...,*volser*) TOTAL

The processing for backing up all the data sets on a volume by command is the same as that explained in [“Phase 4: Backing up SMS-managed volumes from storage groups with the automatic backup attribute” on page 201](#) except:

- DFSMSHsm does not retrieve a list of volumes to be backed up.
- DFSMSHsm checks only the ADMIN OR USER COMMAND BACKUP attribute from the management class for each data set. The SMS-managed data set is eligible to be backed up unless the ADMIN or USER = NONE. All other values for this attribute and all other attributes in the management class are ignored.
- Non-SMS-managed data sets excluded by the ALTERDS or HALTERDS VERSIONS(0) command are not backed up.
- If you specify the BACKUPDEVICECATEGORY parameter of the BACKVOL command, the data sets are backed up to the device type you specify.

When the BACKVOL VOLUME command is issued for an SMS-managed volume, that volume is considered to be DFSMSHsm-managed. DFSMSHsm will maintain control records for the volume until the DELVOL command is issued against it.

Backing up all data sets in one or more storage groups

To back up all the data sets in a storage group, use the command:

BACKVOL STORAGEGROUP(*sgn*,...,*sgn*) TOTAL

The processing for backing up all the data sets in a storage group by command is the same as is explained in [“Phase 4: Backing up SMS-managed volumes from storage groups with the automatic backup attribute” on page 201](#), except for the following difference:

- DFSMSHsm checks the AUTO BACKUP attribute from the management class for each data set. The SMS-managed data set is eligible to be backed up if AUTO BACKUP=YES. All other values for the command backup attribute and all other attributes in the management class are ignored.
- If you specify the BACKUPDEVICECATEGORY parameter of the BACKVOL command, the data sets are backed up to the device type you specify.
- When the BACKVOL STORAGEGROUP command is issued, the volumes processed are considered to be SMS-managed.

Backing up all UNIX files in a directory

To back up all the UNIX files within a directory, use the command:

```
BACKDS '/directory/directory/'
```

```
BACKDS '/directory/directory/*'
```

DFSMSHsm looks for the last forward slash character in the requested name. The name ending in a forward slash indicates to DFSMSHsm that the entire directory should be backed up. Additionally, if the

requested name ends with only an asterisk character, this also indicates that the entire directory should be processed. See the “UNIX file wildcard processing” on page 231 section for more discussion.

DFSMSHsm then reads the directory contents to find the individual file and directory names that may be processed. An internal request for each supported file and empty directory is created and added to the DFSMSHsm backup queue to be processed in parallel according to the definitions of the SETSYS DSBACKUP setting. See “Specifying data set backup tasks” on page 224 for more information on that setting.

To back up files within subdirectories, specify the RECURSE keyword.

```
BACKDS '/directory/directory/' RECURSE(CROSSMOUNTS)
```

The RECURSE keyword indicates that DFSMSHsm should also look at the contents of directories that it finds for files and directories to process. The directory is read at the time DFSMSHsm begins to process that internal request, which could be different from the time of the original request.

The sub-parameter, CROSSMOUNTS, indicates that DFSMSHsm can back up files that are contained in a different filesystem compared to where the original request began. DFSMSHsm will continue to examine sub-directories until no more directories are found. The default processing of NOCROSSMOUNTS can be used to limit the processing to files that are contained within a single zFS data set.

Note: The initial backup of each UNIX file can take longer due to the additional Backup Control Data Set record creation that must be done. This can cause a large growth in the size of the BCDS and should be considered when sizing the BCDS.

UNIX file wildcard processing

The UNIX path name you pass to DFSMSHsm can include wildcard characters to request selection of filenames that match a pattern. This can be used to back up all the log files, or specific subsets of files within a directory.

The UNIX path name passed to DFSMSHsm is made up of two parts: a directory path, and a filename pattern.

```
/dir1/dir2/dir3/file_pattern
```

The directory path ends with the last forward slash character and represents a specific nested set of directories beginning with root (/). Any asterisk (*), question mark (?), or backslash (\) characters in the directory path are treated as simple characters in the directory names and are not used in wildcard matching.

The filename pattern can contain an asterisk anywhere within the pattern to indicate that 0 or more characters can be used to match, question marks to indicate that any single character can match that position. If the pattern should contain an asterisk or question mark character, the backslash character must be used to indicate that the following asterisk or question mark is not a wildcard. Additionally, the pattern may end in an asterisk to indicate that all following characters are acceptable. The pattern is used to compare against file names in a directory. If the RECURSE keyword is specified, then the pattern is used to compare against file names in the sub-directories as well.

Some examples of patterns are:

- BACKDS '/dir1/subdir1/*.txt' RECURSE - Start in the /dir1/subdir1/ directory, and back up all files that end in .txt. Additionally, if there are sub-directories, any files that end in .txt should also be processed. Processing of a sub-directory tree will stop when a different zFS filesystem is encountered.
- BACKDS '/dir1/subdir1/abc*.log' - Back up all files in /dir1/subdir1/ that start with abc and end with .log. Do not examine sub-directories because the RECURSE keyword is not specified.
- BACKDS '/dir1/subdir1/app-12???.log' - Back up all files in /dir1/subdir1 that start with app-12, followed by any two characters, ending with .log.
- BACKDS '/dir1/subdir1/comment\?*out*' - Back up all files in /dir1/subdir1 that start with comment?, and contain .out afterward.

Note: The initial backup of each UNIX file can take longer due to the additional Backup Control Data Set record creation that must be done. This can cause a large growth in the size of the BCDS and should be considered when sizing the BCDS.

Backing up all eligible changed data sets on one or more volumes

You can back up all the eligible changed data sets on one or more volumes by issuing either of the following commands:

BACKVOL VOLUMES(*volser*,...,*volser*)

BACKVOL VOLUMES(*volser*,...,*volser*) INCREMENTAL

Processing for either of these commands is the same as that explained in [“Phase 4: Backing up SMS-managed volumes from storage groups with the automatic backup attribute”](#) on page 201, except:

- No list of volumes to process is retrieved
- If you specify the BACKUPDEVICECATEGORY parameter of the BACKVOL command, DFSMSHsm backs up the data sets to the volume type you specify

Backing up all eligible changed data sets in one or more storage groups

You can back up all of the eligible changed data sets in one or more storage groups by entering either of the following commands:

BACKVOL STORAGEGROUP(*sgn*,...,*sgn*)

BACKVOL STORAGEGROUP(*sgn*,...,*sgn*) INCREMENTAL

Processing for either of these commands is the same as explained in [“Backing up all eligible changed data sets on one or more volumes”](#) on page 232, in which DFSMSHsm determines each volume involved from the definitions of the specified storage groups.

When you use the **STORAGEGROUP** parameter, DFSMSHsm checks the AUTO BACKUP attribute of the management class for each changed data set, and does the following:

AUTO BACKUP=YES...

DFSMSHsm performs the remainder of eligibility processing for the data set.

AUTO BACKUP=NO...

DFSMSHsm does not process the data set.

Backing up all eligible changed UNIX files in a directory

You can back up all of the eligible changed data sets in one or more storage groups by entering either of the following commands:

BACKDS '*dir1/dir2*' CHANGEDONLY

BACKDS '*dir1/dir2/subdir/' RECURSE CHANGEDONLY**

Processing for either of these commands is the same as explained in [“Backing up all UNIX files in a directory”](#) on page 230. Before creating the internal request to process a file, the Last Reference date field, which holds the date of its last backup, in the file metadata is examined. If the Modification date or Metadata Modification date is newer than the value in the Last Reference date, or if DFSMSHsm does not have a backup of the file, the internal request for that file is created. If the UNIX file resides in a file system that is mounted in Read-Only mode, then the Last Reference date is not examined. Instead, the timestamp of the most recent backup in the DFSMSHsm inventory is used to determine if the file has changed since its last backup was taken. DFSMSHsm does not attempt to update the reftime field for the UNIX file to set the last backup date.

Additionally, if you want DFSMSHsm to look in subdirectories for eligible files, the RECURSE keyword might be specified along with the CHANGEDONLY keyword. This might be used to schedule backups of the same directory on a recurring basis, but not take up unnecessary space with duplicate backups.

Note: The CHANGEDONLY parameter is defaulted for UNIX files. If a new backup should be taken, even though the UNIX file has not changed since its last backup, the TOTAL keyword must be specified.

Deleting expired backup versions from the backup volumes

Availability management does not delete data set or UNIX file backup versions when a data set or UNIX file is deleted. Doing so defeats the purpose of availability management. Also, retained backup versions that have met their retention period are not automatically deleted. Therefore, over time, unneeded backup versions can accumulate on your backup volumes. You can use the EXPIREBV EXECUTE command to find and delete these unneeded backup versions.

The EXPIREBV command initiates a search of the BCDS for old, unwanted backup versions and deletes them. The old and unwanted backup versions are deleted based on the BACKDS command retention days value that is specified during data set or UNIX file backup and attributes in the management class for each data set or UNIX file. The EXECUTE parameter causes the BCDS records to be deleted. Each time the EXPIREBV command is issued, all versions that have met their retention duration will expire, except for the most recent version.

Note: While data sets and UNIX files are being changed, new backup versions are created and excess backup versions are deleted or retained. Backup versions are retained if they were created with a BACKDS command retention days value and have not exceeded the retention duration. The EXPIREBV command provides a capability to apply expiration criteria without creating a new backup version. However, if you modify only the maximum backup version limits and data sets or UNIX files are not backed up. Only by issuing the EXPIREBV command (or the BDELETE command) will the new expiration criteria be applied and expired backup versions deleted.

The EXPIREBV command must be issued twice to delete backup versions that were created without a BACKDS command retention days value. The first execution determines that the data set was scratched or UNIX file was deleted and stores the date the EXPIREBV command was issued as the scratch date in the control data set MCB record. The second execution deletes the backup versions when the requested time has passed. The requested time is specified in the management class for SMS data sets or using the EXPIREBV command for non-SMS data sets.

If you want to see a list of backup versions that will be deleted when the EXECUTE parameter is used, use the DISPLAY parameter instead. The output of the DISPLAY parameter can be sent to an output data set or SYSOUT, using the OUTDATASET or SYSOUT parameters. When the EXECUTE parameter is specified, the list is written to the backup activity log.

Related reading

- For information about how the EXPIREBV command works with ABARS, see [“DFSMSHsm coexistence considerations for aggregate groups and management classes”](#) on page 339 and [“Expiring ABR records in the BCDS”](#) on page 354.
- For more information about the parameters of the EXPIREBV command, see Chapter 44, [“EXPIREBV command: Deleting unwanted backup versions of data sets and UNIX files,”](#) on page 851.

EXPIREBV processing

The following terms are referred to in the following description: *date of scratch* or *scratch date*, *cataloged version*, *uncataloged version*, *active copies*, and *retained copies*:

- The date of scratch or scratch date refers to the date on which the EXPIREBV command determines that a data set or UNIX file for which a backup version exists has been deleted. The scratch date is generally not the date on which the data set or UNIX file was actually scratched. Instead, it is the date that EXPIREBV was next run after the data set was scratched or UNIX file was deleted. The date is stored in the BCDS MCB record in the 4-byte field MCBSCRD (offset X'48', packed decimal). It can be displayed using the FIXCDS command.

Note: DFSMSHsm honors a scratch date stored by the DFSMSHsm space maintenance and recovery utilities if the date is newer than the most recent backup date for the data set or UNIX file and if EXPIREBV has not already stored the scratch date itself.

- A cataloged version refers to a backup version of a data set that was cataloged when the backup version was made; that is, the data set was cataloged when it was backed up.
- An uncataloged version is a version of a data set that was not cataloged when the version was made; that is, the data set was uncataloged when it was backed up.
- UNIX files are always considered to be cataloged.
- Active copies are the set of backup copies created that have not yet rolled off.
- Retained copies are the set of backup copies that have rolled off from the active copies and have not yet reached their retention periods.

Expire backup versions (EXPIREBV) processing is performed in eight steps. DFSMSHsm:

1. Issues a starting message.
2. Sets up the output destination.
3. Retrieves a list of management class definitions.
4. Determines the starting point for the scan of the BCDS.
5. Determines the ending point for the scan of the BCDS.
6. Processes each data set or UNIX file record in the BCDS between the starting and ending points.
7. Ends the scan of the BCDS.
8. Issues an ending message.

Issue a starting message

If you specified DISPLAY in the EXPIREBV command, DFSMSHsm sends a message to the user ID that issued the command. If you specified EXECUTE, DFSMSHsm sends the starting message to the user and to the backup activity log.

Set up the output destination

If you specified DISPLAY in the EXPIREBV command, DFSMSHsm allocates and opens the output data set for the expire backup listing. DFSMSHsm uses the OUTDATASET | SYSOUT parameter that you specified in the EXPIREBV command to allocate the data set. If an error occurs in allocating or opening the output data set, DFSMSHsm issues a message, and EXPIREBV processing ends.

Retrieve a list of management class definitions

The management classes for the SMS-managed data sets contain the attributes that determine whether any particular backup version should be deleted. At the start of EXPIREBV processing, DFSMSHsm requests from SMS a list of the management classes that are defined to the configuration in which DFSMSHsm is running. Successful return of the list of management class definitions indicates that SMS is installed and active. If SMS is not installed and active, DFSMSHsm issues a message to indicate that EXPIREBV processing could not be done for SMS-managed data sets or UNIX files in an SMS-managed zFS data set.

Determine the starting point for the BCDS scan

The starting point for the scan of the BCDS depends on:

- Whether STARTKEY is specified in the command
- Whether RESUME is specified in the command
- Whether the key of the last data set processed (last-processed key) was saved at the end of the last EXPIREBV processing in this host

If STARTKEY is specified in the command, DFSMSHsm starts with the BCDS record that the command specifies.

If RESUME is specified in the command and the saved last-processed key is not blank, DFSMSHsm starts with the saved last-processed key.

If neither STARTKEY nor RESUME is specified in the command or if RESUME is specified but the saved last-processed key is blank, DFSMSHsm starts with the first record in the BCDS.

Determine the ending point for the BCDS scan

The ending point for the scan of the BCDS depends on:

- Whether ENDKEY is specified in the command.
- Whether RESUME is specified in the command.
- Whether a saved planned ending key exists from the previous EXPIREBV processing in this host. The planned ending key is saved at the beginning of EXPIREBV processing.

If ENDKEY is specified in the command, DFSMSHsm ends when the next record to be processed has a higher key value than that specified with ENDKEY.

If neither ENDKEY nor RESUME is specified in the command or if RESUME is specified without ENDKEY and the saved ending key is blank, DFSMSHsm ends EXPIREBV processing after the last BCDS record is processed.

If the saved planned ending key is not blank and RESUME is specified without ENDKEY in the command, DFSMSHsm ends EXPIREBV when the next record to be processed has a higher value than the saved planned ending key.

Process each data set record in the BCDS

For each data set or UNIX file record in the BCDS, DFSMSHsm performs the following processing:

- It determines whether the version is a retired version. See [“Data set retirement” on page 114](#) for a description of “delete if backed up”. If the version is a retired version, DFSMSHsm knows that it is not an SMS-managed data set or UNIX file and processes it according to rules for processing non-SMS-managed data sets or UNIX files. EXPIREBV processing for non-SMS-managed data sets is described in [Chapter 7, “Availability management of non-SMS-managed storage,” on page 303](#).

If the version is not a retired version, DFSMSHsm continues processing.

- It determines whether the data set exists and is SMS-managed. Or it determines that the zFS containing the UNIX file is SMS-managed.

For each data set that does not have a retired version and does not indicate that all backup versions are for an uncataloged data set and that does not have a scratch date that is recorded by prior EXPIREBV processing, DFSMSHsm determines the names of the storage class, management class, and data class for the data set or UNIX file.

If a catalog entry exists for the data set, the data set has not been scratched.

If the data set is not cataloged and the MCB record does not contain a scratch date, DFSMSHsm assumes that the data set has been scratched, and it records the current date in the scratch date field of the MCB record in the BCDS. DFSMSHsm resets an existing scratch date to the current date if the date of the most recent backup version is the same or later than the existing scratch date. Because DFSMSHsm cannot be sure, it assumes that the old scratch date is invalid. This is because the data set may have been re-created and backed up after prior EXPIREBV processing found that the data set had been scratched. This creates the possibility that more than one EXPIREBV command will need to be run to remove the backup versions, but it also prevents DFSMSHsm from deleting valid backup versions. The scratch date is stored when either DISPLAY or EXECUTE is specified on the EXPIREBV command.

DFSMSHsm then checks the MCB record to determine whether the data set or UNIX file was SMS-managed when it was last backed up. If the data set or UNIX file was SMS-managed when it was backed up, DFSMSHsm processes all of its cataloged backup versions as described here. If it was not SMS-managed, DFSMSHsm processes it as described in [“Processing for expired backup versions of non-SMS-managed data sets” on page 310](#).

If any error occurs, other than being unable to find the data set, DFSMSHsm assumes that the data set still exists and processes only the uncataloged versions of the data set as described in [“Processing for expired backup versions of non-SMS-managed data sets”](#) on page 310.

If the MCB record already contains a scratch date, DFSMSHsm assumes that the data set or UNIX file has been scratched and determines from the MCB if the data set or UNIX file was SMS-managed when it was last backed up.

- It identifies expired backup versions.

Each SMS-managed data set's or UNIX file's backup versions are processed based on the RETAIN DAYS value and the attributes that are specified in the management class associated with the data set or UNIX file (or in the default management class if the data set or UNIX file is not associated with a management class). If a management class name is found but there is no definition for the management class, DFSMSHsm issues a message and does not process the cataloged backup versions of that data set or UNIX file.

After the management class has been established, processing depends on whether the data set exists:

- If the data set or zFS exists, the management class used is the one associated with the existing data set or zFS.
 1. DFSMSHsm compares the number of backup versions that exist with the value of the NUMBER OF BACKUPS (DATA SET EXISTS) attribute in the management class. If there are more versions than requested, the excess versions are deleted if the versions do not have a RETAIN DAYS value or the RETAIN DAYS value has been met, starting with the oldest. The excess versions are kept as excess active versions if they have an un-met RETAIN DAYS value. These excess versions will then be changed to retained backup versions when a new version is created.
 2. Starting with the now-oldest backup version and ending with the *third-newest* version, DFSMSHsm calculates the age of the version to determine if the version should be expired. If a RETAIN DAYS value was specified when the version was created, then the age is compared to the retain days value. If RETAIN DAYS was not specified, then the age is compared to the value of the RETAIN DAYS EXTRA BACKUPS attribute in the management class. If the age of the version meets the expiration criteria, then the version is expired.

The *second-newest* version is treated as if it had been created on the same day as the newest backup version. Therefore, the second-newest version (newest EXTRA backup copy) is not expired until the number of retention days specified by the RETAIN DAYS value or the number of days specified in RETAIN DAYS EXTRA BACKUPS attribute in the management class have passed since the creation of the newest backup version. (The management class value is only used if RETAIN DAYS was not specified for the version).

If only one backup version exists for the data set or UNIX file, it is not deleted by the EXPIREBV.

- If the data set or zFS is deleted, the management class used is the one (if any) that was associated with the data set or zFS when the last backup version was made.

If this is the first time EXPIREBV has found this data set or zFS to be deleted:

1. DFSMSHsm compares the number of backup versions that exist with the value of the NUMBER OF BACKUPS (DATA SET DELETED) attribute in the management class. If there are more versions than requested, the excess versions are deleted if the versions do not have a RETAIN DAYS value or the RETAIN DAYS value has been met, starting with the oldest. The excess versions are kept as excess active versions if they have an unmet RETAIN DAYS value. These excess versions will then be changed to retained backup versions when a new version is created.
2. Starting with the now-oldest backup version and ending with the *third-newest version*, DFSMSHsm calculates the age of each cataloged version to determine if the version should be expired. If a RETAIN DAYS value was specified when the version was created, then the age is compared to the retain days value. If RETAIN DAYS was not specified, then the age is compared to the value of the RETAIN DAYS EXTRA BACKUPS attribute in the management class. If the age of the version meets the expiration criteria, then the version is deleted.

The *second-newest* version is treated as if it had been created on the same day as the newest backup version. Therefore, the second-newest version (newest EXTRA backup copy) will not be expired until the retention days specified by the RETAINDAYS value or the number of days specified in RETAIN DAYS EXTRA BACKUPS attribute have passed since the creation of the newest backup version. (The management class value is only used if RETAINDAYS was not specified for the version).

If the scratch date has been set by prior EXPIREBV processing:

1. DFSMSHsm compares the number of backup versions to the value of the NUMBER OF BACKUPS (DATA SET DELETED) attribute in the management class. If there are more versions than requested, the excess versions are deleted if the versions do not have a RETAINDAYS value or the RETAINDAYS value has been met, starting with the oldest. The excess versions are kept as excess active versions if they have an unmet RETAINDAYS value. These excess versions will then be changed to retained backup versions when a new version is created.
2. Starting with the now-oldest backup version and ending with the third-newest version, DFSMSHsm calculates the age of each cataloged version to determine if the version should be expired. If a RETAINDAYS value was specified when the version was created, then the age is compared to the retain days value. If RETAINDAYS was not specified, then the age is compared to the value of the RETAINDAYS EXTRA BACKUPS attribute in the management class. If the age of the version meets the expiration criteria, then the version is deleted.

The *second-newest* version is treated as if it had been created on the same day as the newest backup version. Therefore, the second-newest version (newest EXTRA backup copy) will not be expired until the retention days specified by the RETAINDAYS value or the number of days specified in RETAINDAYS EXTRA BACKUPS attribute have passed since the creation of the newest backup version. (The management class value is used only if RETAINDAYS was not specified for the version).

3. If the RETAINDAYS EXTRA BACKUP attribute is NOLIMIT and the RETAINDAYS ONLY BACKUP attribute is other than NOLIMIT, DFSMSHsm calculates the difference between the current date and the scratch date, then compares the result to the value of the RETAINDAYS ONLY BACKUP (DATA SET DELETED) attribute. If the difference between the current date and the scratch date is greater than the specified value for the attribute, DFSMSHsm deletes all of the remaining backup versions that do not have an unmet RETAINDAYS value.
 4. If all of the extra backups are deleted, DFSMSHsm calculates the difference between the current date and the scratch date and compares the result to the value of the RETAINDAYS ONLY BACKUP (DATA SET DELETED) attribute. If the difference between the current date and the scratch date is greater than the specified value for the attribute, DFSMSHsm deletes the single remaining backup version if it does not have an unmet RETAINDAYS value.
- It determines the disposition of the MCB record and the MCBR record.

If you specified the EXECUTE parameter in the EXPIREBV command, DFSMSHsm can change the MCB record in the BCDS as follows:

- If all backup versions are being deleted or no backup version existed when the EXPIREBV command first read the MCB record, DFSMSHsm deletes the MCB record (only for SMS-managed data sets).
- DFSMSHsm updates the MCB record if either of the following is true:
 - The scratch date field has been changed.
 - One or more backup versions, but not all versions, are being deleted.

If all active copies have been deleted and there are one or more retained copies, then the most recent retained copy will be moved from the MCBR record to the MCB record so that the MCB record has at least one valid backup version.

If you specified the EXECUTE parameter in the EXPIREBV command, DFSMSHsm can change or delete the MCBR record in the BCDS as follows:

- If all backup versions are being deleted or no backup version existed when the EXPIREBV command first read the MCBR record, DFSMSHsm deletes the MCBR record.

- DFSMShsm updates the MCBR record for the following:
 - One or more backup versions, but not all versions, are being deleted
 - A version is moved from the MCBR record to the MCB record.

If you specified the DISPLAY parameter for EXPIREBV, DFSMShsm updates the MCB record only if the scratch date field has been changed (updates only that field in the record).

- It deletes the eligible backup versions.

If you specified the EXECUTE parameter in the EXPIREBV command, DFSMShsm deletes a backup version as follows:

- If the backup version resides on an ML1 volume, the version is scratched (with the appropriate erase-on-scratch indication) and the MCM, MCC, and applicable UFN records are deleted. In addition, if a discrete backup profile is associated with a deleted version, the profile is also deleted.
- If the backup version resides on a non-removable DASD backup volume, the version is scratched (with the appropriate erase-on-scratch indication), and the MCC record is deleted. In addition, if a discrete backup profile is associated with a deleted version, the profile is also deleted.
- If the backup version resides on a removable DASD volume, the backup version is invalidated at this time, but it is not scratched until the backup volume is selected for spill or cleanup processing. If the erase option is in effect for the version, the MCC record is updated to reflect that the data set should be erased when it is scratched. If the erase option is not in effect for the version, the MCC record is deleted.
- If the backup version resides on a tape backup volume, the MCC record is deleted and the TTOC entry for the data set is invalidated.
- It reports on the eligibility of backup versions.

If the DISPLAY parameter was specified in the EXPIREBV command, DFSMShsm creates an output line for every backup version that is eligible to be deleted. The expired retained backup versions are appended to the end of active backup versions. The difference in the output data set between the active backup versions and the retained backup version is that the generation number of retained backup version is displayed as "****" rather than generation number.

If the EXPIREBV command specified EXECUTE, DFSMShsm sends ARC0734I messages to the backup activity log under control of the SETSYS ACTLOGMSGLVL value:

- FULL specifies that an ARC0734I message is sent for each backup version for which deletion is attempted, whether deletion is successful or not.
- EXCEPTIONONLY specifies that an ARC0734I message is sent only when a failure occurs in attempting to delete a backup version.
- REDUCED has the same effect as EXCEPTIONONLY.

If DFSMShsm is running in DEBUG mode when the EXPIREBV command is initiated, the output created is identical to the output when DFSMShsm is not in DEBUG mode, but no backup versions are deleted even if EXECUTE is specified.

Note: During the scan of the BCDS, backup versions associated with a data set name of "*prefix*.BACK" may be skipped and not expired. The prefix qualifier is set based on the BACKUPPREFIX parameter of the SETSYS command or the UID specified in the startup procedure. Because backup versions with high level qualifiers of "*prefix*.BACK" may not be expired and deleted, do not create data sets using this naming convention. Data sets not deleted during EXPIREBV processing are still eligible for deletion during rolloff of backup versions, or when a BDELETE or HBDELETE command is issued.

End the scan of the BCDS

The EXPIREBV command ends as planned when the key of the next record is higher than the planned ending key. Because the planned ending key was reached, a blank key value is saved in the backup control record (BCR) as the last processed key and the planned ending key.

The key of the last MCB record processed is saved periodically in the backup control record for this host when the host is processing an EXPIREBV command. This saved last-processed key is used to enable RESUME processing across system outages. If the EXPIREBV command ends early, the key of the last MCB record processed is saved in the backup control record for the host before processing ends. This allows processing to begin at that key if RESUME is specified on the next EXPIREBV command that is issued in that host. The EXPIREBV command can be caused to end early by:

- A HOLD command with the ALL or EXPIREBV parameters. (A RELEASE EXPIREBV or RELEASE ALL command must be issued before another EXPIREBV command can be issued.)
- A SETSYS command with the NOBACKUP or EMERGENCY parameters.
- The STOP command.

Issue an ending message

If the EXPIREBV command specified DISPLAY, DFSMSHsm sends a message to the user ID that issued the command.

If the EXPIREBV command specified EXECUTE, DFSMSHsm sends a message to the user and to the backup activity log.

Deleting backup versions of specific data sets or UNIX files

The EXPIREBV command allows you to delete all expired backup versions from all of your backup volumes. You can also delete the backup versions of specific data sets or UNIX files. The specified backup versions will be deleted even if their RETAIN DAYS value has not been met. You can delete either specific versions or all versions of the data sets or UNIX files. Use the following commands to delete backup versions:

```
BDELETE ('/filepath') ALL
BDELETE (ddsname...) -
    VERSIONS(backup_version_number)
BDELETE (ddsname...) -
    DATE(date) TIME(time)

HBDELETE (ddsname...) ALL
HBDELETE (ddsname...) -
    VERSIONS(backup_version_number)
HBDELETE (ddsname...) -
    DATE(date) TIME(time)
HBDELETE ('/filepath') ALL

hbdelete -a filename /absolutepath
hbdelete -V backup_version_numbers /absolutepath
hbdelete -d date -t time filename
```

where:

backup_version_number

Is a comma-separated list of backup versions to delete.

date

Is the date when the backup version was created.

time

Is the time when the backup version was created.

DFSMSHsm-authorized users can issue the first group of commands. Any user can issue the second and third groups of commands. The second group of commands can be issued from a TSO session. The third group of commands can be issued from a UNIX shell session.

The first form of the commands deletes all backup versions of the data set. An absolute path for the UNIX file name may be specified with single quotation marks instead of ddsname. The second form of the commands deletes only those backup versions of the data set or UNIX file for which you specify the

version numbers. The third form of the commands deletes a backup version that was taken on a specific date and time.

The last backup date field of the catalog entry is altered to a null value when all of the following apply:

- No backup versions exist.
- The data set is cataloged.
- The data set is SMS-managed.
- The data set is not migrated.

The last backup date of the UNIX file is altered to the most recent backup date in the DFSMSHsm inventory. If no more backup versions exist, then the last backup date is set to January 1, 1970.

Transferring the backup profile from one backup version to another

Discrete backup profiles are associated with only the most recent backup version of a data set. If the backup version being deleted is associated with a backup profile, DFSMSHsm reads BCDS backup version records to determine if any remaining backup versions are of a cataloged, RACF-indicated data set.

If one of the backup versions is of a cataloged, RACF-indicated data set, the backup profile is transferred to the latest backup version of a cataloged, RACF-indicated data set.

Note: Backup profiles are not transferred to a backup version of an uncataloged data set.

Profile creation

If the profile creation fails, message ARC0382I is issued, but the backup version associated with the backup profile is still deleted. If any backup profiles remain, they are not deleted.

Authorization checking during HBDELETE processing

During HBDELETE processing for data sets, DFSMSHsm uses RACF to determine the user's authorization to process the data set. First, DFSMSHsm makes an authorization check against the original data set name. If either a generic profile or a discrete profile cannot be found for a cataloged data set that was RACF-indicated at the time of backup, DFSMSHsm determines whether a backup profile exists. If a backup profile exists, DFSMSHsm makes a second authorization check with the backup profile name. If the second authorization check fails, the HBDELETE processing fails.

During HBDELETE and hbdelete processing for UNIX files, DFSMSHsm uses RACF to determine the user's authorization to the UNIX file. Using the invoker's USERID, GROUP, along with the UID, GID, and file permissions at the time of the last backup, Write Access is checked for the requested file. If the authorization check fails, the processing fails.

Command recovery and restore

You use the recovery and restore process to:

- Recover a data set that has been lost or damaged
- Recover a UNIX file that has been lost or damaged.
- Access an earlier version of the data set without deleting the current version
- Restore a data set from a dump copy
- Restore a volume from a dump copy and update it from incremental backup versions
- Restore a volume from a full-volume dump
- Recover a volume from DFSMSHsm backup versions

The recovered backup version can replace an existing data set or UNIX file, or you can rename the recovered data set copy so it exists in parallel with the current data set. You can recover or restore a data set, UNIX file, or volume only by specifying the RECOVER command.

A backup extended format VSAM data set can be recovered only as an extended format data set.

If the data set was RACF indicated at the time that the dump or backup version was created or is RACF indicated at the time of the recovery or restore, DFSMSHsm recovers a discrete profile for the recovered or restored data set. See [“Recovering or restoring RACF-indicated data sets” on page 461](#) for a discussion of profile recovery.

The permissions and attributes of the UNIX file at the time of backup are recovered

If you want to use DFSMSHsm recovery, you must have specified SETSYS BACKUP.

The recovery process has two restrictions:

- DFSMSHsm cannot recover a data set to a migration volume
- DFSMSHsm cannot recover a data set that is currently marked as migrated in the computing system catalog, *unless* the data set is non-VSAM, a recover data set name is issued with NEWNAME specified, and the NEWNAME data set is not a migrated data set.

Data set recovery and restore

You can use the following commands to recover a data set from a DFSMSHsm backup version or dump copy:

- These two forms of the command recover a specific data set when no data set with the same name is cataloged in the system catalog. If there is a data set with the same name, the recovery fails.

```
HRECOVER dsname
RECOVER dsname
```

- These two forms of the command recover a specific data set but recover it under the fully qualified data set name specified by the NEWNAME parameter that is not currently cataloged in the system catalog. The original data set name may or may not exist in the system catalog.

```
HRECOVER dsname NEWNAME(newdsname)
RECOVER dsname NEWNAME(newdsname)
```

- These two forms of the command recover a specific data set under its original name, even though a data set with the same name is cataloged. The cataloged data set is replaced by the recovered data set.

```
HRECOVER dsname REPLACE
RECOVER dsname REPLACE
```

- These two forms of the command recover a specific data set to a fully qualified data set specified by the NEWNAME parameter, even though a data set is cataloged with the same name as is specified by the NEWNAME parameter. The recovered data set replaces the cataloged NEWNAME data set.

```
HRECOVER dsname NEWNAME(newdsname) -
REPLACE
RECOVER dsname NEWNAME(newdsname) -
REPLACE
```

The HRECOVER command can be used by any user; the RECOVER command can be used by DFSMSHsm-authorized users. Three other parameters—DATE, GENERATION, and VERSION—can be specified that cause DFSMSHsm to select a backup version or dump copy other than the most recent. The GENERATION and VERSION parameters are used to select a specific backup version. The DATE parameter allows the selection of either a dump copy or a backup version.

When the RECOVER or HRECOVER command is issued in any of the shown forms, DFSMSHsm recovers the later of the most recent dump copy or the most recent incremental backup version. For a data set to be restored from a dump copy, the dump copy must have been made from a dump class that allows data set restore, and a VTOC copy must exist for the dump (except when an authorized user explicitly specifies a dump volume).

DFSMSHsm's recovery process involves five steps:

- Determine if the data set being recovered currently exists on a level 0 volume
- Select a target volume
- Determine the storage class, management class, and data class to be associated with the data set
- Determine whether to recover a backup version or a dump copy
- Recover the data set to the target volume

Determining whether the data set exists

DFSMSHsm considers the target data set to exist if the target data set name is cataloged. The target data set and target data set name refer to either the original name of the data set (if you did not specify a new name) or to the new name of the data set that you specified with the NEWNAME parameter of the RECOVER or HRECOVER command.

Selecting the target volume for incremental backup version recovery

When SMS is active, it selects the target volume for the data set to be recovered from an incremental backup version.

Selecting the target volume for physical dump copy restore

Since DFSMSDss does not invoke SMS to select the target SMS volume for a physical data set restore, DFSMSHsm selects the final target volume and passes that volume to DFSMSDss for a data set restore from a physical dump copy. The order of precedence of DFSMSHsm target volume selection is:

1. The source volume serial number of the dump copy, if FROMDUMP(DUMPVOLUME(*volser*)) is specified for a multiple-volume data set
2. The volume serial number to which the target SMS data set is currently cataloged
3. The volume serial number of the TOVOLUME parameter specified on the RECOVER command
4. The volume serial number of the FROMVOLUME parameter specified on the RECOVER command
5. The volume serial number to which the target non-SMS data set is currently cataloged
6. The source volume serial number of the dump copy, if FROMDUMP(DUMPVOLUME(*volser*)) is specified on the RECOVER command
7. The source volume serial number of the incremental backup version
8. The volume serial number to which the original data set is currently cataloged, if the NEWNAME data set is not currently cataloged and an incremental backup version is not available

If the source and selected target device types are not similar, or if the target data set and the target volume are incompatible (that is, one is SMS managed and the other is non-SMS managed), DFSMSHsm fails the restore of the data set from the dump copy. If this happens, you may be required to specify the target volume, either by cataloging the target data set or by using the TOVOLUME parameter of the RECOVER command.

Determining the storage class, management class, and data class

Determination of the SMS classes associated with the data set depends on whether SMS is active, whether the target data set exists, and whether FORCENONSMS is specified.

When SMS is not active

When SMS is not active, any data set that currently exists and is SMS-managed or that was SMS-managed when it was backed up or dumped cannot be processed unless FORCENONSMS is specified on the RECOVER command. For a discussion of using the FORCENONSMS parameter, see [“Recovering specific data sets”](#) on page 311.

When SMS is active and the target data set exists

If the target data set exists, DFSMSHsm uses the storage class, management class, and data class from the catalog entry as the SMS classes to be associated with the recovered data set.

When SMS is active and the target data set does not exist

If the target data set does not exist, DFSMSHsm invokes the ACS services of DFSMSdfp to determine the SMS classes to be associated with the recovered data set. In invoking ACS services, DFSMSHsm passes to the ACS routine of DFSMSdfp any of the following that are available:

- The name of the data set.
- The name of the storage class associated with the data set when the data set was incrementally backed up.

No storage class is passed if one of the following is true:

- The data set was not SMS-managed when it was incrementally backed up.
- A physical dump copy is being explicitly recovered because you specified FROMDUMP or FROMDUMP(DUMPVOLUME(*volser*)).
- The data set does not have an incremental backup version.

Note: If the data set was SMS-managed when it was physically dumped, and your ACS storage class routine expects DFSMSHsm RECOVER to provide the storage class in effect when the data set was dumped, the data set may be filtered as non-SMS-managed, and the recovery may fail with an “unable to recover a non-SMS-managed data set to an SMS-managed volume” error. If this occurs, use FORCENONSMS or TOVOLUME to recover the data set to a non-SMS-managed volume.

- The name of the management class associated with the data set when the data set was incrementally backed up. No management class is passed if one of the following is true:
 - The data set used the default management class when it was incrementally backed up.
 - The data set was not SMS-managed when it was incrementally backed up.
 - A physical dump copy is being explicitly recovered because you specified FROMDUMP or FROMDUMP(DUMPVOLUME(*volser*)).
 - The data set does not have an incremental backup version.
- The name of the data class associated with the data set when the data set was incrementally backed up. No data class name is passed if one of the following is true:
 - The data set had no data class when it was incrementally backed up.
 - The data set was not SMS-managed when it was incrementally backed up.
 - A physical dump copy is being explicitly recovered because you specified FROMDUMP or FROMDUMP(DUMPVOLUME(*volser*)).
 - The data set does not have an incremental backup version.
- The volume serial number specified with the TOVOLUME parameter in the request or the volume serial number and unit type of the volume where the data set originally resided.

Note: If your storage class ACS routine checks for a valid DASD unit name, be sure to include any unit names that may have existed in your installation at one time and are no longer in use, because the data set could have resided on such a unit when it was backed up.

- An environment indicator of RECOVER.
- The RACF user ID and group ID for non-DFSMSHsm-authorized users.
- The data set size in KB.
- The data set expiration date.
- A generation data group indication.
- The data set organization.

- The record organization for VSAM data sets.

ACS services returns the SMS class names to be associated with the recovered data set. If ACS services does not return a storage class name, the data set is not SMS-managed and is recovered as a non-SMS-managed data set.

Determine whether to use backup or dump versions for data set recovery

The non-DFSMSHsm-authorized user has little control over whether a restore or a recover is done. The user's ability to cause a physical data set restore rests with the options set for the dump classes and the dump volumes. Unless the user who issues a RECOVER or HRECOVER command specifies the GENERATION, VERSION, or DATE parameter, DFSMSHsm tries to recover the most recent dump copy or backup version available (based on time of day and date).

The following conditions must exist for a data set restore to be performed instead of a recover of an incremental backup version when a RECOVER or HRECOVER command is issued:

- The data set must be currently cataloged or must have been incrementally backed up by DFSMSHsm at some time. If the data set is currently cataloged and no backup version exists for it, it must have been dumped from the volume it is cataloged on.
- If FROMVOLUME is used to direct DFSMSHsm to restore a data set from a dump copy, the dump copy made from the FROMVOLUME specified is used, regardless of the catalog status of the data set when the dump copy was made. The data set is restored to the FROMVOLUME specified and left uncataloged.
- An eligible dump class must exist that allows individual data set restores (that is, the dump class must have the DATASETRESTORE attribute.)
- The data set to be restored must have resided on a volume dumped to an eligible dump copy. See [“Searching for eligible dump copies” on page 245](#) for detailed information.
- The dump volumes on which the dump copy resides must have a retention period other than NOLIMIT. DFSMSHsm assumes that the dump volumes are physically available to be mounted when the RECOVER command is issued.
- The data set must be listed in the dump VTOC copy data set associated with the dump generation in which the dump copy is contained.
- The eligible dump copy must have been created more recently than the most recent incremental backup version. If a date is specified on the HRECOVER or RECOVER command, the recover is done from the backup version or dump copy that was performed *before or on* the specified date.
- A specific generation or version must not have been requested. A generation for a data set refers only to the DFSMSHsm incremental backup versions of the data set and not to the dump copies of the source volume.
- The volume receiving the data set and the original volume from which the dump copy was made must be of similar device type.
- The volume receiving the data set and the data set being restored must be compatible in terms of being SMS-managed. DFSMSHsm will neither restore an SMS-managed data set to a non-SMS-managed volume, nor restore a non-SMS-managed data set to an SMS-managed volume.
- The version of DFSMSdss installed must be at a sufficient level to support SMS.
- The volume receiving the data set and the data set itself must both be SMS managed or non-SMS managed. DFSMSHsm does not restore an SMS-managed data set to a non-SMS-managed volume, or a non-SMS-managed data set to an SMS-managed volume.

Note:

1. DFSMSHsm always uses the backup version of SMS-managed, multiple-volume data sets unless you specify that the data set is to be restored from a dump copy.
2. See [“Restoring a data set from a DFSMSHsm dump copy” on page 251](#) for a discussion of specifying data set restore from a dump copy.
3. Only an authorized user can specify restore from a dump copy.

If any of the conditions are not met, the HRECOVER or RECOVER process continues by attempting to find a DFSMSHsm backup version to recover. If no DFSMSHsm backup version exists, the HRECOVER or RECOVER request fails.

To further identify the backup version or dump copy you want to recover, specify a date or a relative generation number for the backup version. When you identify a backup or dump version with a date, DFSMSHsm recovers the last version DFSMSHsm created on or before the date you specified. When you identify a backup version with a generation number, the number represents the number of backup versions created before the most recent one. A GENERATION(0) indicates the most recently created backup version. If DFSMSHsm requires access to a level 0 volume, daily backup volume, or spill backup volume during recovery and that volume is not online, DFSMSHsm dynamically allocates the volume. If the backup version is on more than one tape volume, DFSMSHsm allocates each required volume and the operator mounts each required volume.

Searching for eligible dump copies

If an HRECOVER command is not restricted to recovering from incremental backup volumes, or if a RECOVER command is not restricted from recovering backup versions or a specified dump volume, DFSMSHsm must determine the volume from which the data set may have been dumped, in order to find an eligible dump copy. The following conditions narrow the search:

- If the data set is currently cataloged, DFSMSHsm begins with the volume the data set is currently cataloged on as the original source volume. Dump copies of this volume are searched to try to find an eligible copy. If no eligible copy can be found, and this data set also has a valid DFSMSHsm backup version, and if this backup version was created from a different volume from the one on which the data set is currently cataloged, the search continues with dump copies of the volume last backed up from.
- If the data set is not currently cataloged, it must have a valid DFSMSHsm backup version to determine the original source volume. The volume where the data set was located, and where incremental backup last found the data set, is used to locate an eligible dump copy.
- If FROMVOLUME is used to direct DFSMSHsm to restore a data set from a dump copy, the dump copy made from the FROMVOLUME specified is used, regardless of the catalog status of the data set when the dump copy was made. The data set is restored to the FROMVOLUME specified and left uncataloged.

If an eligible dump copy is not found that is more recent than an eligible backup version, the data set is recovered from the backup version.

Recovering the data sets to SMS-managed volumes

The recovery action taken depends on whether a DFSMSHsm backup version or a dump copy is used to recover the data set, and on whether the data set being recovered is a VSAM or non-VSAM data set. If you are recovering an existing data set, you must specify either REPLACE or NEWNAME. If you specify REPLACE, DFSMSHsm replaces the existing data set with the recovered data set. If you specify NEWNAME, DFSMSHsm recovers the data set with the NEWNAME data set name.

If an out-of-space condition causes the recovery of an SMS-managed data set to fail, the RECOVER DFDSSOPTION(VOLCOUNT(N(nn) | ANY)) command can be used to ensure the successful recovery of the data set. The RECOVER DFDSSOPTION(VOLCOUNT(N(nn) | ANY)) command overrides the environment established by the SETSYS VOLCOUNT command. If you do not know what your environment is, use the QUERY SETSYS command to determine the current operating environment.

Recovering a non-VSAM data set from a DFSMSHsm backup version

If the target data set exists, DFSMSHsm deletes the existing data set. DFSMSHsm passes the storage class name associated with the data set to SMS or DFSMSdss. SMS selects the volume and allocates the target data set on that volume. DFSMSHsm or DFSMSdss then copies the backup version into the recovered data set.

The catalog status of a recovered generation data set (GDS) is dependent upon its catalog status before being recovered. If the GDS is not cataloged before recovery, its catalog status after recovery is DEFERRED. If the GDS is cataloged before recovery, its catalog status after recovery remains the same,

with the following exception: when DFSMSHsm is used as a data mover to recover a rolled-off GDS, the cataloged status of the GDS, after it is recovered, is DEFERRED. Use the IDCAMS ALTER ROLLIN command to change the status from DEFERRED to ACTIVE.

If the backup version spans multiple tape volumes, DFSMSHsm issues a message to the operator asking if all necessary tapes are available. If they are not all available, DFSMSHsm cannot recover the data set.

If DFSMSHsm requires access to a level 0 volume, a daily backup volume, or a spill backup volume during recovery and if that volume is not on line, DFSMSHsm dynamically allocates the volume. If the backup version is on more than one tape volume, DFSMSHsm allocates a unit where the operator mounts each required volume.

Recovering a VSAM data set from a DFSMSHsm backup version

If the target data set exists, DFSMSHsm deletes the existing data set, uses the Access Method Services IMPORT command or DFSMSdss as the data mover, and passes the storage class name associated with the data set. Access Method Services or DFSMSdss determines the volume to which the data set is recovered and creates space for the data set.

DFSMSHsm recovers VSAM data sets to a level 0 volume by using the Access Method Services IMPORT command or DFSMSdss as the data mover. However, because the backup version of a VSAM data set contains DFSMSHsm control information, a user-issued IMPORT command or DFSMSdss command cannot be used to recover a backup version.

If DFSMSHsm recovers a VSAM data set to a volume on a type of device that is different from the device from which DFSMSHsm backed up the VSAM data set, the allocation for the buffer that the Access Method Services IMPORT command uses could be too small. If this occurs, you can define the target VSAM data set (base cluster and all alternate indexes) and then reissue the command.

The recovery of an integrated catalog facility catalog is restricted to those users who are recognized as DFSMSHsm-authorized users. If TOVOLUME is specified, the volume specified must have the same volume serial number and device type as the volume from which the catalog was backed up. For more information about recovering the integrated catalog facility catalog, see [“Backing up and recovering a catalog”](#) on page 487.

The NEWNAME parameter limits the recovery of VSAM data sets to spheres that do not have two or more paths defined on the base cluster, and recovers only the base cluster even when alternate indexes exist. After DFSMSHsm recovers the base cluster, you can rebuild any necessary alternate indexes using the Access Method Services BLDINDEX command.

For VSAM data sets not cataloged, if the data set with the new name does not exist, DFSMSHsm generates the new names for the data, index, and path components based on the character length of the new user-supplied cluster name:

New cluster name length	Component naming scheme
1–38 characters	<i>cluster.DATA, cluster.INDEX, cluster.PATH</i>
39–42 characters	<i>cluster.D, cluster.I, cluster.P</i>
43–44 characters	<i>fourqual.Dhhmmss, fourqual.Ihhmmss, fourqual.Phhmmss</i>

Where *cluster* is the new user-supplied cluster name, *fourqual* is the first four qualifiers of the new user-supplied cluster name, and *hhmmss* is a time stamp.

If the data set with the new name already exists and the type (KSDS, ESDS, or RRDS) is the same, DFSMSHsm reuses the entry names of the data component and index component. For information about how DFSMSdss generates new names for the data and index components, refer to the [z/OS DFSMSdss Storage Administration](#).

A message is printed containing the generated or existing name.

If you specified the NEWNAME parameter and the REPLACE parameter and if a data set with the name you specified already exists, DFSMSHsm deletes the existing data set before it recovers the data set you specified with the RECOVER command.

The new data set name and the old data set name can be cataloged in the same or different catalogs when you specify the NEWNAME parameter.

For VSAM data sets not cataloged, if you specify NEWNAME the data and index objects of the data set being recovered must be the same as the object names exported. The object name of the data set being recovered is passed to Access Method Services IMPORT to rename the NEWNAME data set. Access Method Services verifies if the name is the same as the one exported.

Restoring a non-VSAM data set from a dump copy

If the target data set exists, DFSMSHsm deletes the data set. DFSMSHsm invokes DFSMSdss with the storage class name and management class name associated with the data set. DFSMSHsm also passes the target volume serial number selected by the criteria mentioned in “[Selecting the target volume for physical dump copy restore](#)” on page 242. With the information that DFSMSHsm passed to it, DFSMSdss then restores the data set to that volume.

Note: Use the FORCENONSMS keyword to recover the data set to a non-SMS managed volume.

If the target data set does not exist, DFSMSHsm invokes DFSMSdss with the storage class name and management class name associated with the data set when it was incrementally backed up. If the data set does not have an incremental backup version, neither the storage class name nor the management class name is passed to DFSMSdss, and the data set may be restored as non-SMS-managed.

If you specified the NEWNAME parameter in the command and a data set with that name exists, DFSMSHsm deletes the existing data set. DFSMSHsm then invokes DFSMSdss to restore the NEWNAME data set. DFSMSdss restores the data set to the volume passed by DFSMSHsm. If the restore is successful, DFSMSHsm catalogs the restored data set.

Restoring a VSAM data set from a dump copy

If the data set does not exist on a level 0 volume, you can create it. If a backup version of the data set exists, you need not create the data set. DFSMSHsm performs a recovery of the backup version (which creates the data set) then invokes DFSMSdss to restore the data set. If you are a DFSMSHsm-authorized user, DFSMSHsm still attempts to restore the data set even if it is not cataloged and no backup version can be found. The restored data set is not cataloged. In this case, you must use the Access Method Services DEFINE RECATALOG command to catalog the data set again. If a new name was specified, it is passed to DFSMSdss. If you are not a DFSMSHsm-authorized user, the recovery fails and DFSMSHsm issues an error message if the data set is not allocated and no backup version can be found.

If the data set is cataloged, DFSMSHsm invokes DFSMSdss, specifying REPLACE, and passes the management class name and storage class name. DFSMSdss attempts to restore the data set over the top of the existing data set. If the space is too small, DFSMSdss fails the request, and DFSMSHsm writes the DFSMSdss error message in the dump activity log and issues an error message to the user.

If you specify NEWNAME and the NEWNAME data set is cataloged when the original data set is not cataloged, DFSMSHsm first recovers the backup version of the original data set, using the storage class and management class appropriate to the NEWNAME data set. DFSMSHsm then restores the dump copy of the data set over the top of the recovered data set. Finally, DFSMSHsm deletes the cataloged NEWNAME data set and renames the restored data set to the new name.

If you specify NEWNAME and the NEWNAME data set is not cataloged when the original data set is cataloged, DFSMSHsm invokes DFSMSdss to restore the dump copy of the data set passing the NEWNAME. DFSMSHsm then attempts to catalog the restored new target data set name when it is a single-volume data set. In the case when it is part of a multi-volume data set, use the Access Method Services DEFINE RECATALOG command to catalog the data set.

Recovery of direct access data sets

The way to access data after a BDAM data set is recovered is an important consideration when recovering a data set to a volume with a different track length than the level 0 volume from which it was backed up. Using the DAOPTION parameter of the RECOVER command allows you to specify how the recovered data set is to be accessed (relative track or relative block). For instance, using the RELBLK option of DAOPTION allows a data set to be recovered to a device type with a smaller track size than the original level 0 volume. In this case, the data is moved to "fill out the track" rather than a "track to track" image as is the case for the RELTRK and SAMETRK options.

The DAOPTION parameter can only be used when recovering a data set as non-SMS-managed. The FORCENONSMS parameter can be used to recover a data set that would otherwise be returned as SMS managed.

Note: If the BDAM data set is to be recovered from a physical volume dump copy, the device types must be the same. Consequently, DAOPTION (RELBLK) cannot be specified.

Recovery of extended format data sets

DFSMSHsm recovers or restores an extended format data set (as indicated in the MCC record), using DFSMSDss as the data mover, as follows:

- If FROMDUMP(DUMPVOLUME(*volser*)) is not specified, DFSMSHsm always recovers an incremental backup version of a multiple-stripe extended format data set, even if a physical dump copy is more recent than the incremental backup version.
- If FROMDUMP(DUMPVOLUME(*volser*)) is not specified, DFSMSHsm fails the recovery of a multiple-stripe extended format data set if the data set has only been backed up by a physical full-volume dump.
- An extended format data set that had been multiple-stripe when it was backed up or dumped is recovered or restored as a single stripe or multiple-stripe extended format data set, regardless of the existence of the target data set before the RECOVER command is issued. The number of stripes depends on the storage class SUSTAINED DATA RATE value, and the availability of the required volumes.
- An extended format data set that had been single stripe when it was backed up or dumped is recovered or restored as a single stripe or multiple-stripe extended format data set, regardless of the existence of the target data set before the RECOVER command is issued. The number of stripes depends on the storage class SUSTAINED DATA RATE value, and the availability of the required volumes.
- A data set that was extended format when backed up or dumped is recovered or restored as a nonextended format but still SMS-managed data set if no SMS volumes supporting extended format data sets are available.
- If the ACS routine is modified to direct an extended format data set to a non-SMS-managed volume, or if no SMS-managed DASD volumes attached to an ESCON 3990 Model 3 are available, DFSMSHsm fails the recover of the data set.
- A data set that was not extended format when it was backed up is always recovered or restored as a data set that is not extended format. DFSMSHsm never converts a data set that is not extended format to an extended format data set through recover or restore processing, even if the data set is preallocated as an extended format data set before the RECOVER command is issued.
- If FORCENONSMS is specified on the RECOVER command to force an extended format data set to be recovered or restored to a non-SMS-managed volume, DFSMSHsm recovers or restores the data set to a non-SMS-managed volume, but the data set becomes a data set that is not extended format.
- If FROMVOLUME is specified on the RECOVER command to recover an extended format data set, DFSMSHsm fails the recover or restore.
- If reblocking is required for an extended format data set, DFSMSHsm performs the reblocking.

To restore a single stripe extended format data set from a physical dump copy on a specific dump tape, you can issue:

RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*))

To restore an existing multiple-stripe extended format data set from a physical dump tape, you must issue the `RECOVER dsname FROMDUMP(DUMPVOLUME(volser))` command for each stripe of the data set. If the extended format data set no longer exists, you must catalog the data set after the restore operations. DFSMSHsm and DFSMSdss do not catalog a multiple-stripe extended format data set during physical data set restore processing.

Last-backup date after recovery

After DFSMSHsm recovers or restores a data set, the settings for the last-backup date are as follows:

- If a new name has been provided for the data set, the last-backup date is set to zero.
- If a new name has not been provided for a non-VSAM data set and the data set was recovered from a backup version, the last-backup date is set to the backup date of the version being recovered.
- If a new name has not been provided for a VSAM data set and the data set was recovered from a backup version, the last-backup date is the value set by `IMPORT` or `DFSMSdss`, whichever recovered the data set.
- If a new name has not been provided and the data set had been restored from a dump copy, the last-backup date is the value restored from the dump copy.

Reblocking during recovery

The purpose of reblocking is to make the most efficient use of available space. If you use DFSMSHsm as the data mover and specify the `CONVERSION` parameter of the `SETSYS` command, DFSMSHsm can reblock some types of record organizations of physical sequential data sets during recovery.

If you use DFSMSdss as the data mover, specify `CONVERSION(REBLOCKTOANY)` to reblock physical sequential and partitioned data sets.

Resetting the date-of-scratch field

During a recovery or restore, DFSMSHsm checks the date-of-scratch field in the MCB record if `NEWNAME` is not specified. If a date-of-scratch exists, DFSMSHsm removes the date.

Tape mount optimization during recovery

The tape mount optimization for data set recoveries is accomplished by recovering all queued or soon arriving data set requests from a single and common backup tape with a single tape mount. For media before 3480s, those requests from a single and common backup tape are processed in ascending file-block ID. For 3480 (single-file-format tape cartridges) and later media, those requests from a single and common backup tape are processed in priority order on the recovery queue. The tape mount optimization is not applied to data set requests from dump tapes or when a data set spans backup tapes.

Processing priority of data set recoveries

Programming Interface Information

All data set recovery requests are queued to and processed from a queue of management work elements called the recovery queue (separate from the volume recovery queue). DFSMSHsm gives all `WAIT` requests a higher priority on the queue than `NOWAIT` requests.

You can change the priorities of these requests as they are put on the queue by using the return-priority installation exit `ARCRPEXT`. `ARCRPEXT` allows you to assign a relative queuing priority such that `WAIT` requests can be prioritized relative to other `WAIT` requests, and `NOWAIT` requests prioritized relative to other `NOWAIT` requests. (In the case of a `NOWAIT` request initiated from a volume recover request, the queuing priority is initialized with the value propagated from the volume recover request passed earlier to the exit.) See *z/OS DFSMS Installation Exits* for a description of `ARCRPEXT`.

End Programming Interface Information

UNIX file recovery

You can use the following commands to recover a UNIX file from a DFSMSHsm backup version:

- These three forms of the command recover a specific UNIX file when no existing file has the same absolute path. If there is a UNIX file with the same absolute path, the recovery fails.

```
HRECOVER 'absolute_pathname'
RECOVER 'absolute_pathname'
hrecover filename absolute_pathname
```

- These three forms of the command recover a specific UNIX file, even though an existing file with the same absolute path exists. The UNIX file is replaced by the recovered UNIX file.

```
HRECOVER 'absolute_pathname' REPLACE
RECOVER 'absolute_pathname' REPLACE
hrecover -f filename absolute_pathname
```

- These three forms of the command recover all files and empty directories within a directory

```
HRECOVER '/directory_path/'
RECOVER '/directory_path/'
hrecover directory/
HRECOVER '/directory_path/*'
RECOVER '/directory_path/*'
hrecover "directory/*"
```

- These three forms of the command recover all files and directories in a sub-tree.

```
HRECOVER '/directory_path/' RECURSE
RECOVER '/directory_path/' RECURSE
hrecover -R directory/
```

The HRECOVER TSO command and hrecover UNIX shell command can be used by any user. The RECOVER command can be used by DFSMSHsm-authorized users. Three other parameters—DATE, GENERATION, and VERSION—can be specified that cause DFSMSHsm to select a backup version or dump copy other than the most recent. The GENERATION, VERSION, and DATE/TIME parameters are used to select a specific backup version. The RECURSE parameter allows sub-directories to be examined for versions to recover. Because other filesystems may be mounted under a directory, the RECURSE parameter allows specification of NOCROSSMOUNTS to indicate that the other filesystem should not be entered when looking for files to recover. The default setting of RECURSE is NOCROSSMOUNTS, which indicates it is not acceptable to enter other filesystems when looking for files to recover.

The commands shown above result in DFSMSHsm recovering the most recent backup version.

DFSMSHsm's recovery process for a single UNIX file involves the following:

- Determine if a UNIX file exists with the same absolute path as the UNIX file being recovered
- Recover the UNIX file, recreating any necessary directories.

UNIX file recovery wildcard processing

To allow recovery of multiple UNIX files in a single request, DFSMSHsm supports the limited use of the wildcards, asterisk (*) and question mark (?). The request pathname passed to DFSMSHsm is considered to be made up of two parts; a directory path from root, and a filename pattern.

- directory path - starting from root (/), a series of directory names separated by forward slash, ending in a forward slash. e.g. /dir/dir2/dir3/
- filename pattern - a series of characters that should be used to match with filenames in the DFSMSHsm inventory under a particular directory. The asterisk (*) character may be used once within the pattern

to indicate 0 or more characters may match, and optionally at the end of the pattern to indicate any number of trailing characters may match. The question mark (?) character may be used multiple times within the pattern to indicate that any single character is acceptable in that place. If the requested filename should include the asterisk, question mark, or backslash character, a preceding backslash character must be used as an escape character. e.g. file?5*.txt , file\?5*.txt

DFSMSHsm's recovery process for multiple files involves the following:

- Examine the request pathname. If it ends in a forward slash (/), or a wildcard is found in the pattern portion, it examines the DFSMSHsm inventory for matching names to recover. Each directory that is encountered, is additionally examined for matching names to recover.
- Create an internal request to process a specific file.
- Because it is possible that an absolute pathname can be for a UNIX file at one backup time, and a directory at other backup times, DFSMSHsm will stop processing that sub-tree of backups as it is unable to determine which version is required (file or directory).

Last-backup date after recovery

After DFSMSHsm recovers a UNIX file, the last-backup date of the file is set to the date of the backup version that was recovered.

Resetting the date-of-scratch field

During a recovery, DFSMSHsm checks the date-of-scratch field in the MCB record. If a date-of-scratch exists, DFSMSHsm removes the date.

Tape mount optimization during recovery

The tape mount optimization for data set recoveries is accomplished by recovering all queued or soon arriving data set requests from a single and common backup tape with a single tape mount. For media before 3480s, those requests from a single and common backup tape are processed in ascending file-block ID. For 3480 (single-file-format tape cartridges) and later media, those requests from a single and common backup tape are processed in priority order on the recovery queue. The tape mount optimization is not applied to data set requests from dump tapes or when a data set or a UNIX file backup spans backup tapes.

Processing priority of UNIX file recoveries

Programming Interface Information

All Unix file recovery requests are queued to and processed from a queue of management work elements called the recovery queue (separate from the volume recovery queue). DFSMSHsm gives all WAIT requests a higher priority on the queue than NOWAIT requests. DFSMSHsm gives all single UNIX file requests a higher priority than the internal requests that are generated from a directory recover request.

You can change the priorities of these requests as they are put on the queue by using the return-priority installation exit ARCRPEXT. ARCRPEXT allows you to assign a relative queuing priority such that WAIT requests can be prioritized relative to other WAIT requests, and NOWAIT requests prioritized relative to other NOWAIT requests. (In the case of a NOWAIT request initiated from a volume recover request, the queuing priority is initialized with the value propagated from the volume recover request passed earlier to the exit.) See z/OS DFSMS Installation Exits for a description of ARCRPEXT.

End Programming Interface Information

Restoring a data set from a DFSMSHsm dump copy

A DFSMSHsm-authorized user can specifically request that a data set be restored from a dump copy rather than recovered from a backup version. The following commands specifically request that a data set be restored from a dump copy:

- RECOVER *dsname* FROMDUMP

This form of the command restores the data set from the most recent dump copy. If a data set is cataloged with the same name as the data set being restored, the restore fails.

- RECOVER *dsname* FROMDUMP(DUMPCLASS(*class*))

This form of the command restores the data set from the most recent dump copy made to the specified dump class. If a data set is cataloged with the same name as the data set being restored, the restore fails.

- RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*))

This form of the command restores the data set from a specific dump volume. The volume might not be the most recent dump copy. If a data set is cataloged with the same name as the data set being restored, the restore fails.

- RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) REPLACE

This form of the command restores the data set from a specific dump volume. The restored data set replaces the cataloged data set having the same name.

- RECOVER *dsname* TOVOLUME(*volser*) UNIT(*unittype*) - FROMDUMP(DUMPGENERATION(*dgennum*))

This form of the command restores the data set to the specified volume from a specified generation of a dump volume.

- RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - UNIT(*unittype*) TOVOLUME(*volser*)

This form of the command restores the data set from a specific dump volume. This command must be used to restore a data set from a former device type when a new volume of a similar device type—but having the same volume serial number—resides on the system (for example, a 3380 volume is replaced with a 3390 volume, the same volume serial number is retained, and the 3390 volume is running in a 3380 emulation mode). The recovery is supported only when the target device type is similar to the source device type from which the dump copy was made.

The FROMDUMP parameter of the RECOVER command directs that the data set be restored from a dump copy. If the dump copy starts in file 2 or beyond on a stacked dump tape, DFSMSHsm supplies the file block ID for high-speed positioning to the start of the dump copy.

You can also specify a DATE parameter to direct DFSMSHsm to restore the data set from a dump made on or before the specified day.

Non-VSAM data sets and non-key-range VSAM data sets cataloged in an integrated catalog facility catalog that can be restored within the restrictions of DFSMSdss can be restored by DFSMSHsm. Key-range VSAM data sets can be recovered from backup versions, but cannot be restored from dump volumes. The DFSMSHsm default for a dump class does not allow physical data set restores from dump volumes. You can allow DFSMSHsm to perform restores by specifying the DATASETRESTORE parameter of the DEFINE DUMPCLASS command.

During the data set restore, the dump volumes must be mounted in the same order that they were written during the full-volume dump. The dump volumes are read until the desired data set is found. For a dump copy stacked on a dump tape, with its own unique data set name, the search for the data set starts in the file containing the dump copy. If the dump volume has more than one valid dump copy, the desired data set is not cataloged or is migrated, and the data set has no backup copy, you must use the SOURCEVOLUME(*volser*) parameter of the FROMDUMP parameter to tell DFSMSHsm which dump copy to use.

To restore an SMS-managed, multivolume, non-striped data set from a full-volume dump copy, you must recover each segment of the data set to a non-SMS-managed volume. This is accomplished by issuing the RECOVER command with the FORCENONSMS parameter for each segment of the data set (you can use the TOVOLUME parameter to direct the recovery of the data set to a specific volume). After recovering each segment of the data set, you can catalog the data set as a multivolume, non-SMS data set using IDCAMS. To make the data set SMS-managed, you can copy or move the data set to SMS-managed volumes. The restoration of the data set is complete after the data set is cataloged. The data set is not cataloged by the recovery operation.

You can use the LIST command to ensure that a dump copy contains the data set to be restored. For example, you may need the list if you are not running the incremental backup function and the data set is no longer cataloged.

To request that a data set be *specifically restored* from a dump copy, the following conditions must be satisfied:

- The volume receiving the data set and the original volume from which the dump copy was made must be of similar device type.
- The volume receiving the data set and the data set being restored must be compatible in terms of being SMS-managed. (You can override this restriction by using the FORCENONSMS parameter to restore an SMS-managed data set to a non-SMS-managed volume, and the TOVOLUME parameter to direct recovery to a specific volume.)
- The FROMDUMP parameter must be specified.
- If the DUMPVOLUME subparameter is not specified, the data set must be currently cataloged or must have been incrementally backed up by DFSMSHsm at some point in time. There must also be a dump copy of the original source volume in the specified class.

If the DUMPCLASS subparameter or the DUMPVOLUME subparameter is specified with the RECOVER command, DFSMSHsm does not enforce the DATASETRESTORE subparameter or the permanent retention restrictions of the DEFINE command for the dump class that the dump copy is associated with. Dump volume availability is the responsibility of the authorized user in these cases; therefore, be careful not to request a data set restore from a dump volume that is not in the on-site tape library.

Rules: If neither the DUMPCLASS subparameter nor the DUMPVOLUME subparameter is specified with the FROMDUMP parameter, then the following conditions *must* be met or the command fails:

- The data set must be currently cataloged or must have been incrementally backed up by DFSMSHsm at some time. (This condition is also necessary when DUMPCLASS is specified.)
- An eligible dump class must exist that allows individual data set restores. For example, the dump class must have the DATASETRESTORE attribute defined for the class.
- The data set to be restored must have resided on a volume dumped to an eligible dump copy, as described by [“Searching for eligible dump copies”](#) on page 245.
- The dump volumes on which the dump copy resides must have a retention period other than NOLIMIT. DFSMSHsm assumes the dump volumes are physically available to be mounted when the RECOVER command is issued.
- The data set must be listed in the dump VTOC copy data set associated with the dump generation in which the dump copy is contained. This condition must also be met even if DUMPCLASS is specified.

Restoring VSAM data sets

The restoration of a VSAM data set does not recatalog the data set. However, DFSMSHsm can recatalog a VSAM data set if a backup version of the data set exists. To recatalog the restored data set, DFSMSHsm first recovers the backup copy. The recovery of the backup copy recatalogs the data set. After recovering the backup version of the VSAM data set, DFSMSHsm invokes DFSMSdss to restore the data set by replacing the version just recovered.

If a backup version of the data set does not exist or the attempt to recover the backup version fails, the VSAM data set that is restored to the target volume is uncataloged. A warning message is issued to alert you to your need to recatalog the data set. When a VSAM data set does not have a DFSMSHsm backup version or has not been CATALOGED on the target volume, you must recatalog the data set by using the Access Method Services DEFINE command with the RECATALOG option. DFSMSHsm does not recatalog a VSAM data set after a restore. When a restored VSAM data set replaces an existing data set or a recovered backup version, you should ensure that the catalog records reflect the desired conditions.

If DFSMSHsm successfully recovered a backup version but the restore failed (for example, the allocated space or the control interval was too small), you should delete the recovered data set. After deleting the recovered data set, you should:

1. Predefine the data set on the target volume, using the proper allocation and control interval sizes
2. Reissue the RECOVER command to request the restore

If the VSAM data set exists on the target volume when the recovery request is made, neither DFSMSHsm nor DFSMSdss alters any preexisting catalog information for the data set. You are responsible for ensuring that the catalog information is correct.

No catalog entries are created or modified for these cases. DFSMSdss does restore the VSAM volume data set catalog entry. The data set cannot be an integrated catalog facility catalog. DFSMSHsm processing causes the catalog to be opened if it is not already opened. DFSMSdss does not restore an open integrated catalog facility catalog. If you need to restore the catalog from a dump copy, refer to [“Restoring data sets or volumes directly with DFSMSdss” on page 458](#).

Rule: The original VSAM data set must have been backed up by DFSMSHsm backup for the recover function to operate correctly.

Managing volume backups with fast replication

Fast replication is a function that uses volume-level fast replication to create backup versions for sets of storage groups. You define a set of storage groups with the SMS "copy pool" construct. Fast replication target volumes contain the fast replication backup copies of DFSMSHsm-managed volumes. Fast replication target volumes are defined with the SMS "copy pool backup" storage group type. The fast replication backup versions on DASD can then be dumped to tape using either the FRBACKUP command or with Automatic Dump.

Recovery from the fast replication backup versions can be performed at the data set, volume, or copy pool level. The entire copy pool, individual volumes and data sets within a copy pool, can be recovered from the fast replication backup versions on DASD or tape. Individual data sets are recovered to the volume or volumes that they existed on at the time of backup.

The fast replication function enables the backup and recovery of a large set of volumes to occur within a small time frame. The fast replication function supports the FlashCopy and SnapShot functions.

DFSMSHsm supports the following FlashCopy functions:

COPY and NOCOPY modes

A FlashCopy relationship is established in either COPY or NOCOPY mode. COPY mode results in a background copy which causes all tracks in the FlashCopy relationship to be physically copied from the source volume tracks to the target volume tracks. When the background copy operation completes, the FlashCopy relationship ends. When you define a copy pool with a VERSIONS value of 1 to 85, a COPY mode FlashCopy relationship is established.

NOCOPY mode suppresses the background copy and causes the relationship to be established without immediately initiating a background copy. When the device receives an update to a source track in a FlashCopy relationship, a copy of the point-in-time (pre-update) data is preserved on the target volume. A FlashCopy relationship established in NOCOPY mode remains active until the FlashCopy relationship is withdrawn, or all specified source device tracks or a threshold number of source tracks are updated. When you define a copy pool with a VERSIONS value of 0 (zero), a NOCOPY mode FlashCopy relationship is established.

Incremental FlashCopy

Incremental FlashCopy provides the capability to refresh a volume in a FlashCopy relationship by copying the changed data only. It reduces background copy time when only a subset of data has changed. An incremental FlashCopy relationship persists until it is withdrawn. DFSMSHsm supports incremental FlashCopy in COPY mode.

For more information, see [“Creating a copy pool incremental version” on page 267](#) and [“Recovering from a copy pool incremental version” on page 279](#).

Preserve mirror operation

DFSMSHsm allows the target of the FlashCopy operation to be a Peer-to-Peer Remote Copy (PPRC) primary device. When the tracks associated with the FlashCopy relationship are copied to the PPRC secondary device, the PPRC (Metro Mirror) pair goes into a duplex pending state, to ensure the integrity of the mirror between the local site and the remote site. When the FlashCopy operation completes, the Metro Mirror volume pair returns to full duplex state.

To avoid placing the Metro Mirror pair in an unsynchronized state, IBM Remote Pair FlashCopy, also known as preserve mirror operation, mirrors the FlashCopy command that is issued at the local site, to the remote site, if proper configuration exists. Preserve mirror operation preserves the Metro Mirror duplex state and at the same time ensures that the Metro Mirror remote site is a true mirror of the local site.

For more information, see [“Using Metro Mirror primary volume during fast replication backup” on page 267](#) and [“Using Metro Mirror primary volume during fast replication recovery” on page 279](#).

Space efficient FlashCopy

Space efficient FlashCopy refers to a FlashCopy relationship for which the target volume is a space efficient volume. With space efficient FlashCopy, the amount of physical space consumed by the target volume is limited to the minimum amount of space required to maintain the copy. Space efficient FlashCopy is supported in NOCOPY mode only and cannot be used in combination with preserve mirror operation.

Fast reverse restore

Fast reverse restore provides the capability to reverse the direction of an existing FlashCopy relationship and restore the source volume to the point-in-time state when it was last flashed to the target without waiting for the background copy to complete. Once a fast reverse restore has completed, the contents of the backup volume (the original FlashCopy target) become invalid. When using fast reverse restore, it is best to keep dump tape copies in addition to DASD copies.

DFSMSHsm supports fast reverse restore of a copy pool in both COPY and NOCOPY modes. However, fast reverse restore cannot be used in combination with the preserve mirror operation. When the storage subsystem supports cascaded FlashCopy function, the additional FlashCopy targets will not need to be removed prior to attempting fast reverse restore. Ensure the correct backup version is specified on the command you intend to use for fast reverse restore.

For more information, see [“Recovering a copy pool using fast reverse restore” on page 278](#).

FlashCopy consistency group

When logically related data spans across multiple volumes, FlashCopy consistency group can be used to create consistent copies while minimizing application impact.

To accomplish this, the process consists of freezing the source volumes as each volume pair in the copy pool is copied using FlashCopy, and thawing all the frozen volumes using the Consistency Group Created command after a consistent copy pool backup version has been created. When the FlashCopy freeze option is used, the control unit puts the source volume in an extended long busy (ELB) state causing I/O activities to be held at the host. During the time window starting when the first volume is frozen until the volumes are thawed, no dependent-write updates will take place, which allows a consistent backup version to be created.

The FlashCopy consistency group function can be used in combination with other FlashCopy options.

Related reading

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

Combining fast replication options in DFSMSHsm

“Combining fast replication options in DFSMSHsm” on page 256 shows the current allowable fast replication options in the DFSMSHsm environment. Note, however, that IBM System Storage DS8000® series might permit different combinations of FlashCopy options.

Table 27. Allowable combinations of fast replication options

Can FR option → be used with an FR operation or option ↓?	Multiple (up to 12) FlashCopy targets	COPY mode (Copy pool VERSIONS = 1 to 85)	NOCOPY mode (Copy Pool VERSIONS = 0)	Incremental FlashCopy	Preserve Mirror	FlashCopy SE	Fast reverse restore	FlashCopy consistency group
FRBACKUP COPYPOOL	Yes	Yes	Yes	Yes	Yes	Yes	Yes (1)	Yes
FRRECOV COPYPOOL	No	Yes	Yes/No (2)	Yes	Yes	Yes (2)	Yes	No
FRRECOV TOVOLUME	No	Yes	No (4)	Yes	Yes	No (4)	No	No
FRRECOV DSNAME	Yes	Yes	No (4)	No (5)	Yes	No (4)	No	No
Multiple (up to 12) FlashCopy targets	–	Yes	Yes (3)	Yes (6)	Yes	Yes (3)	Yes (7)	Yes
COPY mode	Yes	–	No	Yes	Yes	No	Yes	Yes
NOCOPY mode	Yes (3)	No	–	No	Yes	Yes	Yes	Yes
Incremental FlashCopy	Yes (6)	Yes	No	–	Yes	No	Yes	Yes
Preserve Mirror	Yes	Yes	Yes	Yes	–	No	No	Yes
FlashCopy SE	Yes (3)	No	Yes	No	No	–	Yes	Yes
Fast reverse restore	Yes (7)	Yes	Yes	Yes	No	Yes	–	Yes (8)
FlashCopy consistency group	Yes	Yes	Yes	Yes	Yes	Yes	Yes (8)	–

Notes:

1. When a copy pool is defined to allow for fast reverse restore, its copy pool backup versions will be created with fast reverse restore capability.
2. DFSMSHsm FR copy pool recovery from NOCOPY or space efficient backup volumes can only be used with fast reverse restore. When the NOCOPY version is defined without fast reverse restore capability, the FRRECOV copy pool operation can be performed from dump tapes.
3. A DASD backup version created in NOCOPY mode without fast reverse restore capability is withdrawn when the dump tape has been created. A NOCOPY DASD backup version with fast reverse restore capability can be used for recovery and is kept until it is withdrawn, deleted, or replaced.
4. FRRECOV in VERSIONS=0 environment can be performed FROMDUMP.
5. If there is an existing incremental or persistent relationship, the FRRECOV data set recovery command can be performed using standard I/O. The FASTREPLICATION (PREFERRED | NONE) parameter can be specified to override the SETSYS FASTREPLICATION(DATASETRECOVERY) specification. You can also specify FROMDUMP to recover from dump tapes.
6. When the software and microcode requirements are met, Incremental FlashCopy requests result in Incremental FlashCopy Version 2 (V2), which allows a volume to have more than one incremental relationship. With Incremental FlashCopy Version 1 (V1), a FlashCopy source can have only one incremental target.

7. Multiple backup copies with fast reverse restore capability can be created. When the storage subsystem supports cascaded FlashCopy function, additional FlashCopy targets do not need to be removed before using fast reverse restore. Otherwise, only one relationship can exist when fast reverse restore is used; all additional relationships must be removed before fast reverse restore is initiated or the FRR operation will fail.
8. A copy pool can be defined with either or both the FlashCopy consistency group option and fast reverse restore capability. In the case where both FlashCopy consistency group = Yes and fast reverse restore = Yes are designated, the copy pool will be backed up using the FlashCopy consistency group option; the copy pool backup DASD version will have fast reverse restore capability. When FRRECOV COPYPOOL FROMDASD is performed, fast reverse restore will be used. Consistency groups will not be used in any FRRECOV operation.

Combining copy services operations in DFSMS

Hardware platforms support different copy services combinations. For information about which copy services combinations are allowed on the same device in a DFSMS environment, see [Combining copy services operations in z/OS DFSMS Advanced Copy Services](#).

Using the copy pool construct for fast replication setup

The SMS copy pool construct contains the names of up to 256 storage groups that can be processed collectively for fast replication operations.

When you define a copy pool through ISMF, the individual source pool storage group names are recorded. The actual volumes that are associated with each storage group are retrieved during subsequent processing. Consequently, the volumes that are processed during a function, such as backup, might not be the same volumes that are associated with the storage group at the time that the copy pool was established.

Example: If one or more volumes are added to a storage group within a copy pool, those volumes are automatically processed as part of the next backup. You are not required to update the copy pool definition unless you want to add or remove storage groups from the pool.

Restriction: An individual storage group can be associated with more than one copy pool, but with no more than 50. When using FlashCopy version 1 as the fast replication utility, do not associate a storage group with multiple copy pools. This is because each individual copy pool, with its associated background copies for all the volumes in the common storage group, must process completely before the next copy pool can be processed. This processing can take several hours and prevents the volumes in storage groups associated with a single copy pool from having a timely backup created. FlashCopy version 2 and SnapShot do not have this restriction.

You can maintain as many as 85 backup versions for each copy pool. Each version can have a DASD copy, a dump tape copy, or both. The number of backup versions kept on DASD is specified on the SMS copy pool definition panel. Each DASD maintained copy requires a unique target volume for each source volume. If you specify five dedicated versions, there must be five target volumes that are available for each source volume that is associated with the copy pool. All target volumes must be available when you issue the FRBACKUP COPYPOOL(*cpname*) PREPARE command. If the PREPARE function is not performed, the target volumes need not be available until the actual backup version is created. The number of tape versions that are maintained depends on the expiration settings of the dump classes used to create the individual tape copies.

In the COPY POOL DEFINE and COPY POOL ALTER panels, you can use the **Number of DASD Fast Replication Backup Versions with Background Copy**, **FRBACKUP to PPRC primary Volumes allowed**, **FRRECOV to PPRC primary Volumes allowed**, **Capture Catalog Information for Data Set Recovery**, **Allow Fast Reverse Restore**, and other fields to tailor your copy pool definition.

You can use the **Number of DASD Fast Replication Backup Versions with Background Copy** field to specify the desired number of DASD fast replication backup versions with background copy between 0 and 85. A value of 0 causes DFSMSHsm to use NOCOPY mode when creating the DASD backup copy. When the copy pool is defined with fast reverse restore capability, a DASD backup copy created in

NOCOPY mode can be used for recovery and as a backup copy from which to create a dump tape copy. The DASD backup copy is kept until it is withdrawn or deleted. When the copy pool is defined without fast reverse restore capability, it is only used to create a point-in-time backup copy from which a dump tape copy is created. Once the tape copy is successfully created, the DASD backup copy is deleted. Values between 1 and 85 cause DFSMSHsm to use COPY mode. A COPY mode environment allows recovery of the latest versions from DASD or dump tapes. For example, a COPY mode environment with 2 versions specified and all versions dumped, the latest two versions are on DASD and tape, while the older versions (up to 83) are only available as dump tape copies. When a new DASD version is created, it replaces an existing DASD version. Specifying two copies guarantees that one backup copy is always available on DASD.

You can specify whether DFSMSHsm uses Metro Mirror and Global Mirror primary volumes as targets of the FRBACKUP and the FRRECOV commands in the **FRBACKUP to PPRC primary Volumes allowed** and **FRRECOV to PPRC primary Volumes allowed** fields. The following are valid values and their meaning for the **FRBACKUP to PPRC primary Volumes allowed** and **FRRECOV to PPRC primary Volumes allowed** fields:

NO

Metro Mirror and Global Mirror primary volumes cannot be targets.

PN

Metro Mirror and Global Mirror primary volumes can be used as FlashCopy target volumes. Configuration requirements for preserve mirror operation are not considered.

PP

Metro Mirror and Global Mirror primary volumes can be FlashCopy target volumes. If the target volume is a Metro Mirror primary device, a preserve mirror operation is preferred.

The IBM DS8880 series and newer IBM storage systems do not support the Preserve Mirror Preferred option (PP) for Metro Mirror. If you are using one of those storage systems, and you want to preserve your disk mirror or copy, specify PR.

PR

Metro Mirror and Global Mirror primary volumes can be FlashCopy target volumes. If the FlashCopy target is a Metro Mirror primary volume, a preserve mirror operation is required.

[blank]

None specified. This is the default.

Note: You can use the ALLOWPPRC keyword on the FRRECOV command to override the SMS copy pool PPRC primary volumes allowed settings.

You can specify whether DFSMSHsm uses FlashCopy consistency groups to create consistent copy pool backup copies in the **FlashCopy Consistency Group** field. If you specify Y (yes), the FlashCopy consistency group option will be used and consistent copies are required. If you specify N (no) or leave the field as blank, the FlashCopy consistency group option will not be used.

You can specify whether DFSMSHsm uses FlashCopy consistency group to create consistent copy pool backup copies in the FlashCopy Consistency Group field. If you specify Y (yes), FlashCopy consistency group option will be used and consistent copies are required. If you specify N (no) or leave the field as blank, FlashCopy consistency group will not be used.

You can specify whether catalog information for all catalogs should be collected at time of backup by changing the value in the **Capture Catalog Information for Data Set Recovery** field. If you enter R (required), the fast replication backup request fails if the catalog specified in the copy pool definition cannot be found or if the catalog information cannot be retrieved and captured in its entirety. If you enter P (preferred), under the same conditions, the backup copy pool version is valid, but you might not be able to recover some data sets in the version. Entering N (no) disables the recovery of moved or deleted data sets, this is the default.

Because mature or highly utilized catalogs often contain logical errors, capturing catalog information from a catalog with errors fails and causes the catalog search interface to return an error. When the catalog search interface returns an error, message ARC1812I is issued.

When required is specified, all catalog errors must be corrected to allow fast replication backup to succeed. Alternatively, if preferred is specified, capturing the catalog information will fail, but the fast replication backup request will continue processing.

You can also specify whether fast reverse restore is allowed by changing the value in the **Allow Fast Reverse Restore** field. If you enter Y (yes), recovery of a FlashCopy source from the FlashCopy target can occur without waiting for the background copy to complete. Entering N (no) disables fast reverse restore and requires you to wait for the original FlashCopy operation to complete prior to starting recovery, this is the default. The copy pool fast reverse restore setting is saved at the time of the fast replication backup. During fast replication recovery, the use of fast reverse restore is based on the DASD backup version setting and the state of the current FlashCopy relationships.

The other fields on the COPY POOL DEFINE and COPY POOL ALTER panels allow you to further tailor your copy pool definition.

Note:

1. With FlashCopy version 1, the source and target volumes must be in the same logical subsystem (LSS), which is limited to 256 total volumes. Thus, for three unique versions, you could have up to 64 source volumes, leaving 192 volumes available as target volumes. The source volumes within a storage group can span logical and physical subsystems.
2. When defining a copy pool, consider how many DASD backup versions to keep, whether backup versions should also be kept on tapes, and whether DFSMSHsm automatic dump processing should be used to create the copies on tapes. For tape dumps, your definition of the copy pool through ISMF includes values for the tape related options: Auto Dump, Dump Sys/Sys Group Name, and up to five dump classes.

Related reading

- For information about the ALLOWPPRCP keyword on the FRRECOV command, see [“Syntax of the FRRECOV command”](#) on page 884.
- For more information about the catalog information data set, see [“Saving catalog information at the time of backup”](#) on page 263.
- For more information about the copy pool construct, see *z/OS DFSMSdfp Storage Administration*.
- For more information about IBM Remote Pair FlashCopy, preserve mirror operation, Metro Mirror, also known as synchronous Peer-to-Peer Remote Copy (PPRC), and other copy services functions, see *z/OS DFSMS Advanced Copy Services*.

Using the copy pool backup storage group type for fast replication setup

The SMS copy pool backup storage group type contains the candidate target volumes for DFSMSHsm fast replication requests. The volumes associated with a copy pool backup storage group are for DFSMSHsm use. If you need to restrict use of a volume in a copy pool backup storage group, vary the volume off-line.

The **Copy Pool Backup SG Name** field, in the SMS storage group definition, is provided for you to specify the name of the backup storage group associated with the copy pool. You can associate a backup storage group with a copy pool storage group only. You can define a copy pool storage group with its own backup storage group or to with a backup storage group that is shared by multiple copy pool storage groups.

For each source volume in a storage group to be copied, enough eligible target volumes must exist in the copy pool backup storage group to satisfy the number of volumes required by the number of specified backup versions.

When the space efficient FlashCopy function is enabled, space efficient volumes defined in a copy pool backup storage group are eligible FlashCopy target volumes for a copy pool defined with a VERSIONS value of 0 (NOCOPY mode). Space efficient FlashCopy is supported in NOCOPY mode only. Do not assign space efficient volumes to a copy pool backup storage group associated with a copy pool defined with a VERSIONS value of 1 to 85 (COPY mode).

An eligible target volume must meet the following requirements:

- Have the same track format as the source volume.
- Be the same size as the source volume.
- For FlashCopy:
 - For FlashCopy version 1:
 - Reside in the same LSS as the source volume.
 - Not be in a FlashCopy relationship at the time of backup.
- For SnapShot:
 - Reside in the same RVA or SVA as the source volume.

Related reading

For more information about the copy pool backup storage group type and the pool storage group, see [z/OS DFSMSdfp Storage Administration](#).

Using the extend and overflow storage group types with fast replication

The SMS "extend" and "overflow" storage group types allow you to allocate data sets on volumes, or to extend data sets onto volumes that are not a part of a main source storage group. In most environments, SMS extend or overflow storage groups should *not* be used in a copy pool environment. These storage group types should only be used if all source storage groups utilizing a particular extend or overflow storage group and the extend or overflow storage group itself are all contained within the same copy pools. Extend and overflow storage groups must not be used if the source storage groups that are utilizing them are contained within different copy pools.

Related reading

For more information about the extend and overflow storage group types, see [z/OS DFSMSdfp Storage Administration](#).

Preparing for the fast replication backup

Because the task of establishing a valid DFSMSHsm fast replication environment can be nontrivial, DFSMSHsm provides a PREPARE keyword that you can use with the FRBACKUP command. The PREPARE keyword allows you to validate your fast replication environment for DFSMSHsm. DFSMSHsm preassigns the required number of target volumes to each source volume in each storage group in the specified copy pool.

When you specify the PREPARE keyword, DFSMSHsm verifies that there are sufficient eligible target volumes in the specified copy pool backup storage group. The PREPARE function does not create a backup of the copy pool, but only does initial processing. If VERSIONS=0 is defined for the copy pool, DASD backup volumes are not preassigned to the copy pool source volumes, but PREPARE does check that sufficient target volumes are available. You should only define VERSIONS=0 for a copy pool without fast reverse restore capability when only a tape backup copy is desired. With VERSIONS=0 and Allow Fast Reverse Restore=N specified, you cannot recover from the DASD backup copy.

Example: If you specified in the copy pool definition that DFSMSHsm should keep two backup versions (VERSIONS=2), DFSMSHsm preassigns two target volumes to each source volume in the copy pool.

The PREPARE function supports volume preferencing. DFSMSHsm prefers an eligible copy pool backup storage group volume in the same cluster as the source volume. If such a volume is unavailable, DFSMSHsm selects any fast replication eligible volume.

Use the PREPARE keyword whenever there is a change in the fast replication environment. Changing any of the following definitions will change the fast replication environment:

- The volumes in the pool or copy pool backup storage group are changed or removed.
- The number of backup versions for DFSMSHsm to maintain is changed.

- The storage groups defined to the copy pool have changed.

You can use the PREPARE keyword on the FRBACKUP command to verify whether a Metro Mirror primary volume is allowed to become a target of a FlashCopy operation. For more information about the SMS copy pool settings for allowing Metro Mirror primary volumes to be used during fast replication backup, see [“Using Metro Mirror primary volume during fast replication backup” on page 267](#).

Note:

1. If you specify the PREPARE keyword, DFSMSHsm does not save the catalog information.
2. If the PREPARE function fails, make the necessary corrections and reissue the PREPARE function until it is successful.

Creating a fast replication backup

The FRBACKUP command creates a fast replication backup version for each volume in every storage group defined within a copy pool. You can use the following FRBACKUP command to initiate a fast replication backup for a copy pool:

```
FRBACKUP COPYPOOL(cpname) EXECUTE TOKEN(token)
```

The fast replication backup function is not available as part of automatic backup or dump. You can invoke the FRBACKUP command from a system console, with a TSO command (HSEND CMD), with a batch job, or by using the ARCHSEND macro interface. DFSMSHsm invokes the DFSMSdss COPY FULL function to create a fast replication backup for each volume in the specified copy pool.

The FRBACKUP COPYPOOL command is successful only after DFSMSdss has successfully established a fast replication relationship for each source volume. If one of the volumes fails, the entire backup version is failed with the following results:

- DFSMSHsm continues to process volumes to determine if any remaining volumes will also encounter an error.
- For each volume in error, the target volume is disassociated from the source volume. If there is a problem with the target volume, then it should be removed from the copy pool backup storage group. Otherwise, it may be re-selected during the next FRBACKUP command.
- FlashCopy relationships that are successfully established are withdrawn after all volumes are processed.
- The backup version is marked as invalid after all volumes are processed.

DFSMSHsm skips all volumes whose SMS status has not been defined to the system. DFSMSHsm fails volumes for which it cannot obtain a UCB, such as a volume that is varied offline.

You can use the **FRBACKUP to PPRC Primary Volumes allowed** field in the SMS copy pool definition panel to indicate whether a Metro Mirror primary volume is allowed to become a target of a FlashCopy operation, and whether a preserve mirror operation is required, preferred, or not desired when the target volume is a Metro Mirror primary volume during backup. For more information about the SMS copy pool settings for the **FRBACKUP PPRC Primary Volumes allowed** field and the ALLOWPPRC keyword, see:

- [“Using Metro Mirror primary volume during fast replication backup” on page 267](#).
- [“Syntax of the FRRECOV command” on page 884](#).

Considerations for the FRBACKUP COPYPOOL command

The following topics are related to your use of the FRBACKUP COPYPOOL command:

- **Processing a FRBACKUP COPYPOOL command:** During the processing of a FRBACKUP COPYPOOL command, the new backup version is directed to the target volumes that contain the oldest backup version (after the requested number of versions has been reached). If a FRBACKUP COPYPOOL command fails, there are only $n-1$ valid versions available, where n is the number of requested versions. When a failed version exists, the failed version is the target of the new copy until that copy is successful.

The backup version is not incremented for subsequent backups until a successful version has been created.

- **When FlashCopy is the fast replication technique being used:** DFSMSHsm returns control to the system after a FlashCopy relationship for each volume has been established. However, the background copies for those volumes last from minutes to hours longer, depending on criteria such as:
 - The number of background copies being processed.
 - The amount of other higher priority activity that exists in the storage subsystem.

Notes:

1. When you are using FlashCopy version 1, any subsequent FRBACKUP COPYPOOL command that is issued against the same copy pool or against another copy pool with a common storage group before all the background copies are complete causes the command to fail. This restriction does not exist for FlashCopy version 2 because a source volume can be in multiple concurrent relationships.
 2. If catalog information is saved at the time of backup, you must have sufficient space available on your ML1 volumes to accommodate catalog information data sets. For more information on saving catalog information at time of backup, see [“Saving catalog information at the time of backup” on page 263](#).
 3. Some of the FlashCopy options are incompatible. For example, preserve mirror operation cannot be used in combination with space efficient FlashCopy or fast reverse restore. Consider these restrictions when you define a copy pool.
- **Using the TSO command:** When you use commands that specify WAIT or NOWAIT, the results that are sent back to the user depend on whether the WAIT parameter is specified, and the success or failure of one or more volumes:
 - If WAIT is specified and all volumes process successfully, a return code of zero is returned.
 - If WAIT is specified and one or more volumes fail, a nonzero return code and associated messages are returned.
 - If WAIT is not specified, control is returned to the user after the command has been accepted and verified. All messages continue to be issued, but you are not notified when the command completes.
 - **Using the TOKEN parameter:** The TOKEN parameter accepts a token of up to 40 bytes in length that is associated with that particular version of the copy pool. The token is used to reference a particular backup version. **Tip:** Use unique tokens for each version so that you can easily reference the correct version.

The token is returned as part of the LIST and ARCXTRCT data for a copy pool.

- **Refreshing the UCB on remote systems:** The VTOC and the volume serial on the target volume may have changed as a result of this operation. Before the volume can be accessed on any remote system, the UCB must be refreshed. The refresh occurs automatically if the volume is online and the device manager REFUCB function is enabled. You enable the REFUCB function through PARMLIB member DEVSUPxx or the MODIFY DEVMAN command. For more information, refer to the description of the REFUCB keyword in [z/OS MVS Initialization and Tuning Reference](#) or [z/OS MVS System Commands](#).

Serialization for the fast replication function

Attention: It is the responsibility of the caller of DFSMSHsm to ensure that the source volumes are in a state such that DFSMSHsm can process them for a meaningful recovery. This includes protecting against data set level updates (actual changes to the data) and volume level changes (DADSM functions such as allocations, renames, extends and scratches). By default, DFSMSdss serializes on the VTOC of each volume as it is processed to prevent DADSM functions from occurring against that volume.

If the caller needs to prevent DADSM functions from being performed against the set of volumes in the copy pool during the backup, it is the caller's responsibility to do so. This should be done by serializing on all of the VTOCs before invoking DFSMSHsm. Serializing on the VTOC prevents DADSM functions and also prevents DFSMSdss functions such as DEFRAG, COPY and RESTORE from modifying the VTOCs. When you perform this serialization, use the NOVTOCENQ parameter with the FRBACKUP command to indicate that DFSMSdss should not serialize on the VTOC of each volume.

Saving catalog information at the time of backup

At the time of backup, DFSMSHsm can save the catalog information for the data sets within a copy pool and store it in a DFSMSHsm-managed data set. During fast replication recovery processing, saved catalog information is used to recover and re-catalog data sets that are either deleted or moved.

You can use the COPY POOL DEFINE and COPY POOL ALTER panels to specify up to 10 catalog names. The catalog names can have up to 44 alphanumeric characters, including a delimiter. The catalogs must reside on volumes within the copy pool, collectively represent all data sets in the copy pool, and cannot contain any data sets outside of the copy pool in which they are defined.

You can specify whether catalog information for all catalogs should be collected at time of backup by changing the value in the **Capture Catalog Information for Data Set Recovery** field. If you enter R (required), the fast replication backup request fails if the catalog specified in the copy pool definition cannot be found or if the catalog information cannot be retrieved and captured in its entirety. If you enter P (preferred), under the same conditions, the backup copy pool version is valid, but you might not be able to recover some data sets in the version. Entering N (no) disables the recovery of moved or deleted data sets, this is the default.

Because mature or highly utilized catalogs often contain logical errors, capturing catalog information from a catalog with errors fails and causes the catalog search interface to return an error. When the catalog search interface returns an error, message ARC1812I is issued.

When required is specified, all catalog errors must be corrected to allow fast replication backup to succeed. Alternatively, if preferred is specified, capturing the catalog information will fail, but the fast replication backup request will continue processing.

You can also specify whether fast reverse restore is allowed by changing the value in the **Allow Fast Reverse Restore** field. If you enter Y (yes), recovery of a FlashCopy source from the FlashCopy target can occur without waiting for the background copy to complete. Entering N (no) disables fast reverse restore and requires you to wait for the original FlashCopy operation to complete prior to starting recovery, this is the default.

The following example shows the second page of the COPY POOL DEFINE and COPY POOL ALTER panels:

```

Panel  Utilities  Scroll  Help
-----
DGTDCPP5                      COPY POOL DEFINE/ALTER                      Page 2 of 5
Command ==>

SCDS Name . . : IBMUSER.SCDS
Copy Pool Name : COPYPL1

To DEFINE/ALTER Copy Pool, Specify:
Catalog Name ==> -----
               ==> -----
               ==> -----
               ==> -----
               ==> -----
               ==> -----
               ==> -----
               ==> -----
               ==> -----
               ==> -----

Capture Catalog Information for Data Set Recovery . . N   (R, P or N)
Allow Fast Reverse Restore . . . . . N   (Y or N)

Use ENTER to Perform Verification; Use UP/DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 33. COPY POOL DEFINE/ALTER panel, Page 2.

You can use the FRBACKUP command to create a fast replication backup version for sets of storage groups followed by a memory dump of each fast replication target volume. During FRBACKUP processing, information from the catalogs that are specified in the copy pool definition in SMS is captured and saved to a DFSMSHsm-managed data set, called catalog information data set (CIDS). If preferred is specified and the catalog specified in the copy pool definition cannot be found or the catalog information cannot be

retrieved, a warning message is issued, indicating that only partial catalog information is captured, and all catalog information that can be saved is saved. This allows you to recover the entire copy pool, individual copy pool volumes, cataloged data sets, and moved or deleted data sets for which the catalog information was captured. If required is specified and the catalog specified in the copy pool definition cannot be found or the catalog information cannot be retrieved, the FRBACKUP request is failed and terminates.

During processing of the FRBACKUP command, one CIDS is created for each copy pool version. Because you can specify 10 catalog names in the copy pool definition, each CIDS might contain information from up to 10 catalogs. The size of a CIDS depends on the number and type of catalog entries contained in each of the catalogs specified in the copy pool definition. Each CIDS is allocated on an ML1 volume and is cataloged. The catalog data sets are treated similarly to the way VTOC copy data sets are treated. They are not eligible for SDSPs and cannot be migrated to ML2 volumes. When you enter the FRBACKUP command, you must ensure that you have sufficient space available on ML1 volumes to accommodate the catalog information data sets. Catalog information data sets are deleted when their associated versions are rolled off, deleted, expired, or withdrawn.

Note: Catalog information data sets are allocated on 50 primary and 50 secondary cylinders, unless this setting is modified by patches described in the [Tuning DFSMSHsm](#) topic of *z/OS DFSMSHsm Implementation and Customization Guide*.

To move valid backup versions and migration copies from DFSMSHsm-owned DASD volumes, you can use the FREEVOL command. FREEVOL allows you to move the catalog data sets from a single ML1 volume to other ML1 volumes. Catalog information data sets are moved from the freed volume to other ML1 volumes, regardless of the target level of the FREEVOL command.

The naming convention for a CIDS is as follows:

```
PREFIX.HSMCIDS.Dyyddd.Thhmmss.Cnumseq
```

where:

PREFIX

is the value of the BACKUPPREFIX parameter of the SETSYS command, if specified. Otherwise, PREFIX is the DFSMSHsm default UID.

yddd

is the year and day of the backup.

hhmmss

is the time in hours, minutes, and seconds of when the fast replication backup was created.

numseq

is the qualifier that ensures that the data set name is unique. It is a 3-digit number. A '0' followed by a value derived from the host ID.

Valid host IDs are: 0 - 9, A - Z, \$, #, and @. DFSMSHsm converts the host ID to a value of 1 through 39 in ascending order. If a data set name collision occurs, a '1' is substituted in the '0' position. If another data set name collision occurs, a '2' is substituted in the '0' position. If that fails, creation of the CIDS fails.

Note: HSMCIDS, D, T, and C are constants.

Dumping a fast replication backup to tape

You can dump fast replication DASD backup copies to tape. You can create these dumps on tape through the FRBACKUP command or through an automatic dump processing. For more information on automatic dump processing of copy pools, see [“Dump processing” on page 193](#)

To create a fast replication DASD backup and dump it to tape, you can use the following command:

```
FRBACKUP CP(COPYPOOL1) EXECUTE DUMP RETURNCONTROL(FREND)
```

In this example, DFSMSHsm creates a fast replication backup copy on DASD for each volume of each storage group defined in the COPYPOOL1 copy pool. After the fast replication DASD copy has been

successfully created, DFSMSHsm returns control to the user and then dumps the fast replication DASD copy to tape. The time stamp of all the dump copies reflects the time that the DASD copy is created. If RETURNCONTROL(DUMPEND) is specified, then control is returned to the user after the dump copy has been created. The COPYPOOL keyword and associated parameter indicate the name of a previously defined copy pool which the FRBACKUP command should process.

The copy pool definition specifies how the dump copies are created. Up to five dump classes can be specified for each copy pool. Each dump class correlates to one dump copy. The dump class defines settings such as frequency, expiration and whether the dump class is optional or required. For more details on the REQUIRED or OPTIONAL definition of dump classes, see [“Specifying whether the dump class is required or optional” on page 176](#).

A dump class of a Fast Replication version can have the following states: COMPLETE, PARTIAL, or FAILED. COMPLETE indicates that the volumes are successfully dumped for that dump class. PARTIAL indicates that not all volumes for the dump class are processed. FAILED indicates that *no* volume attempted for the dump class has been dumped successfully.

A Fast Replication dump version can have the following states: ALLCOMPLETE, REQUIRED COMPLETE, PARTIAL, NONE, and FAILED. ALLCOMPLETE indicates that all dump classes specified in the copy pool definition or the FRBACKUP command have completed successfully. REQUIRED COMPLETE indicates that all dump classes marked as REQUIRED are successfully dumped and that one or more dump classes marked as OPTIONAL are incomplete. PARTIAL indicates that one or more dump classes are incomplete. This can be any combination of REQUIRED and OPTIONAL dump classes. FAILED indicates that no attempted volumes have been successfully dumped for any of the specified dump classes. Use the LIST COPYPOOL(*cpname*) command to display the status of the dump classes for each copy pool. The SELECT(DUMPSTATE) option can be used to find incomplete dump copies. NONE indicates that a dump copy has not been created for this version.

Dump copies can only be created from valid DASD copies. DFSMSHsm ensures that once the creation of a dump copy has begun for a DASD copy, that the DASD copy will not roll-off until the dump copy is complete. If an FRBACKUP command is issued that will roll-off a DASD version that has a dump version with a status FAILED or PARTIAL, and one or more dump classes are REQUIRED, then the FRBACKUP command fails. If all of the dump classes are OPTIONAL, then the FRBACKUP command is allowed to continue. Failing the FRBACKUP command ensures that the dump copy can be retried with the FRBACKUP DUMPONLY command before the DASD copy is rolled-off. If it is acceptable to have an incomplete dump copy so that a new DASD backup copy can be created, then use the FORCE keyword on the FRBACKUP command.

You can specify, in the copy pool definition, the number of DASD versions with background copy maintained by DFSMSHsm, from 0 (NOCOPY mode) to 85 (COPY mode). If you specify 0 for a copy pool defined with fast reverse restore capability, it indicates a DASD backup copy created in NOCOPY mode can be used for recovery and as a backup copy from which to create a dump tape copy. The DASD backup copy is kept until it is withdrawn, deleted, or replaced. If you specify 0 for a copy pool defined without fast reverse restore capability, it indicates that you do not require a DASD copy to recover from and DFSMSHsm is to withdraw the DASD version when the dump tape version has been created successfully. The number of tape versions required is independent of the number of DASD versions requested. The number of available tape versions is managed by expiration processing, or by explicit deletion of the dump version using the FRDELETE command. AUTODUMP processing expires the fast replication dumps based on expiration criteria specified in each dump class. Because up to 85 versions of a copy pool can be kept at any time, FRBACKUP DUMP processing also expires the 85th dump when a subsequent dump is requested.

DUMPCCLASS is a keyword for the FRBACKUP command. From 1 to 5 dump classes can be specified, with 1 dump copy of each volume produced for each dump class. If DUMPCCLASS is specified on the FRBACKUP command, dump classes defined to the copy pool are not considered. Since stacking can be specified in the dump class, the FRBACKUP command does not support any separate stacking keyword.

DUMPONLY is used to create one or more dump copies from an existing DASD copy. DUMPONLY can be used to:

- create one or more dump copies when there are no existing dump copies.

- create additional dump copies (up to five) when one or more dump copies already exist.
- attempt to complete existing dump copies that failed.

VERSION, DATE, TOKEN and GENERATION are mutually exclusive keywords describing which DASD version should be dumped. If none of those keywords are specified, then the latest DASD copy is dumped. A DASD copy must be valid in order for it to be dumped. If the DUMPONLY command is issued against a version with an invalid DASD copy, then the command fails.

Only 5 dump classes can be associated with a single dump version, so an FRBACKUP command specifying one or more dump classes fails if *both* the number of existing dump copies for a version and the number of dump classes specified on the FRBACKUP command exceed 5.

If DUMPONLY is issued against a version for which there are no existing dump copies, then the following statements hold true:

- If DUMPCLASS is not specified, then the dump classes from the copy pool definition are used.
- If DUMPCLASS is specified, then the specified dump classes are used.

If DUMPONLY is issued against a version for which there are existing successful dump copies, then the following statements hold true:

- If DUMPCLASS is not specified, then the command fails.
- If DUMPCLASS is specified, then only those dump classes that were not already created are processed. The number of existing dump copies and new dump copies cannot exceed five.

If DUMPONLY is issued against a version that is marked partially successful, then the following statements hold true:

- If DUMPCLASS is not specified, the dump is resumed for any dump classes not previously successfully dumped.
- If DUMPCLASS is specified and the dump class previously failed, then only that dump class is resumed.
- If DUMPCLASS is specified with a dump class that is not previously attempted, and fewer than 5 dump classes have been dumped for the version, a new dump copy of the version is created for the specified dump class.
- If DUMPCLASS is specified with a dump class that is not previously attempted, and 5 dump classes have already been dumped for the version, the command fails.

In a VERSIONS = 0 (NOCOPY) environment, if the DASD version is marked NONE or FAILED, and the dump for the version is marked ALL COMPLETE or REQUIRED COMPLETE, any DUMPONLY request for that version fails.

In the following example, the DATE subparameter is used to dump to tape the backup of COPYPOOL1 that was created on June 2, 2006:

```
FRBACKUP CP(COPYPOOL1) DUMPONLY(DATE(2006/06/02))
```

RETURNCONTROL, a keyword telling DFSMSHsm when to return control for WAIT type user commands. RETURNCONTROL(FASTREPLICATIONEND) tells DFSMSHsm to return control just before dump processing starts. RETURNCONTROL(DUMPEND) tells DFSMSHsm to return control after all volumes have been dumped. The keyword applies to the DUMP keyword but not to the DUMPONLY keyword. DUMPEND is the default. If a user specifies the RETURNCONTROL() keyword with NOWAIT, the command fails. If a user specifies the RETURNCONTROL() keyword without WAIT, the command fails with a syntax error. NOWAIT is the default.

Related reading

For more information about FRBACKUP and the parameters related to dumping fast replication backups, see Chapter 46, [“FRBACKUP command: Requesting a fast replication backup or dump version,”](#) on page 865.

Creating a copy pool incremental version

DFSMSHsm supports the FlashCopy Incremental function for Fast Replication copy pools. A copy pool can be designated to have an incremental backup version by using the FCINCREMENTAL keyword on the FRBACKUP command. When the software and microcode requirements are met, you can use the FCINCREMENTAL keyword on multiple FRBACKUP commands to specify that those copies are to be incremental, or add the ALL subparameter to specify that all subsequent copy pool backup copies should also be incremental.

Once an incremental version has been established in a copy pool, the copy pool has an incremental version until either it is deleted or the FCINCRREMENTALLAST keyword is used. The volumes in a copy pool incremental version have persistent relationships with the target volumes. When a single disk version is maintained, the FlashCopy Incremental function is used each time a new disk version is created. When multiple disk versions are being maintained, then the FlashCopy Incremental function is used only for the version that was originally identified as the Incremental version by the FCINCREMENTAL keyword. All other versions are full copies. For example, if two disk versions are being maintained, then every other disk version uses the FlashCopy Incremental function.

The FCINCREMENTAL keyword is optional when creating a new incremental version. If the FCINCREMENTAL or FCINCRREMENTALLAST keyword is specified, and the oldest disk generation to be overwritten to create the new version is not an incremental version, the command fails.

When a copy pool incremental version is no longer needed for a copy pool, issue FCINCRREMENTALLAST when you create the next incremental version. After the background copies for the incremental version have completed, the persistent relationship for each volume is withdrawn. The disk version remains recoverable. An alternative method is to delete the incremental version with the FRDELETE command. After using either the FCINCRREMENTALLAST keyword or the FRDELETE command, all future disk versions are full copies until the FCINCREMENTAL keyword is used again. Adding the ALL subparameter to FCINCRREMENTALLAST causes a new incremental of the oldest copy to be taken, the persistent relationships of that copy to be withdrawn, and the relationships of all other backup copies to be withdrawn after the next increment is taken of each of those copies.

The initial establishment of a FlashCopy Incremental version creates a full copy of each volume (0-n tracks). Subsequent copies utilize the FlashCopy Incremental technology only to copy tracks from the source that have changed. This results in the background copy completing in a shorter time period.

Fast Replication incremental copy pool versions are not supported when the copy pool is defined with zero DASD backup versions. If the number of DASD backup versions in the copy pool definition is changed to zero and an incremental version exists, the subsequent FRBACKUP command fails. The incremental version must be removed by the FCINCRREMENTALLAST keyword or deleted by the FRDELETE command.

If the number of versions is reduced so that an incremental version is automatically deleted, the FRBACKUP command fails. The incremental version either has to be deleted or has to be the oldest disk generation for FRBACKUP processing to be successful.

Using Metro Mirror primary volume during fast replication backup

You can use the **FRBACKUP to PPRC Primary Volumes allowed** field in the SMS copy pool definition panel to indicate whether a Metro Mirror or Global Mirror primary volume is an eligible FlashCopy target, and whether a preserve mirror operation is required, preferred, or not desired when the target volume is a Metro Mirror primary volume during backup processing.

If you specify a value in the **FRBACKUP to PPRC Primary Volumes allowed** field, the value specified is considered in selecting FlashCopy target volumes for FRBACKUP processing during volume pairing. Otherwise, if you do not specify a value, DFSMSHsm uses the currently stored preserve mirror indicator for the copy pool. If no preserve mirror indicator is set for the copy pool, the default is NO.

If you specify a value in the **FRRECOV to PPRC Primary Volumes allowed** field in the SMS copy pool definition panel, the L0 volumes are not examined for FlashCopy to Metro Mirror or Global Mirror primary eligibility for FRRECOV processing during volume pairing. If preserve mirror is required during fast replication recovery using FlashCopy, you must ensure that your copy pool environment meets preserve mirror configuration requirements.

If preserve mirror is used in combination with incremental FlashCopy, the use of the PMPREF (preserve mirror preferred) or PMREQ (preserve mirror required) option needs to be consistent with the existing relationship when taking an increment. For example, if the existing incremental relationship was established without preserve mirror, an increment cannot be taken with the PMREQ option because the remote relationship will not exist and, therefore, an increment cannot be taken at the remote site. Do not change your preserve mirror settings between increments.

Note:

1. A Metro Mirror or Global Mirror primary volume is not allowed to be a FlashCopy target volume when:
 - NO is specified in the **FRBACKUP to PPRC Primary Volumes allowed** field
 - the NO option is in effect by default
 - the capability is not supported by the storage controller
2. Specifying PMNO, PMPREF, or PMREQ in the **FRBACKUP to PPRC Primary Volumes allowed** field does not guarantee that a Metro Mirror or Global Mirror primary volume is selected. You can ensure that a Metro Mirror or Global Mirror primary volume is selected by ensuring that only Metro Mirror or Global Mirror primary volumes exist in the copy pool backup storage group to be used.
3. When you specify PMNO or PMPREF in the **FRBACKUP to PPRC Primary Volumes allowed** field, use SETSYS FASTREPLICATION(DATASETRECOVERY(NONE)) to prevent the source (LO) volume from going into a duplex pending state during the recovery of an individual data set from disk.
4. The preserve mirror operation cannot be used in combination with fast reverse restore. Do not specify PMPREF or PMREQ in the **FRBACKUP to PPRC Primary Volumes allowed** or **FRRECOV to PPRC Primary Volumes allowed** fields for a copy pool defined allowing fast reverse restore.
5. The preserve mirror operation cannot be used in combination with space efficient FlashCopy. When you specify PMPREF or PMREQ in the **FRBACKUP to PPRC Primary Volumes allowed** field, space efficient volumes are not eligible FlashCopy targets.
6. When in a mixed release environment, releases prior to z/OS V1R11 specify the ALLOWPPRCP settings on the FRBACKUP command. z/OS V1R11 will always use the SMS values first. If the value has been set in the SMS copy pool definition in z/OS V1R11, the DFSMSShsm setting will be updated and the lower release hosts will use that value.
7. A Global Mirror primary volume is not allowed to be a FlashCopy target volume when:
 - NO is specified in the FRBACKUP to PPRC Primary Volumes Allowed field
 - the NO option is in effect by default
 - the capability is not supported by the storage controller
 - DISABLE(FLASHCOPYTOGM) is specified in your DEVSUPxx parmlib member
 - the DISABLE(FLASHCOPYTOGM) is in effect by default

Related reading

- For more information about IBM Remote Pair FlashCopy, Metro Mirror, also known as synchronous Peer-to-Peer Remote Copy (PPRC), and other copy services functions, see [z/OS DFSMS Advanced Copy Services](#).
- For more information about the ALLOWPPRCP keyword, see [“Syntax of the FRRECOV command” on page 884](#).

Using the XRC primary volume during fast replication backup

You can use the FRBACKUP to XRC Primary Volumes allowed field in the SMS copy pool definition panel to indicate whether an XRC primary volume is an eligible FlashCopy target during backup processing.

If you specify a value (YES/NO) in the FRBACKUP to XRC Primary Volumes allowed field, the value specified is considered in selecting FlashCopy target volumes for FRBACKUP processing during volume pairing. If no XRC Primary allowed indicator is set for the copy pool, the default is NO. When you specify a value in the FRRECOV to XRC Primary Volumes allowed field in the SMS copy pool definition panel, the LO

volumes are not examined for FlashCopy XRC primary eligibility for FRRECOV processing during volume pairing. If remote pair FlashCopy to XRC is required during fast replication recovery, you must ensure that your copy pool environment meets remote pair FlashCopy/XRC configuration requirements.

The fast reverse restore option cannot be combined with the FlashCopy to XRC Primary option. Do not specify FRBACKUP to XRC Primary Volumes (YES) or FRRECOV to XRC Primary Volumes (YES) for a copy pool defined allowing fast reverse restore.

In a Metro/zOS GM environment, both XRC Primary YES and the PMREQ option must be set to allow FlashCopy to metro mirror and XRC primary volumes. See *z/OS DFSMS Advanced Copy Services* for more information.

Creating consistent copies using FlashCopy consistency groups

When logically related data spans across multiple volumes, FlashCopy consistency groups can be used to create consistent copies while minimizing application impact. You can indicate consistent backup copies are required for a copy pool by specifying **FlashCopy consistency group** in the SMS copy pool definition.

When this is specified, a consistency group will be formed during FRBACKUP. As each copy pool volume is processed, the FCFREEZE option will be used to specify that the source volume is to be part of a FlashCopy consistency group. The control unit will put the source volume in an extended long busy (ELB) state, causing I/O activities to be held at the host. When all volumes have been processed, DFSMSHsm will issue the DFSMSdss CGCREATED command to thaw all of the frozen volumes and I/O activities will resume on the volumes.

The frozen volumes will remain in extended long busy state until the Consistency Group Created command is received or when the FlashCopy consistency group timer expires. The default timer value is 2 minutes for CKD devices. If the frozen FlashCopy volumes are not already thawed when timer expires, the control unit will reset the extended long busy state and the held I/O activities will resume.

Since it is possible that the FlashCopy consistency group timer might expire before the consistency group has been formed, or that one or more frozen volumes might be thawed unexpectedly by another FlashCopy consistency group operation or by the thaw command, DFSMSHsm will always validate the frozen (ELB) state of the source volumes before thawing the volumes. If DFSMSHsm detects that the copy pool backup version might not be a consistent backup, it will fail the FRBACKUP operation and invalidate the new copy pool backup version.

Note:

1. To ensure a consistent copy can be created within the time window, the DFSMSHsm FlashCopy consistency group function is intended to be used for small LOG copy pools.
2. The thaw command is processed by each LSS receiving the command for all frozen FlashCopy volumes residing in the LSS, regardless of which FRBACKUP command (or job) froze them. Therefore, users should avoid creating multiple FlashCopy consistency groups with volumes in the same LSS at the same time. For example, do not run FRBACKUP of multiple copy pools using the FlashCopy consistency group option from multiple hosts concurrently, where the source volumes residing in the same LSS belong to the different copy pools being backed up.
3. When the FlashCopy consistency group option is used, all volumes are processed under one DFSMSdss server (address space). The total number of source volumes allowed in a consistency group is limited to 252.

Parallelism for fast replication backup copies to DASD

DFSMSHsm processes FRBACKUP COPYPOOL commands serially. To maximize throughput, DFSMSHsm creates the fast replication backup for the volumes in a copy pool in parallel. There are two levels of parallelism. The first level is that DFSMSHsm invokes DFSMSdss with the PARALLEL option and up to 254 volume pairs. The second level of parallelism is the number of DFSMSdss invocations that DFSMSHsm processes concurrently. DFSMSHsm processes between 1 and 64 DFSMSdss invocations concurrently, the default being 15. The number of concurrent DFSMSdss invocations that are processed and the number of parallel volumes passed to DFSMSdss can be modified with the SETSYS MAXCOPYPOOLTASKS(FRBACKUP(*nn*) DSS(*nnn*)) command.

Note: When the FRBACKUP COPYPOOL command uses FlashCopy consistency groups, DFSMSHsm processes all the copy pool volumes in one DFSMSdss invocation, regardless of the SETSYS MAXCOPYPOOLTASKS(FRBACKUP(*nn*) DSS(*nnn*)) setting.

Parallelism for fast replication backup copies to dump tape

DFSMSHsm uses the existing dump processing structure to dump fast replication target volumes to tape. Use the SETSYS MAXDUMPTASKS command to specify the number of concurrent volume dump tasks to use.

Recovering a fast replication backup

You can use the FRRECOV command to recover copy pools, volumes and data sets from the fast replication backup versions on DASD or tape. Prior to the recovery of a copy pool, DFSMSHsm will unallocate the specified catalogs. If DFSMSHsm cannot unallocate any of the catalogs, the FRRECOV operation might fail. Catalog reallocates the catalogs when used.

Examples

- You can use the following command to recover all of the volumes from a copy pool:

```
FRRECOV COPYPOOL(cpname)
```

The COPYPOOL(*cpname*) keyword indicates that all volumes associated with the named copy pool are to be recovered. You can specify the VERIFY keyword to verify that none of the volumes are in an unexpected FlashCopy relationship before starting the recovery. When the copy pool is defined allowing fast reverse restore, VERIFY(Y) indicates DFSMSHsm is to verify that the state of the FlashCopy relationships meet the fast reverse restore requirements. Otherwise, regular FlashCopy can be used for the copy pool or the FRRECOV command will be failed. VERIFY(N) indicates that the user has verified that the recovery can proceed successfully. VERIFY(N) cannot be specified when the copy pool is defined allowing fast reverse restore unless a previous FRRECOV COPYPOOL operation has determined the copy pool backup version is no longer eligible to use fast reverse restore. Regular fast replication recovery can be used instead. If one or more volumes fail, the FRRECOV command ends with a nonzero return code. If the copy pool was recovered without using fast reverse restore, volumes that fail recovery can be tried again using the TOVOLUME keyword. If the copy pool was recovered using fast reverse restore and the recovery was incomplete, volumes that were not recovered can be tried again by issuing the FRRECOV COPYPOOL command again. The RESUME(YES) keyword can also be used to retry recovery of volumes that previously failed to recover.

- You can use the following command to recover individual volumes:

```
FRRECOV TOVOLUME(volser)
```

The TOVOLUME keyword indicates that only 1 volume should be recovered from the copy pool. If the source volume specified by volume serial number is contained within a storage group that is defined within multiple copy pools, you must also specify the optional parameter FROMCOPYPOOL(*cpname*).

FRRECOV recovers the most recent DASD version available if it is not given a more specific source. If GENERATION(0) exists on both DASD and tape, the DASD copy is recovered. If the keyword FROMDUMP is used without any further specification of which dump to recover from, recovery of GENERATION(0) is attempted. If GENERATION(0) does not have a dump, then the command fails.

You can use the following command to retrieve the most recent copy of GENERATION(0) (DASD or DUMP) for the volume in the specified copy pool:

```
FRRECOV TOVOLUME(volser) FROMCOPYPOOL(cpname)
```

You can use the following command to recover the GEN(0) dump version of volume *volser*:


```
FRRECOV TOVOLUME(volser) FROMCOPYPOOL(cpname) FROMDUMP
```

You can specify the FROMDASD keyword to only consider DASD copies.

You can specify the FROMDUMP keyword to consider dump copies only. If two or more dump classes are used for the dump, you can use the DUMPCLASS(*dclass*) keyword to only consider dumps from *dclass*. Specify the DUMPVOLUME(*dvol*) keyword to use tape *volser* as the source.

Note: DUMPCLASS and DUMPVOLUME are mutually exclusive.

If a version other than the most recent must be recovered, you can specify the GENERATION, VERSION, DATE, or TOKEN keyword. Select the appropriate generation or version from the available versions that are returned by the DFSMSHsm LIST command or ARCXTRCT macro.

If DATE is specified, and more than one version exists for that date, then the most recent version from that date is recovered.

Use the TOKEN keyword to recover the unique version that corresponds to that token. If a token of less than 40 characters is passed in, then the token is left justified and padded with binary zeros before a backup version with a matching token is searched for. A hexadecimal token can be specified by using the X'...' Syntax. For example, TOKEN(12345) is equivalent to TOKEN(X'F1F2F3F4F5'). If the token is not unique, the most recent version that corresponds to the token is recovered. If no version exists with the specified token, the recover operation fails.

Results sent back to the user depend on whether the WAIT parameter is specified, and the success or failure of one or more volumes:

- If WAIT is specified by the user and all volumes are successfully processed, a successful return code of zero is returned.
- If WAIT is specified by the user and one or more volumes failed, a non-zero return code, and associated messages, are returned to the caller.
- If WAIT is not specified by the user, control is returned to the caller after the command has been accepted and verified. All messages continue to be issued, but the user is not notified when the command completes.

There is no keyword to perform a recovery at the storage group level.

Related reading

For more information on using the FRRECOV command, see [Chapter 49, “FRRECOV command: Requesting a fast replication recovery,” on page 883](#)

Recovering a data set from a fast replication backup

You can use the FRRECOV command to recover individual data sets from a copy pool backup. On FRRECOV, specify one or more data set names (fully or partially qualified). If you specify a partially qualified data set name with a filter (wild card), DFSMSHsm processes every cataloged data set in the standard search order that matches your filtering criteria. The FRRECOV command allows you to:

- Specify fully qualified data set names.
- Specify partially qualified data set names.
- Use wild cards for data sets that have been moved and are cataloged. FRRECOV processing will attempt to recover and catalog moved data sets to the volume where they were allocated at the time of backup. If unable to recover the data set to the original volume, the most eligible volume with the most free space within the storage group will be selected.
- Recover an uncataloged data set, if the data set is fully qualified.
- Recover data sets that have been deleted or moved. FRRECOV processing uses the catalog information gathered at the time of backup to recover the data set. If the data set is cataloged at the time of recovery, FRRECOV compares the current catalog information to catalog information gathered at the time of backup to determine if it is a moved data set. If the data set has been deleted, DFSMSHsm

references the catalog information collected at the time of backup to recover the data set. FRRECOV will attempt to recover and catalog moved and deleted data sets to the volumes in which they were allocated at the time of backup. If unable to recover the data set to the original volume, the most eligible volume with the most free space within the storage group will be selected.

- Specify more than one data set and use wild cards in a multi-data set request.

Table 28 on page 272 shows the wildcard support on the FRRECOV command.

<i>Table 28. Wildcard support on the FRRECOV command.</i>			
Type of data sets that are specified with the FRRECOV command	Moved data sets	Deleted data sets	Original allocation
Single fully qualified data set	Process	Process	Process
Multiple fully qualified data sets	Process	Process	Process
Wildcard specified on a single or multiple data sets	Process	Not processed	Process

Note: You cannot use wild cards for deleted data sets.

For a FRRECOV of a data set to be successful, DFSMSHsm must have catalog information from either the catalog information data set captured at backup time or current catalog information at the time of recovery. If the catalog information is not captured, but the data set is cataloged at the time of recovery, the data set recovery is attempted. If the catalog information is not captured and the data set is cataloged at backup time, but it is not cataloged at the time of recovery, the individual data sets cannot be recovered using the DSNAMES() keyword.

If you omit a copy pool catalog name from the copy pool definition, data sets from that catalog are eligible for recovery if they were backed up in the copy pool and if they are currently cataloged to the same volumes in which they were allocated to at the time of backup. If you specify a catalog name that does not belong to the copy pool, DFSMSHsm captures the information for that catalog at the time of backup. During FRRECOV processing, DFSMSHsm uses the irrelevant catalog information to attempt to recover the data set, determines that the data set is not backed up as part of the copy pool, and fails the request. You can identify data sets that are backed up as part of a copy pool version using the DATASETS keyword on the LIST COPYPOOL command.

When you attempt to recover a data set that has changed volumes, the data set might be scratched and re-cataloged or just re-cataloged. If the data set is cataloged at the backup time, but not cataloged at the recovery time, the data set is allocated to the same volume on which the data set was allocated at the time of backup and DFSMSHsm re-catalogs the data set. If a data set is cataloged to different volumes at the recovery time than it was cataloged at the backup time, the data set is scratched, allocated to the volumes on which the data set was allocated at the time of backup, and then DFSMSHsm re-catalogs the data set. If this initial recovery attempt fails, DFSMSHsm will perform SMS volume selection within the primary storage group and retry data set recovery.

If you define a storage group in multiple copy pools, and then create a backup, the data sets in this storage group have multiple copies on the backup volumes. To recover these data sets, use the FROMCOPYPOOL(*cpname*) keyword on FRRECOV command to identify which copy of the data set to recover. You must also specify FROMCOPYPOOL(*cpname*) and a fully qualified data set name to recover a deleted data set. If you do not specify FROMCOPYPOOL for a deleted data set, DFSMSHsm fails.

FRRECOV processes each data set independently. If a data set is not found, FRRECOV continues processing until the other specified data sets are processed. On completion, FRRECOV processing issues messages to indicate which data sets are recovered and which requests fail. For wait-type requests, FRRECOV returns control when all of the data set recoveries have completed.

For a specified data set name, the high-level qualifier can be fully or partially qualified. You cannot specify two asterisks ('**') as the high-level qualifier or as the only qualifier but it can be specified in place of subsequent qualifiers. You can specify '*' within the high-level qualifier as long as it is not the first character specified. The remainder of the data set name can include any arrangement of wildcard characters. These restrictions help to prevent inadvertent requests for large numbers of data sets.

You cannot recover the following types of data sets using fast replication data set recovery:

- User catalog
- VVDS
- VTOC index
- VSAM key-range
- Migrated data sets
- GDG base

In the following example, DFSMSHsm requests a fast replication recovery for two data sets (DSNAME1 and DSNAME2).

```
FRRECOV DSNAME(DSNAME1,DSNAME2) REPLACE
```

You can use the following command to recover individual data sets:

```
FRRECOV DSNAME(USER.DATA.**,USER.LOG.**,USER.JOURNAL) REPLACE
```

The DSNAME keyword is mutually exclusive with the TOVOLUME and COPYPOOL keywords. If GEN, VER, DATE, TOKEN, or FROMDUMP(DVOL) are not specified, then the default is GEN(0).

NEWNAME is an optional keyword that allows you to recover a data set to different name on the same or different system, while preserving the originally-named data set for analysis or other purposes. You must specify the DSNAME keyword if you are also specifying NEWNAME(*newdsname*). If a data set already exists with the same name as the new name you are specifying, the REPLACE keyword must be specified to replace the existing data set.

REPLACE is a required keyword for cataloged data sets. FROMCOPYPOOL is a required keyword for deleted data sets. In order to recover a data set that is uncataloged but exists on a volume, you must specify both REPLACE and FROMCOPYPOOL.

FROMDASD and FROMDUMP are mutually exclusive optional parameters that can be used to indicate the source of the recovery. The default is to recover the data set from DASD. If a DASD copy is not available, then the tape copy is recovered. You cannot specify DVOL(*dvol*) with GEN, VER, DATE, or TOKEN.

FROMCOPYPOOL is an optional keyword, but you must specify this keyword if the volume or volumes that a data set resides on belong to multiple copy pools. FROMCOPYPOOL is only valid with the TOVOLUME and DSNAME keywords. You must specify FROMCOPYPOOL for recovery of deleted data sets. NOCOPYPOOLBACKUP is an optional keyword. When a partially qualified data set name is specified and a copy pool backup version is not found for a data set that matches the filtering criteria, the default return code is 8. When RC4 is specified with this keyword, a return code of 4 is returned instead. This keyword is ignored for fully qualified data set names.

FASTREPLICATION is an optional keyword. It overrides any value specified on the SETSYS FASTREPLICATION command. If not specified, the value of SETSYS FASTREPLICATION is used. This keyword is only valid with DSNAME. The command fails if this keyword is specified with COPYPOOL or TOVOLUME.

You can use the LIST COPYPOOL command to list all data sets that are backed up with a specified version. The list output also contains the data sets that were backed up but have been deleted, and indicates whether a full, partial, or no catalog information is captured for a specific copy pool version. Data sets are listed in alphabetical order by catalog:

```
LIST COPYPOOL(cpname) DATASETS(GENERATION(gen)|TOKEN(token))
```


Related reading

- For information on how to specify PPRC options for FRBACKUP and FRRECOV, see [Figure 33 on page 263](#).
- For information about the AUDIT command, see [Chapter 30, “AUDIT command: Auditing DFSMSHsm,” on page 737](#).
- For information about the FRBACKUP command, see [Chapter 46, “FRBACKUP command: Requesting a fast replication backup or dump version,” on page 865](#).
- For information about the FRRECOV command, see [Chapter 49, “FRRECOV command: Requesting a fast replication recovery,” on page 883](#).
- For information about the LIST command, see [Chapter 53, “LIST command: Listing information from the MCDS, BCDS, and OCDS,” on page 919](#).

Considerations for the FRRECOV command

The following topics are related to your use of the FRRECOV command:

- **Before processing an FRRECOV COPYPOOL command:** DFSMSDss restores a volume only if the volume is not allocated to another job. If the volume contains a catalog, the catalog address space in a DFSMSDfp environment may have the catalog (and the volume) allocated. If the copy pool catalog names are specified in the SMS copy pool definition, DFSMSHsm attempts to unallocate the catalogs prior to recovery. If the catalog names are not specified in the copy pool definition or the unallocation fails, follow these steps:
 1. Issue the F CATALOG,OPEN(*volser*) command to determine which catalogs on the target volume are allocated.
 2. For each catalog that is open, issue the F CATALOG,UNALLOCATE (*catalogname*) to cause the catalog address space to close the catalog and deallocate it. When the last catalog on the volume is deallocated, the target volume is also deallocated by the catalog address space.
 3. Issue a D U,,addr,1 command to ensure that the volume is not allocated by another user.
 4. If the volume is not marked as allocated, issue the FRRECOV command to DFSMSHsm.

If the catalogs on copy pool volumes are shared in ECS mode, the following steps must be followed. Otherwise, a catalog might be allocated on a different LPAR than the one doing the restore. This could cause a catalog breakage which might not be detected immediately. .

To avoid this, follow following steps before performing FRRECOV:

1. Remove any catalogs on the volumes to be restored from ECS using F CATALOG,ECSHR(REMOVE,*catalogname*) command.
2. LOCK or SUSPEND all the catalogs on the volumes which will be restored using F CATALOG,RECOVER,LOCK | SUSPEND(*catalogname*). SUSPEND is preferred.
3. UNALLOCATE all the catalogs on the volumes which will be restored from all systems using F CATALOG,UNALLOCATE(*catalogname*).
4. Issue FRRECOV to restore the volumes.
5. .If the backup copy of the catalogs were not LOCKed or SUSPENDed, there is no need to unlock them, but this can be done anyway.
6. Bring back any ECS catalogs that were removed from ECS with F CATALOG,ECSHR(ENABLE,*catalogname*).

Note: You may need to issue F CATALOG,VCLOSE(*volser*) after the full volume restores (Step 4) for each volume restored on each system. Or you could simply restart Catalog afterward to ensure every system is looking at the current VVDSs.

- **Modifying FRRECOV wait time for unallocating catalogs:** In a Common Recover Queue SYSPLEX environment, FRRECOV will issue the catalog unallocate command for each catalog name specified in the copy pool definition at the time of the FRRECOV. Wait time applies to all catalogs in the copy pool definition, it is not per catalog.

The wait time allows users to modify the amount of time FRRECOV will wait for the catalog to be unallocated prior to continuing with the recover process. The wait time is only utilized if operating in a CVQ environment or environment where the FRGCBF_ROUTEALL_UNALLOC bit has been patched on in the FRGCB.

The FRGCBF_ROUTEALL_UNALLOC bit defaults to ON in release V2R3 and later.

The default time is 3 seconds, a wait of 3 seconds = 300 one-hundredths of a second. This wait time can be adjusted with a patch to the FRGCB. For example, to change from 3 seconds to 6 seconds:

```
PATCH .FRGCB.+84 X'00000258' VERIFY(.FRGCB.+84 X'0000012C')
```

The field name being patched is FRGCB_CAT_UNALLOC_WTIME. If you do not want a wait time to be used, then patch the FRGCB_CAT_UNALLOC_WTIME field to X'00000000'.

- **Processing an FRRECOV command:** When SnapShot is used as the fast replication utility, a recovery can be performed immediately after the DFSMSHsm FRBACKUP command completes.

When FlashCopy is used as the fast replication utility, a recovery from a fast replication backup cannot be performed if any of the volumes are in an existing FlashCopy relationship unless fast reverse restore is to be used. If fast reverse restore is not intended, do not attempt to recover the copy pool until all of the background copies have completed from an invocation of the FRBACKUP command. You can use the QUERY COPYPOOL command to determine if any of the source volumes for a copy pool are in a FlashCopy relationship.

When a copy pool is defined without fast reverse restore capability, you can use the FRRECOV command with the VERIFY keyword to prevent the initiation of a recovery of a copy pool for which all of the background copies from the backup have not yet completed. When you specify VERIFY(Y), before initiating any volume recoveries, DFSMSHsm first determines if any of the volumes are in an existing FlashCopy relationship. If so, the recover fails. If you need an immediate recovery, you have two options:

1. Wait until all of the background copies have completed. Then the latest backup version can be used for the recovery.
2. For FlashCopy version 1, withdraw the outstanding FlashCopy relationships using the FRBACKUP COPYPOOL(*cpname*) WITHDRAW command. Doing this invalidates the latest version of the backup and enables an immediate recovery from the prior version. This process only works for FlashCopy version 2 when the generation 1 background copies are complete.



Attention: Use the LIST COPYPOOL command or the ARCXTRCT macro to ensure that you do not invalidate the only valid fast replication backup version.

When a copy pool is defined with fast reverse restore capability, the VERIFY(Y) keyword allows DFSMSHsm to determine whether fast reverse restore or regular FlashCopy can be used. For more information, see [“Recovering a copy pool using fast reverse restore” on page 278](#).

- **TSO usage considerations:** When you use commands that specify WAIT or NOWAIT, the results sent back to the user depend on whether the WAIT parameter is specified and the success or failure of one or more volumes:
 - If WAIT is specified and all volumes process successfully, a return code of zero is returned.
 - If WAIT is specified and one or more volumes fail, a nonzero return code and associated messages are returned.
 - If WAIT is not specified, control is returned to the system after the command has been accepted and verified. All messages continue to be issued, but you are not notified when the command completes.
- **Trying a copy pool recovery again:** During recovery of a copy pool, one or more volumes might fail to be recovered. In order to complete the recovery, you must identify and correct the cause of the failures. After correcting the cause of the failures, your options for completing recovery depend on the source of the recovery (DASD or tape) and if fast reverse restore was used.

- When the copy pool was recovered from DASD without using fast reverse restore (that is, the background copy completed before the recovery began), you can try to recover the individual volumes that failed, or try to recover the entire copy pool again.
 - If a small number of volumes failed recovery, issue a FRRECOV COPYPOOL(*cpname*) TOVOLUME(*volser*) command for each volume to try recovery of that volume again.
 - If FlashCopy version 2 or SnapShot is your fast replication utility, you can immediately try to recover the entire copy pool again.
 - If FlashCopy version 1 is your fast replication utility, you must wait for the FlashCopy relationships that were successfully established during the prior recovery to complete before you can try to recover the copy pool again.
- When the copy pool was recovered from DASD using fast reverse restore or the recovery was from tape, the recovery can be resumed. This is possible because the recovery status of each volume in the copy pool is tracked in the CDS records. The LIST COPYPOOL(*cpname*) command can be used to determine if a copy pool version previously failed recovery. If the recovery of a copy pool version failed, the LIST COPYPOOL output will indicate "RECOVERYINCOMPLETE=Y" and, if fast reverse restore was used, the fast replication state will be FCFRRINCOMPLETE. To resume recovery of the copy pool, issue a FRRECOV COPYPOOL(*cpname*) command specifying the same token, version, date, or generation as the failed command. The volumes that failed during the prior recovery are determined and a recovery is attempted for only those volumes that failed.

Note:

1. If the initial copy pool recovery was from a DASD version using fast reverse restore (the FASTREPLICATIONSTATE is FCFRRINCOMPLETE) and the recovery is resumed from a dump version, the default is to resume recovery of the remaining volumes. You can specify RESUME(NO) to perform a full recovery from tape. The copy pool DASD backup version will be invalidated when the copy pool recovery is completed successfully.
2. If the initial copy pool recovery was from a dump tape version and you are now trying to recover from the DASD version, a full recovery is performed.
3. The VTOC and the volume serial on the target volume may have changed as a result of the command. Before the volume can be accessed on any remote system, the UCB must be refreshed. The refresh occurs automatically if the volume is online and the device manager REFUCB function is enabled. You enable the REFUCB function through PARMLIB member DEVSUPxx or the MODIFY DEVMAN command. For more information, refer to the description of the REFUCB keyword in [z/OS MVS Initialization and Tuning Reference](#) or [z/OS MVS System Commands](#).

Related reading

- For more information about FRRECOV and the parameters for recovering fast replication backups, see Chapter 49, “FRRECOV command: Requesting a fast replication recovery,” on page 883.
- For more information about recovering a copy pool using fast reverse restore, see [“Recovering a copy pool using fast reverse restore”](#) on page 278.

Selecting a data set level recovery method

If a backup version resides on disk, DFSMSHsm can perform data set recovery through either fast replication or traditional copy methods. You can use the SETSYS command to select a preferred method for data set recoveries.

To indicate the method in which data set recoveries are to be performed, enter the SETSYS command, as the following example:

```
SETSYS FASTREPLICATION(DATASETRECOVERY(PREFERRED))
```

In this example, PREFERRED is specified to indicate that fast replication should be used whenever possible. For situations in which fast replication cannot be used to recover a data set, recovery is done

through traditional copy methods. You cannot use fast replication for recovery when the backup version's background copy for the volume that the data set resides on, has not yet completed.

To use only fast replication, specify `FASTREPLICATION(DATASETRECOVERY(REQUIRED))`. Here, if fast replication cannot be used, the recovery request fails. To use only traditional copy methods, specify `FASTREPLICATION(DATASETRECOVERY(NONE))`.

Note: The SETSYS setting can be overridden by specifying FASTREPLICATION on the FRRECOV command.

Data set level recovery performance considerations

The elapsed time for data set level recovery varies significantly based on the recovery method that is used.

- **FlashCopy from disk.** When FlashCopy is used to perform the recovery, the recovered data set is available for use in a matter of seconds. Note that FlashCopy is not the default method for performing a recovery, because the recovery creates a FlashCopy relationship between the backup volume and the production volume. This relationship prevents a backup of the copy pool from being created until the background copy has completed and the relationship is terminated. The SETSYS `FASTREPLICATION(DATASETRECOVERY(option))` command can be used to change the default.
- **Standard I/O from disk.** When the recovery is from disk, standard I/O is the default recovery technique. The elapsed time for the recovery varies depending on the size of the data set. Small data sets may complete in seconds, while larger data sets may take minutes to tens of minutes.
- **Standard I/O from tape.** When the recovery is from tape, the elapsed time is significantly longer. The increase in the length of time is due to the time that it takes to find the data set on tape. Since the backup copies are created as a full volume dump, DFSMSHsm does not record the file block ID of each data set. This results in DFSMSHsm having to read through the entire dump volume contents to find the data set before it can begin recovering the data set. The time of the search is longer for larger volumes. This means that Db2 object level recovery from tape takes significantly longer than a recovery from disk. It also takes significantly longer than an object level recovery from a Db2 image copy created on tape.

Making dump classes available for data set level recovery and volume recovery

When multiple dump copies have been created for a fast replication backup version, DFSMSHsm needs to know which dump copy to use during recovery. You can specify this with the optional parameter `FRRECOV(AVAILABLEFORMOUNT(YES|NO))` on the `DEFINE DUMPCLASS` command. The default value, `FRRECOV(AVAILABLEFORMOUNT(YES))`, indicates that dump copies created for the dump class are available for recovery and can be selected during FRRECOV processing. `FRRECOV(AVAILABLEFORMOUNT(NO))` indicates that dump copies created for this dump class are not immediately available for recovery and the dump class name must be specifically specified on the FRRECOV command with the `FROMDUMP(DUMPCLASS(dclass))` keyword to use this dump class. When multiple dump copies are created for a backup version and more than one of the dump classes have `FRRECOV(AVAILABLEFORMOUNT(YES))` specified, the first dump copy found with YES specified is used. If `FRRECOV(AVAILABLEFORMOUNT(NO))` has been specified for all of the dump classes, then the user must specify `DUMPCLASS` or `DUMPVOLUME` on the FRRECOV command, otherwise the command fails. The `DUMPCLASS` and `DUMPVOLUME` keywords of the FRRECOV command always take precedence over the `DUMPCLASS FRRECOV` setting.

This example shows how to use the `FRRECOV(AVAILABLEFORMOUNT)` keyword:

```
DEFINE DUMPCLASS(ONSITE FRRECOV(AVAILABLEFORMOUNT(YES)))
DEFINE DUMPCLASS(OFFSITE FRRECOV(AVAILABLEFORMOUNT(NO)))
```

In this example, dump class `ONSITE` represents a dump class whose copies are kept at the production site. Since these tape copies are immediately available for recovery, the dump class uses the option `AVAILABLEFORMOUNT(YES)`. Dump class `OFFSITE` represents a dump class whose copies are stored offsite. This dump class specifies `AVAILABLEFORMOUNT(NO)` since the tapes are kept offsite and are not immediately available for recovery. When an FRRECOV command is processed, it automatically selects

dump copies associated with dump class ONSITE. To use the dump copies associated with dump class OFFSITE, specify FROMDUMP(DUMPCCLASS(OFFSITE)) on the FRRECOV command.

This FRRECOV dump class setting applies to FRRECOV tape processing at the data set and volume level.

Recovering a copy pool using fast reverse restore

You can use the FRRECOV COPYPOOL(*cpname*) command to recover all volumes associated with the named copy pool defined with fast reverse restore capability. DFSMSHsm saves the copy pool fast reverse restore capability setting at the time of the fast replication backup. There are no additional keywords to specify to recover from a DASD copy pool fast reverse restore eligible version. If the state of the FlashCopy relationships meet the fast reverse restore requirements, DFSMSHsm uses fast reverse restore to recover the copy pool. Otherwise, DFSMSHsm will use regular FlashCopy if the background copy has completed.

Following a successful fast reverse restore, the contents of the DASD backup volume become invalid. DFSMSHsm invalidates and initializes the individual DASD backup volume so it is ready to be reused. When the entire copy pool has been successfully recovered, DFSMSHsm invalidates the DASD copy pool backup version.

If some of the volumes fail to recover, the FASTREPLICATIONSTATE of the DASD version becomes FCFRRINCOMPLETE. You can issue the FRRECOV COPYPOOL(*cpname*) command again after you have addressed the reason for the failure. DFSMSHsm will try to recover the remaining volumes again using fast reverse restore.

Using fast reverse restore does not require the removal of all other FlashCopy targets when cascaded FlashCopy is supported by the storage subsystems. For a successful fast reverse restore when multiple targets are not supported, verify there are not multiple FlashCopy targets before issuing the FRRECOV COPYPOOL(*cpname*) command.

To determine if a copy pool is defined to allow fast reverse restore and select a fast reverse restore eligible backup version, issue LIST COPYPOOL (*cpname*). If the copy pool is defined allowing fast reverse restore at time of backup, the output displays the backup version with: FCFRR=Y. Otherwise, the output contains: FCFRR=N.

Example: LIST COPYPOOL(CP1) will list information about the copy pool named CP1, which is defined to allow fast reverse restore.

```
LIST COPYPOOL(CP1) OUTDATASET
-- DFSMSHsm CONTROL DATASET --COPY POOL--LISTING -- AT 09:25:44 ON 09/03/07 FOR SYSTEM=1234
COPYPOOL=CP1
ALLOWPPRCP FRB=PN FRR=PN
VERSION  VTOCENQ    DATE        TIME        FASTREPLICATIONSTATE  DUMPSTATE
  002      Y      2009/03/06    16:35:24    RECOVERABLE          ALLCOMPLETE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=000002, INCREMENTAL=N, CATINFO=N, FCFRR=Y, RECOVERYINCOMPLETE=N
```

To determine if fast reverse restore can be used for a copy pool version, issue the QUERY COPYPOOL command. When fast reverse restore can be used, the QUERY COPYPOOL command output will display "background copy percent-complete" (PCT-COMP) information other than "***". Percent-complete information (a percentage) is available for full-volume FlashCopy pairs with an incomplete background copy only. A full-volume FlashCopy relationship is established when the FlashCopy technique (such as fast reverse restore or incremental) designates it, or when SETSYS FASTREPLICATION(FCRELATION(FULL)) has been specified.

Because fast reverse restore will invalidate the entire DASD backup version following a successful recovery, you can use the PCT-COMP percentage to determine how far along the background copy is and decide whether fast reverse restore should be used for recovery.

Note:

1. A DASD backup version can be used for fast reverse restore only once because the process invalidates the data on the backup volume. Keep dump tape copies in addition to DASD backup copies when you plan to use the fast reverse restore function.
2. VERIFY(N) cannot be specified when the copy pool is defined allowing fast reverse restore unless a previous FRRECOV COPYPOOL operation has determined the copy pool backup version is no longer eligible to use fast reverse restore. Regular fast replication recovery can be used instead.
3. When a recovery operation such as fast reverse restore from a DASD backup will invalidate an existing DASD copy that has an incomplete dump copy, the recovery operation fails. To force the use of a DASD copy with an incomplete dump copy, specify the FORCE parameter on the FRRECOV command.
4. The preserve mirror operation cannot be used in combination with fast reverse restore. Do not specify PMPREF or PMREQ in the **FRBACKUP to PPRC Primary Volumes allowed** or **FRRECOV to PPRC Primary Volumes allowed** fields for a copy pool defined allowing fast reverse restore.
5. Do not use the TOVOLUME operation to recover individual volumes in a copy pool defined with fast reverse restore capability.

Recovering from a copy pool incremental version

There are no additional keywords to recover from a DASD copy pool incremental version. When a volume from an incremental version or a copy pool incremental version is recovered, only the tracks that have changed since the time the backup version was created are copied back to the source volumes. After the background copy for the recovery is complete, the persistent incremental relationships are withdrawn. The copy pool still remains incremental, and the incremental relationships are re-established once the new incremental version is created. When the new backup copy is created for the incremental version, the FlashCopy incremental relationship is re-established by creating a full copy of each volume.

The following restrictions apply to recovering from an incremental version:

- The FlashCopy incremental relationship must be the only relationship when the storage subsystem does not support cascaded FlashCopy.
- To use reverse restore, the background copy must have completed.
- To use fast reverse restore, the background copy must be in progress.

When recovering from an incremental copy pool version, the FlashCopy incremental relationship must be the only relationship when cascaded FlashCopy is not supported by the storage subsystem. When unsupported, if there are multiple incremental FlashCopy relationships, the unwanted FlashCopy targets (backup versions) must be removed prior to attempting recovery. The FRBACKUP WITHDRAW or FRDELETE commands can be used to remove DFSMSHsm managed backups.

In a configuration where the storage subsystems do not support cascaded FlashCopy, when a non-incremental copy pool version is recovered while an incremental version exists, the FlashCopy incremental volume relationships are withdrawn if the background copies of the incremental volume relationships have completed. The incremental version remains recoverable. If the FlashCopy incremental background copies are still active at the time the non-incremental version recovery is requested, the request fails. If the FlashCopy incremental relationships for the incremental version are persistent, individual volume recovery using the TOVOLUME keyword from a non-incremental copy pool version fails regardless of completion of the background copy.

Using Metro Mirror primary volume during fast replication recovery

The FRRECOV command uses fast replication to recover one or more data sets, a single volume, or a pool of volumes from the managed backup versions. You can use the SMS copy pool field **FRRECOV to PPRC Primary Volumes allowed** to indicate to DFSMSHsm whether a Metro Mirror or Global Mirror primary volume is an eligible FlashCopy target, and whether a preserve mirror operation is required, preferred, or not desired when the target volume is a Metro Mirror primary volume during fast replication recovery. DFSMSHsm indicates this option to DFSMSdss for fast replication recovery.

If you did not specify a value in the **FRRECOV to PPRC Primary Volumes allowed** field in the SMS copy pool definition panel, you can specify the ALLOWPPRCP keyword on the FRRECOV command. Specifying

ALLOWPPRCP on the FRRECOV command overrides the currently stored preserve mirror indicator for the copy pool. The ALLOWPPRCP value specified on the FRRECOV command, however, is used only once during the FRRECOV operation and is not stored in the copy pool record. You can use the LIST COPYPOOL(cpname) command prior to recovery to determine the value of the ALLOWPPRCP indicator.

If Preserve Mirror is used in combination with incremental FlashCopy, the use of the PMPREF or PMREQ option needs to be consistent with the existing relationship when recovering from an incremental version. For example, if the existing incremental relationship was established without Preserve Mirror, it cannot recover with the PMREQ option because the remote relationship will not exist and, therefore, an incremental FlashCopy cannot be done at the remote site. Do not change your Preserve Mirror settings between increments.

Note:

1. If you do not specify ALLOWPPRCP on the FRRECOV command, DFSMSShsm uses the currently stored preserve mirror indicator for the copy pool. If no preserve mirror indicator has been set for the copy pool, the default is NO.
2. ALLOWPPRCP is ignored when recovering from a tape.
3. When you specify a value in the **FRRECOV to PPRC Primary Volumes allowed** field, DFSMSShsm considers the value's setting when selecting FlashCopy target volumes for FRBACKUP processing during volume pairing. DFSMSShsm does not check if the LO volumes are eligible for FlashCopy to Metro Mirror and Global Mirror primary or preserve mirror operation for recovery processing during volume pairing, even if a value is specified in the **FRRECOV to PPRC Primary Volumes allowed** field in the SMS copy pool definition. If preserve mirror is required during fast replication recovery using FlashCopy, you must ensure that your copy pool environment meets preserve mirror configuration requirements.
4. A Metro Mirror or Global Mirror primary volume is not permitted to be a FlashCopy target volume when:
 - NO is specified in the **FRRECOV to PPRC Primary Volumes allowed** field
 - the NO option is in effect by default
 - the capability is not supported by the storage controller
5. When you specify PN or PP in the **FRRECOV to PPRC Primary Volumes allowed** field in the SMS copy pool definition or with the YES, PRESERVEMIRRORNO, or PRESERVEMIRRORPREFERRED parameter of the ALLOWPPRCP parameter on the FRRECOV command, use SETSYS FASTREPLICATION(DATASETRECOVERY(NONE)) to prevent the source (LO) volume from going into a duplex pending state during the recovery of an individual data set from disk.
6. A Global Mirror primary volume is not allowed to be a FlashCopy target volume when:
 - NO is specified in the FRRECOV to PPRC Primary Volumes Allowed field
 - the NO option is in effect by default
 - the capability is not supported by the storage controller
 - DISABLE(FLASHCOPYTOGM) is specified in your DEVSUPxx parmlib member
 - the DISABLE(FLASHCOPYTOGM) is in effect by default

Related reading

- For more information about the IBM Remote Pair FlashCopy, Metro Mirror, also known as synchronous Peer-to-Peer Remote Copy (PPRC), and other copy services functions, see [z/OS DFSMS Advanced Copy Services](#).
- For more information about the ALLOWPPRCP keyword, see [“Syntax of the FRRECOV command” on page 884](#).

Using the XRC primary volume during fast replication recovery

The FRRECOV command uses fast replication to recover one or more data sets, a single volume, or a pool of volumes from the managed backup versions. You can use the SMS copy pool field FRRECOV to

XRC Primary Volumes allowed to indicate to DFSMSHsm whether an XRC primary volume is an eligible FlashCopy target. DFSMSHsm indicates this option to DFSMSdss for fast replication recovery.

If you did not specify a value (YES/NO) in the FRRECOV to XRC Primary Volumes allowed field in the SMS copy pool definition panel, you can specify the ALLOWXRCP keyword on the FRRECOV command. Specifying ALLOWXRCP on the FRRECOV command overrides the currently defined FlashCopy to XRC Primary indicator for the copy pool. The ALLOWXRCP value specified on the FRRECOV command, however, is used only once during the FRRECOV operation and is not stored in the copy pool record. You can determine the value of the current FRRECOV to XRC Primary Allowed indicator for the COPYPOOL using the ISMF COPY POOL panels.

The fast reverse restore option cannot be combined with the FlashCopy to XRC Primary option. Do not specify FRBACKUP to XRC Primary Volumes (YES) or FRRECOV to XRC Primary Volumes (YES) for a copy pool defined allowing fast reverse restore.

In a Metro/zOS GM environment, both XRC Primary YES and the PMREQ option must be set to allow FlashCopy to metro mirror and XRC primary volumes. See [z/OS DFSMS Advanced Copy Services](#) for more information.

Parallelism for fast replication

DFSMSHsm processes the FRRECOV COPYPOOL commands serially. To maximize throughput, DFSMSHsm recovers the fast replication source volumes from the target volumes in parallel. There are two levels of parallelism. The first level is that DFSMSHsm invokes DFSMSdss with the PARALLEL option and up to 254 volume pairs. The second level of parallelism is the number of DFSMSdss invocations that DFSMSHsm processes concurrently. DFSMSHsm processes between 1 and 64 DFSMSdss invocations concurrently, the default being 15. The number of concurrent DFSMSdss invocations that are processed and the number of parallel volumes passed to DFSMSdss can be modified with the SETSYS MAXCOPYPOOLTASKS(FRRECOV(*nn*) DSS(*nnn*)) command.

Parallelism for fast replication data set recovery

The default value for the maximum number of concurrent data set recovery tasks is 2. Two SETSYS keywords are defined to manage the maximum number of tasks for data set recoveries. SETSYS MAXDSRECOVERTASKS is used to specify the maximum number of data set recovery tasks from either DASD or tape. SETSYS MAXDSTAPERECOVERTASKS is used to specify the maximum number of data set recovery tasks specifically from tape. The maximum number of tasks for data set recoveries from tape should be limited to the number of available tape drives.

The following example shows that as many as four tape devices are available to be used for data set recovery.

```
SETSYS MAXDSTAPERECOVERTASKS(4)
```

This value must be less than or equal to the maximum number of allowable tasks for data set recoveries from DASD and tape.

To prevent any data set recoveries from tape, specify:

```
SETSYS MAXDSTAPERECOVERTASKS(0)
```

To enable up to 32 data set recovery tasks from both disk and tape, specify:

```
SETSYS MAXDSRECOVERTASKS(32)
```

Up to 64 tasks can be specified. To maximize throughput for fast replication volume recovery, use the SETSYS DSSXMMODE command to start DFSMSdss in a separate address space to process these requests. For more information about this address space and controlling it, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#) and [z/OS DFSMSHsm Implementation and Customization Guide](#).

Parallelism for fast replication volume recovery from tape

The default value for the maximum number of concurrent volume recovery from dump tasks is 1. A SETSYS parameter is defined to manage the maximum number of tasks for volume recovery from dump. SETSYS MAXDUMPRECOVERTASKS is used to specify the maximum number of concurrent volume recovery from dump tasks. The maximum level of parallelism for multi-tasked recovery is limited by the number of tape drives available and how volumes are stacked on the dump tapes. If all volumes being recovered are stacked on a single tape, then only a single recovery task can be started.

The following example shows that as many as 32 volume recovery from dump tasks can run concurrently:

```
SETSYS MAXDUMPRECOVERTASKS(32)
```

A maximum of 64 tasks can be specified. To reduce the time required for fast replication volume recoveries from dump, DFSMSHsm scans the dump volume recover queue for other volumes that are on the same tape before requesting the tape be dismounted. For more information about using the MAXDUMPRECOVERTASKS parameter, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.](#)

Withdrawing DFSMSHsm background copies

A FlashCopy relationship between a source volume and a target volume that was established through a FRBACKUP command should never be withdrawn outside of DFSMSHsm control. Doing so will invalidate the backup version, but DFSMSHsm will still manage the version as if it was valid. This may result in a data integrity exposure.

When it is necessary to withdraw one or more relationships established by a FRBACKUP command, you can use the following command:

```
FRBACKUP COPYPOOL(cpname) WITHDRAW
```

This command withdraws all current FlashCopy relationships associated with the volumes in the most recent VALID version for the specified copy pool. The version of the backup is marked as invalid unless it has a dump associated with it. In that case it becomes a dump only version.

If a dump is in progress for the version, then all incomplete dump copies for that version are marked as partial and are to be resumed. Any dumps completed before submission of the WITHDRAW command are kept and marked successful. Any volume that is currently being dumped when the WITHDRAW command is submitted is marked as a failure, though the dump is allowed to finish. Volumes that have yet to begin dumping at the time the command is submitted are marked as failed.

When a copy pool incremental version is withdrawn, the persistent relationships are withdrawn and the version is marked failed if the background copy was in progress. If the background copy is completed, the version remains VALID and recoverable.

If the most recent valid version is not generation zero and any relationships are withdrawn, then the withdrawn version is also deleted because it is no longer usable.

The withdraw can only be performed at the copy pool level. If any part of a copy pool backup version is invalidated, then the entire version is invalidated.



Attention: Use the LIST COPYPOOL command or the ARCXTRCT macro to ensure that you do not withdraw the only valid fast replication backup version. Withdrawing the only valid fast replication backup version will prevent recovery from being performed.

Note:

1. If a volume pair is withdrawn before DFSMSHsm completes the copy of the tracks that the target VTOC resides on, the target volume might become inaccessible. To avoid this, DFSMSHsm checks during WITHDRAW processing if a FlashCopy relationship exists on tracks where the VTOC resides, and invokes ICKDSF to initialize the volume. FlashCopy relationships on FlashCopy version 2 enabled volumes are withdrawn and the VTOC is recreated.

2. When a volume is initialized in a sysplex, only the host on which the volume was initialized is aware of the changes to the VTOC. You should vary off, and then on, the volumes identified in message ARC1838I to allow all hosts that share those volumes to recognize the changes to the VTOC. Message ARC1838I is most often issued when a volume is initialized in a sysplex using a copy pool defined with zero versions (NOCOPY mode).

Holding and releasing the fast replication functions

You can use the HOLD FRBACKUP and HOLD FRRECOV commands to hold and release the FRBACKUP function, as follows:

- **HOLD FRBACKUP.** This command indicates to DFSMSHsm that no new FRBACKUP commands are selected for processing.
- **HOLD FRRECOV.** This command indicates to DFSMSHsm that no new FRRECOV commands are selected for processing.

When you enter the HOLD FRBACKUP or HOLD FRRECOV command, all WAIT requests fail and all NOWAIT requests remain on the respective queue. Requests that have already been selected continue with normal processing.

The HOLD FRBACKUP command has a different effect if the host issuing the command is connected to a common dump queue (CDQ). New dump requests (any type from FRBACKUP, BACKVOL or AUTODUMP) from this host are submitted and processed by the CDQ, but the CDQ does not use the resources on this host to process them. If the host is the master scheduler in the CDQ, it is unable to perform scheduling of CDQ work. Consider using SETSYS MAXDUMPTASKS(0) from the master scheduler instead of this HOLD if you want the master scheduler to schedule CDQ work but not use its resources for CDQ dump processing.

You can use the RELEASE command from a host that is connected to a common dump queue to allow new a dump command, and auto dump volumes, to be selected and processed.

You can use HOLD DUMP and HOLD DUMP(FRBACKUP) commands to prevent any new dumps of FRBACKUP target volumes from being started by the FRBACKUP command.

You can use the HOLD FRRECOV and HOLD FRRECOV(TAPE) commands to prevent any new recoveries from a FRBACKUP generated dump. HOLD FRRECOV(TAPE) command only affects recovery from dump, but not from DASD backups.

You can use the HOLD FRRECOV(DATASET) command to hold data set recoveries from copy pool backup versions. HOLD RECOVER(TAPEDATASET) does not apply to the FRRECOV command.

You can use the RELEASE FRBACKUP and the RELEASE FRRECOV commands to release the holds that were placed on the respective functions.

You can use the RELEASE DUMP and RELEASE DUMP(FRBACKUP) to allow all FRBACKUP dump functions to be processed if no higher level HOLD is in effect.

You can use the RELEASE FRRECOV and RELEASE FRRECOV(TAPE) to allow all FRRECOV dump functions to be processed if no higher level HOLD is in effect.

You can use the RELEASE FRRECOV(DATASET) command to release only data set recoveries if they are not held at a higher level.

Deleting unneeded DFSMSHsm fast replication backup versions

Through normal processing, new backup versions replace older backup versions. There are two instances in which a process other than the normal roll-off case is required to delete an unneeded copy version. The first instance is when the number of versions specified for a copy pool is decreased. The second instance is when a copy pool is renamed or is no longer needed.

When the number of versions for a copy pool is decreased (by modifying the SMS copy pool definition), the next time that the FRBACKUP COPYPOOL command is issued for that copy pool, the unneeded versions are deleted. The target volumes that contained those versions are made available for other use.

When a copy pool is renamed or no longer needed, you must delete the backup versions that relate to that copy pool. You can use the following FRDELETE command to delete the backup and dump versions:

```
FRDELETE COPYPOOL(cpname) BOTH ALL
```

When you specify the VERSIONS keyword, the named backup versions are deleted. If you specify the TOKEN keyword, the version with the specified token is deleted. If you specify the ALL keyword, all backup versions are deleted. Before a version is deleted, any outstanding FlashCopy relationships are withdrawn. When a Fast Replication version is deleted, the associated catalog information data set is also deleted.

To limit your deletions to tape backups only, while preserving any DASD backups, specify the DUMPONLY keyword. The named backup versions are deleted only from tape dumps, but not from DASD backups. For example:

```
FRDELETE CP(COPYPOOL1) TOKEN(FRBACKUP_JUNEDATA) DUMPONLY
```

Here, DFSMSHsm deletes the dump copy of any fast replication backup for copy pool *COPYPOOL1* associated with token *FRBACKUP_JUNEDATA*. Omitting *DUMPONLY* from this command causes both tape and DASD backups associated with this token to be deleted.

You can also specify the DASDONLY keyword of the FRDELETE command to limit your deletions to DASD backups only.

When FRBACKUP fails because of an invalid request for an incremental version, the incremental version might need to be deleted to continue processing:

- Changing to a NOCOPY environment.
- Reduction in the number of copy pool versions necessitates an incremental version deletion.
- Current incremental version is not the oldest disk version and FCINCREMENTAL or FCINCREMENTALLAST is specified.

There are no additional keywords needed to delete an incremental version. Once an incremental version is deleted, the FCINCREMENTAL keyword must be used to re-establish the copy pool as having an incremental version.

When a Fast Replication version is deleted, the associated catalog information data set is also deleted. To limit your deletions to tape backups only, while preserving any DASD backups,

Related reading

For more information about FRDELETE and the parameters for deleting fast replication backups, see [Chapter 47, “FRDELETE command: Deleting fast replication backup or dump versions,” on page 873.](#)

Obtaining DFSMSHsm fast replication backup information

You can use the QUERY command, the LIST command, the REPORT command, and the ARCXTRCT macro to obtain information regarding the fast replication backup versions that DFSMSHsm is managing.

Using the QUERY command to obtain fast replication information

[Table 29 on page 285](#) describes the QUERY command parameters that you can use to obtain information about the fast replication functions.

Table 29. QUERY command parameters used to obtain information about fast replication functions

Parameter	Description of use
COPYPOOL(<i>cpname</i>)	<p>When you use the COPYPOOL(<i>cpname</i>) parameter, DFSMSHsm returns message ARC1820I for each version that has one or more source volumes in a FlashCopy relationship. If there are no outstanding relationships, DFSMSHsm returns the ARC1821I message. For FlashCopy version 2, the source volume can be in more than one relationship. When using FlashCopy version 1, this command can be used to determine if another backup version can be created. This command can also be used to determine when all FlashCopy relationships are complete, so that a recovery can be performed. When you query a copy pool with an incremental version, only the versions with an active background copy are listed.</p> <p>When the FlashCopy technique (such as fast reverse restore or incremental) establishes a full-volume FlashCopy relationship, or when SETSYS FASTREPLICATION(FCRELATION(FULL)) has been specified, message ARC1820I will display "background copy percent-complete information" (PCT-COMP) for each full-volume FlashCopy pair. Three asterisks ("***") indicate percent-complete information is not available. The percent-complete information (the percentage) can be used to determine how far along the background copy is. Depending on the percent-complete information, you can decide whether to use fast reverse restore immediately, and reduce recovery time, or wait for the background copy to complete in order to use regular fast replication recovery and preserve the DASD backup version.</p> <p>If SnapShot is the fast replication utility, then the QUERY COPYPOOL(<i>cpname</i>) command is of no value. If you issue this command, DFSMSHsm will indicate that there are no outstanding FlashCopy relationships.</p>
ACTIVE	<p>When you use the ACTIVE parameter, DFSMSHsm lists two messages that relate to fast replication. Message ARC1822I describes an active FRBACKUP COPYPOOL or FRRECOV COPYPOOL command. Message ARC1826I shows whether FRRECOV is HELD at the level of EOD or EOV, shows whether FRBACKUP DUMP is HELD, ACTIVE, or both, and shows whether FRRECOVER(DATASET) is HELD, ACTIVE, or both. Message ARC0162I reports active fast replication data set recovery tasks.</p> <p>Note: If FRBACKUP DUMP is active, it could indicate that FRBACKUP DUMP command or FRBACKUP DUMPPONLY command is active. FRBACKUP dump could run several hours before it comes to the end. Multiple FRBACKUP DUMP MWEs may still be running after FRBACKUP MWE is purged, therefore, whether FRBACKUP DUMP is active does not necessarily mean that FRBACKUP command is active. If FRRECOVER(DATASET) is active, it indicates that FRRECOV DSNAME(<i>dsname</i>) is being processed.</p>
AUTOPROGRESS	<p>Message ARC1824I shows the number of copy pools that have not been completed by the active automatic function, and the total number of copy pools that are eligible to be processed. This message also shows the affinity and non-affinity copy pool count. Message ARC1828I shows the number of volumes for a specific copy pool that have not been processed, and the total number of volumes for this specific copy pool that are eligible to be processed by the active automatic function. ARC1828I also shows the affinity and non-affinity volumes count for the specific copy pool.</p>

Table 29. QUERY command parameters used to obtain information about fast replication functions (continued)	
Parameter	Description of use
SETSYS	When you use the SETSYS parameter, message ARC1823I indicates the current COPYPOOL settings for backup and recovery, and FASTREPLICATION settings. Message ARC0274I indicates the MAXDSRECOVERTASKS and MAXDSTAPERECOVERTASKS value.
STATISTICS	Individual data set recoveries from FRRECOV commands are reported together with data set restores from RECOVER commands in message ARC0641I. The number of FRRECOV DATASET commands are not recorded.
DATASET, USER, or REQUEST	When you use either the DATASET, USER, or REQUEST parameters, DFSMSHsm lists four messages that relate to fast replication. Messages ARC0162I and ARC0167I report on both the main FRRECOV DSNAME MWE and the individual data set recoveries that are derived from a FRRECOV command. Message ARC0167I displays VOLUME volser, COPY POOL= <i>cpname</i> waiting to be processed, data set name, and the number of MWES ahead of the present one. Message ARC1822I displays FRBACKUP DUMP or DUMPONLY MWE from COPY POOL <i>cpname</i> processed and waiting to be processed. The TOTAL number of MWES indicates the number of volume MWEs within same FRBACKUP command. Message ARC0161I displays the volser, copy pool name, user and request number for a volume being dumped as part of a copy pool.
WAITING	When you use the WAITING parameter, DFSMSHsm issues message ARC0168I to indicate the number of DFSMSHsm commands waiting to be processed. The fast replication backup commands are reported in the FRBACKUP= <i>nfrbackup</i> field. The individual dump commands derived from an FRBACKUP DUMP or FRBACKUP DUMPONLY command are included in the BACKUP= <i>nbackup</i> field. The fast replication recover commands are reported in the FRRECOV= <i>nfrrecov</i> field. The individual data set recover commands derived from an FRRECOV DSNAME command are included in the RECOVER= <i>nrecover</i> field.

Using the LIST command to obtain fast replication information

Table 30 on page 286 describes the LIST command parameters that you can use to obtain information about the fast replication functions.

Table 30. LIST command parameters used to obtain information about fast replication functions	
Parameter	Description of use
COPYPOOL	When you use the COPYPOOL parameter, with no copy pool name, DFSMSHsm returns version information for each copy pool that has at least one attempted backup. This information includes the copy pool name, version number, valid or invalid, VTOCENQ, token, backup date, and time. Source and target volume pairs are not listed. Information about whether a version has a dump, or DASD copy, or both, is also included. The SELECT keyword allows the user to limit the scope of output based on the state of the FRBACKUP or FRDUMP.

Table 30. LIST command parameters used to obtain information about fast replication functions (continued)	
Parameter	Description of use
COPYPOOL(<i>cpname</i>)	<p>When you use the COPYPOOL(<i>cpname</i>) parameter, DFSMSHsm returns detailed version information for each backup version for the specified copy pool. This information includes the copy pool name, version number, valid or invalid, VTOCENQ, token, backup date, and time. Source and target volume pairs are included in this list. (The date and time fields are zero for "Prepare" versions). COPYPOOL(<i>cpname</i>) displays the most recent ALLOWPPRCP settings for the copy pool as follows:</p> <p>NO is displayed for the NO option.</p> <p>PN is displayed for the PRESERVEMIRRORNO or PMNO option.</p> <p>PP is displayed for the PRESERVEMIRRORPREFERRED or PMPREF option.</p> <p>PR is displayed for the PRESERVEMIRRORREQUIRED or PMREQ option.</p>
COPYPOOL(<i>cpname</i>) ALLVOLS(TOKEN(<i>token</i>))	When you use the ALLVOLS(TOKEN(<i>token</i>)) parameter, DFSMSHsm limits the LIST information to the version or versions associated with a specific TOKEN request.
DATASETS GENERATION(<i>gennum</i>)	Specifies that DFSMSHsm is to list data sets associated with a particular generation. The default, DATASETS GENERATION(0), lists only the most recent DASD and dump copies.
DATASETS TOKEN(<i>token</i>)	Specifies that DFSMSHsm is to list data sets associated with the version with the specified token.
DUMPCLASS	When you use the DUMPCLASS parameter, DFSMSHsm returns information regarding whether the dump class associated with the copy pool is required or optional.
DVOL	When you use the DVOL parameter, DFSMSHsm returns information regarding whether the volume resides in a copy pool.
HOST	When you use the HOST parameter, DFSMSHsm lists the host holding a copy pool resource. If subparameter RESET is specified, DFSMSHsm resets associated serialization fields.
PVOL	When you use the PVOL parameter, with either the BCDS or BOTH subparameters, DFSMSHsm returns information about which copy pools the volume is associated with. With ALLDUMPS or BCDS subparameters, should also consider copy pool dumps. Copy pool dumps versions should include the name of the source copy pool. If the volume is in a copy pool, associated information needs to be displayed. Target volumes of a copy pool source/target pair does not display information. To determine which target volumes are associated with a particular source volume, you can use the LIST COPYPOOL(<i>cpname</i>) command.

Using the REPORT command to obtain fast replication information

Table 31 on page 288 describes the REPORT command parameters that you can use to obtain information about the fast replication functions.

Table 31. *REPORT* command parameters used to obtain information about fast replication functions

Parameter	Description of use
DAILY	When you use the DAILY parameter, DFSMSHsm reports the following: <ul style="list-style-type: none"> • The total number of requested fast replication volume backups. • The total number of fast replication volume backups that failed. • The total number of requested fast replication volume recovers. • The total number of fast replication volume recovers that failed.
VOLUMES	When you use the VOLUMES parameter, DFSMSHsm reports the following: <ul style="list-style-type: none"> • The total number of times a fast replication backup was attempted for the volume. • The total number of failed backup attempts for that volume. • The total number of times a fast replication recover was attempted for the volume. • The total number of failed recover attempts for that volume.

Note: Statistics for dump copies created for copy pool backup versions are reported with existing report categories for dump and recovery. Also, statistics for data set recoveries from copy pool backup versions are reported with the existing data set restores from dump copies. Note that bytes/tracks written statistics are not recorded for data set recoveries from copy pool backup versions.

Using the ARCXTRCT macro to obtain fast replication information

You can use the ARCXTRCT macro to obtain information about the fast replication function. When you use the data type of COPYPOOL and the COPYPOOL keyword with its associated subparameter, the ARCXTRCT macro returns information for each version of a specified copy pool. This information includes the version number, generation number, valid or invalid indicator, incremental indicator, date, time, token, dump class name, and other dump class information.

Preventing DFSMSHsm functions from processing target volumes

The DFSMSHsm automatic functions (Primary Space Management, Autobackup, and Interval Migration) only process pool type storage groups. These automatic functions do not process copy pool backup storage group types.

The BACKVOL STORAGEGROUP command only processes pool type storage groups. This BACKVOL command will not process copy pool backup storage groups.

The BACKVOL VOLUME and the MIGRATE VOLUME commands will not process any volumes that reside in a copy pool backup storage group.

Other copy pool considerations

You might encounter the following situations as you use copy pools and fast replication methods.

Migration of data sets within a copy pool

DFSMSHsm *does not* support the migration of data sets within a copy pool, although it is not explicitly prevented. Recovering a copy pool or a volume within a copy pool from which data was migrated will cause data loss.

Data set recovery errors

A data set that fails recovery remains cataloged, but its integrity is unknown. It is your responsibility to ensure that a data set that fails recovery is not used for processing. When one or more pieces of a

multivolume data set fail, the failed pieces are listed as a part of message ARC1861I. The successful pieces are not listed as a part of message ARC1860I.

Maintaining catalogs with fast replication backup

Because a fast replication backup is a physical copy of the data, you should take special care to ensure that the catalogs stay synchronized with the data. Catalogs should be copied with the data that they represent so that both can be recovered to the same point-in-time.

If it becomes necessary for you to recover an individual volume and not the corresponding catalog, you must be aware that the data on the volume will not be synchronized with the catalog and the following may occur:

- Data sets may exist on the volume but not in the catalog.
- The catalog may have entries for data sets on the volume that are not actually present.
- Data set information for existing data sets in the catalog may be inaccurate.

Increasing the number of volumes in a copy pool

When one or more volumes are added to a storage group within a copy pool, DFSMSHsm includes those new volumes with any subsequent backup versions. If you must recover a version that does not contain the new volumes, and data has been placed onto those volumes, after the recovery, those volumes will contain residual data that is unusable and must be cleaned up.

Example: If you have a storage group that has 2 volumes (volumes A and B) and you request a fast replication of those two volumes. Later on, you add a third volume (volume C) to the storage group. New data is then placed onto volume C. You discover that an error has occurred and that you need to recover the last backup version. During the recovery, volumes A and B are recovered to the point of the fast replication. Because volume C was not a part of the original fast replication backup, it is not processed during the recovery. Volume C may contain data that is no longer valid and you will need to review this data on the volume. DFSMSHsm issues message ARC1810I for each volume that fits this scenario.

Decreasing the number of volumes in a copy pool

Removing a volume from a storage group defined to a copy pool Special care must be taken when removing a volume from a storage group that is included in one or more copy pool definitions. When a volume is removed from a storage group, it will no longer be included in subsequent backup copies, but it will still be required to be available for copy pool recovery operations. Copy pool recovery is a volume to volume operation and requires all of the original volumes to be available at the time of the recovery. A volume that is removed from a storage group should not be reused for data set allocations until there are no remaining copy pool backup versions that include that volume. An exception to this rule are copy pools for which only data set level recoveries will be performed. In this case, a volume can be removed and reused immediately elsewhere in the configuration because data set level recovery performs volume selection and can recover a data set to a different volume than the original.

The LIST COPYPOOL(*cpname*) command can be used to determine which volumes are included in each copy pool backup version. When the removed volume is no longer returned in the output of this LIST command, it is no longer included in a backup version and may be used elsewhere in the environment.

When a copy pool recovery is performed, as each volume is recovered, DFSMSHsm will verify that the volume is still defined to the same storage group within the copy pool as it existed at the time of the backup. Any volume that fails this verification will fail to be recovered. This prevents the recovery process from overlaying any data that is written to the volume after it was removed from the copy pool. Any valid data that exists on a volume that failed recovery must be moved to a different volume. Once there is no more valid data on the volume, the volume must be redefined to the original storage group within the copy pool. After this has been completed for each volume that failed, the entire copy pool may be recovered once again, or just the individual volumes that failed may be recovered using the TOVOLUME keyword.

Example: Copy pool CPA is defined with one storage group, SGA. SGA is comprised of 2 volumes, VOLA and VOLB. Copy pool CPA is backed up nightly to disk and tape. After the disk backup completes, the disk

backup is dumped to tape. For this copy pool, one backup copy is maintained on disk and seven backup copies are maintained on tape.

A single, larger volume, VOLC is added to SGA within the copy pool. All of the data is moved from volumes VOLA and VOLB to single volume VOLC. VOLA and VOLB are removed from the storage group definition for SGA.

Before the next backup copy is created, there are 8 backup versions that contain volumes VOLA and VOLB. If one of these versions need to be recovered, then the recovery will require volumes VOLA and VOLB to be returned to storage group SGA.

These are the steps that can be taken:

1. If VOLA and VOLB have not been reused at all, as is recommended, redefine them to storage group SGA before the recovery. The copy pool recovery will be successful.
2. If VOLA and VOLB are not returned to the storage group before the copy pool is recovered, each volume will fail with message ARC1811E. Remove any valid data from the volumes, redefine them to storage group SGA and retry the recovery.
3. In the case of a larger copy pool definition where most volume recoveries are successful and only a small number of volumes fail with the ARC1811E message, each of the failed volumes can be recovered individually after redefining them to the original storage group:

```
FRRECOV TOVOLUME(volser) FROMCOPYPOOL(cpname) VERSION(vernum)
```

FROMCOPYPOOL and VERSION are optional keywords used in this example. (If fast reverse restore was used for the recovery, there is no need to selectively recover just the failed volumes. DFSMSHsm knows which volumes failed and will selectively retry only those volumes if the FRRECOV COPYPOOL command is reissued).

4. In all cases, VOLC should be reinitialized as part of the recovery process since the data on it is no longer valid.

As new backup copies are created, the newer backup copies will contain only volume VOLC. If one of these versions need to be recovered, no additional action is required because the active configuration is what existed at the time of the backup.

After the single disk backup and all seven tape backup copies that contained volumes VOLA and VOLB have been replaced with newer backup copies that contain only VOLC, then volumes VOLA and VOLB can be used elsewhere in the system configuration.

Decreasing the number of volumes in a copy pool backup storage group

You must take special care when removing a volume from a copy pool backup storage group. Before doing so, you **must** ensure that the volume is not the target of a valid DFSMSHsm backup version. You can verify this by examining the output from the DFSMSHsm LIST COPYPOOLBACKUPSTORAGEGROUP command.

If a target volume needs to be removed from a valid source to target pairing, you should use the FRDELETE command to delete the DFSMSHsm backup version that the volume is a part of before you remove the volume.

Defining a storage group to multiple copy pools

Although you may include a storage group in multiple copy pools, IBM recommends against doing this when FlashCopy version 1 is used as your the fast replication utility. If the FRBACKUP COPYPOOL command is issued for a copy pool with a common storage group, other copy pools with that same common storage group cannot be backed up until the background copies for all of the volumes in the common storage group have completed. Even though the first FRBACKUP COPYPOOL command logically completes in a manner of minutes, allowing the subsequent FRBACKUP commands to be issued, the subsequent FRBACKUP commands will fail until all of the background copies are complete, which might take several hours. This prevents the volumes in the noncommon storage groups in those copy pools from having a timely backup created.

FlashCopy version 2 and SnapShot do not have this restriction.

Using multivolume data sets in a fast replication environment

You must give special consideration to multivolume data sets in a fast replication environment. Because VTOC enqueues are obtained as each volume is processed, it is possible for a multivolume data set to be updated on one of its volumes while the other volume is being fast replicated. To prevent this from occurring, you should ensure that data set activity for multivolume data sets is quiesced during the fast replication process.

When recovering multivolume data sets, DFSMSHsm creates individual data set recovery commands to recover each piece of the data set from each volume. After all of the individual pieces have completed processing, DFSMSHsm determines if the multivolume data set has been recovered successfully or not. To determine if a multivolume data set recovered successfully, look for messages ARC1863I, ARC1864I and ARC1865I. It is possible that individual pieces recover successfully, while the data set as a whole fails. It is also possible for individual pieces to fail, while the data set as a whole is successful (This occurs if the data set has extended to one of more volumes that it does not reside on at the time of the backup. DFSMSHsm determines that the data set has grown, scratch the space off of the volumes that the data set extended to after the backup, and recatalog the data set onto only the original volumes).

Multivolume VSAM data sets with AIXs fail with message ARC1866I RC16. In order to recover the VSAM data set, all associated AIXs must be deleted. You can then recover the data set and recreate the AIXs. If a multivolume data set that has changed sizes cannot be recataloged, message ARC1865I lists the volumes to which the user should recatalog the data set.

Using the APPLYINCREMENTAL keyword

The APPLYINCREMENTAL (AINC) keyword of the FRRECOV command is *only* valid with the TOVOLUME keyword. This keyword specifies that an incremental volume recovery process immediately follows after the volume restore, and is part of the single request to recover a specific volume. APPLYINCREMENTAL is supported on eligible dump volumes created by the FRB DUMP command or automatic dump.

Using an alternate copy pool backup storage group

For the FRBACKUP command, you can override the copy pool backup storage group defined in the pool storage groups with an alternate storage group by specifying the BACKUPSTORAGEGROUP(*bsgname*) keyword with a copy pool backup storage group (CPBSG) name. For example:

```
FRBACKUP CP(cpname) BACKUPSTORAGEGROUP(bsgname)
```

When you specify the BSG keyword, there cannot be any existing DASD backup versions for the copy pool.

When the keyword is specified, DFSMSHsm will override the CPBSG defined in the pool storage groups with the alternate CPBSG. Once the override has been established, DFSMSHsm will continue to use the alternate CPBSG for subsequent FRBACKUP commands even if the BSG keyword is not present.

To reset fast replication backup to the original CPBSG, issue the FRBACKUP command with the BSG keyword without specifying a CPBSG name. As with setting up an override, no existing fast replication DASD backup versions for the copy pool are allowed. For example:

```
FRBACKUP CP(cpname) BACKUPSTORAGEGROUP
```

Defining an override copy pool backup storage group

The storage group type should be defined as a COPY POOL BACKUP.

Collecting fast replication messages in message data sets

In order to simplify problem determination in the event of an error, DFSMSHsm can collect all DFSMSHsm and DFSMSdss messages issued for an FRBACKUP and FRRECOV COPYPOOL request, auto dump of a copy pool and FRRECOV TOVOLUME FROMDUMP in a dynamically allocated, unique message data

set. Fast replication messages are additionally written to activity logs. To enable the collection of fast replication messages in message data sets, issue the SETSYS FASTREPLICATION(MESSAGE DATASET(YES HLQ(*hlq*))) command. For more information, refer to Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.

Message data set allocation

Fast replication message data sets must be SMS-managed. DFSMSHsm allocates the physical sequential data sets with these attributes:

Record Format

FBA

LRECL

121

Block size

1210

System defaults for primary and secondary space, as specified in ALLOCxx, will be used. To override this, define a data class that specifies appropriate space attributes and modify the ACS routine to assign the data class to DFSMSHsm message data sets.

Message data set names have the following format:

- For FRBACKUP requests: *msgdsprefix.FB.Ccccccc.ippppppp.Dyyddd.Thhmmss.z*
- For FRRECOV requests: *msgdsprefix.FR.Ccccccc.ippppppp.Dyyddd.Thhmmss.z*
- For FRRECOV TOVOLUME requests: *msgdsprefix.FR.Vvolser.Dyyddd.Thhmmss.z*

where:

msgdsprefix

is the value of the HLQ parameter of the SETSYS FASTREPLICATION(MESSAGE DATASET) command, if specified. If the HLQ parameter is not specified, the default prefix of 'HSMMMSG' is used.

cccccc

is up to 7 characters to identify the copy pool that was processed. For a non-Db2 copy pool, this will be the first 7 characters of the copy pool name. For a Db2 copy pool, this will be the first 7 characters of the location qualifier in the Db2 copy pool name.

Underscore characters in the copy pool name will be replaced with a hyphen to satisfy MVS data set naming convention.

i

is a 1-character identifier, indicating whether the message data set contains messages regarding:

D

Db2 data

L

Db2 log

X

non-Db2 copy pool data

ppppppp

is up to seven characters to identify the copy pool that was processed. For a non-Db2 copy pool, this is the next seven characters of the copy pool name (the characters following ccccccc). For a Db2 copy pool, this is the next seven characters of the location qualifier in the Db2 copy pool name.

volser

is the volser of the volume that was recovered.

yyddd

is the year and day when the FRBACKUP, FRRECOV, or copy pool auto dump operation started.

hhmmss

is the time in hours, minutes, and seconds when the FRBACKUP, FRRECOV, or copy pool auto dump operation started.

z

indicates the status of the FRBACKUP, FRRECOV, or copy pool auto dump operation, and is set to one of the following:

S

Success. The operation completed successfully.

F

Fail. One or more errors occurred while processing the operation.

I

In progress. DFSMSHsm allocates the data set with this final qualifier, and renames the data set with one of the other qualifiers once the operation completes

When DFSMSHsm attempts to allocate a message data set and a data set with the selected name already exists, DFSMSHsm increments the time stamp in the data set name by one second and attempts to allocate with this new name. DFSMSHsm renames the data set up to a maximum of 10 times, and if a unique message data set name is not found, an ARC1873I message is issued and messages are not captured for this request.

You can specify the date and time used in the message data set name with the MESSAGEDATASET parameter of the FRBACKUP command and the FRRECOV COPYPOOL command. Refer to [Chapter 46, “FRBACKUP command: Requesting a fast replication backup or dump version,”](#) on page 865 and [Chapter 49, “FRRECOV command: Requesting a fast replication recovery,”](#) on page 883.

Message data set contents

DFSMSHsm allocates a message data set for a request once the FRBACKUP, FRRECOV or auto dump operation has been verified as eligible for processing. Therefore, messages indicating parse errors or termination prior to the ARC1801I or ARC1841I start message being issued are not placed in a message data set.

DFSMSHsm prefixes each message written to the message data set with the date and time that the message was written to the data set, along with the ID of the host that wrote the message. The following format is used: *yy/mm/dd hh:mm:ss Hx message-text* where *x* is the host ID. Some messages appear multiple times in a message data set.

Message data set example

This command is issued:

```
FRBACKUP COPYPOOL(DSN$PAYROLLA123$DB) DUMP
```

DFSMSHsm performs an FRBACKUP DUMP of Db2 copy pool DSN\$PAYROLLA123\$DB using the common dump queue. Message data set support has been enabled, so DFSMSHsm allocates message data set HSMMSG.FB.CPAYROLL.DA123.D14325.T100910.S.

```
14/11/21 10:09:10 H2 ARC1801I FAST REPLICATION BACKUP DUMP IS STARTING FOR COPY POOL DSN$PAYROLLA123$DB, AT 10:09:10
ON 2014/11/21
14/11/21 10:09:10 H2 ARC0640I ARCFRTM - PAGE 0001 5695-DF175 DFSMSDSS V2R02.0 DATA SET SERVICES 2014.325
10:09
14/11/21 10:09:10 H2 ARC0640I ARCFRTM - ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK
DEFAULT TO YES
14/11/21 10:09:10 H2 ARC0640I ARCFRTM - PARALLEL
...
14/11/21 10:09:11 H2 ARC1805I THE FOLLOWING 00003 VOLUME(S) WERE SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP
OF COPY POOL DSN$PAYROLLA123$DB
14/11/21 10:09:11 H2 ARC1805I (CONT.) SRC001
14/11/21 10:09:11 H2 ARC1805I (CONT.) SRC003
14/11/21 10:09:11 H2 ARC1805I (CONT.) SRC002
.
.
.
```


Volume recovery and restore

Volume recovery refers to the process of recovering a level 0 volume to its original condition as of a specified date. Volume recovery is typically necessary only after a complete volume has been damaged.

Four methods exist for DFSMSHsm volume recovery:

- Full-volume restore with update from incremental backup versions
- Full-volume restore
- Volume recovery from incremental backup versions
- Volume recovery with the fast replication function

Because full-volume restore with update is somewhat a combination of the other two methods, and therefore the most complete, it is discussed first.

When recovering volumes by using backup versions either for update or as the only source, DFSMSHsm provides optimized tape usage. The optimization is accomplished by recovering all required data sets from each individual tape in priority order with a single tape mount.

When recovering volumes by using backup versions either for update (APPLYINCREMENTAL) or as the only source, DFSMSHsm recovers all cataloged non-VSAM data sets—whether they are SMS-managed—and all cataloged VSAM non-SMS-managed data sets to the volume being recovered. DFSMSHsm may recover VSAM SMS-managed data sets to other volumes in the same storage group as the volume being recovered.

The procedures for recovering an ML1 volume differ from those for recovering a level 0 volume. See [“Recovering a migration volume”](#) on page 552 for a discussion of the procedures for ML1 volumes.



Attention: DFSMSHsm incremental recovery and volume restore with APPLYINCREMENTAL does not support a device type change on the volume being recovered. If the device type is changed after the volume is backed up and if the device type in the catalog for all data sets residing on the volume is not changed, all VSAM data sets will fail to be recovered. All non-VSAM data sets may appear to be recovered successfully, but the first allocation after the recovery will fail because of the inconsistent unit name and device type that were created by the recovery.

Specifying the maximum number of data set recovery tasks in each host

In recovering from incremental backup versions (as the only source or during APPLYINCREMENTAL processing after a volume restore), DFSMSHsm can recover multiple data sets in parallel.

The MAXDSRECOVERTASKS parameter of the SETSYS command controls the number of concurrent data set recovery tasks (to a maximum of 64) that should be started for both DASD and tape; and the MAXDSTAPERECOVERTASKS parameter of the SETSYS command specifies the maximum number of concurrent data set recovery tasks that should be started for tape. To decide how many concurrent tasks DFSMSHsm should run, consider how many tape drives are available if you use tape backup volumes.

MAXDSTAPERECOVERTASKS is an optional SETSYS keyword. SETSYS MAXDSTAPERECOVERTASKS(0) should be specified if you do not want any data set recoveries from tape. The default value of MAXDSTAPERECOVERTASKS is the value of MAXDSRECOVERTASKS, and cannot exceed the value specified for MAXDSRECOVERTASKS. This requires you to specify MAXDSRECOVERTASKS before MAXDSTAPERECOVERTASKS, if the value for MAXDSTAPERECOVERTASKS is greater than the default value for MAXDSRECOVERTASKS.

Specifying the maximum number of volume recovery from dump tasks

DFSMSHsm can process up to 64 volume recovery from dump tasks concurrently. The MAXDUMPRECOVERTASKS parameter of the SETSYS command defines the maximum number of volume recovery from dump tasks that can be concurrently processed. The default value is 1. A value greater than 1 must be specified to allow multi-tasked volume recovery from dump tapes. The MAXDUMPRECOVERTASKS value specified becomes the maximum number of simultaneous volume recovery tasks that can be processed using the FRRECOV and RECOVER commands. When recovering

volumes without using fast replication, multiple RECOVER commands must be issued to take advantage of multi-tasked volume recovery from dump.

For example, the following SETSYS command specifies that a maximum of 64 volume recovery from dump tasks can be processed concurrently:

```
SETSYS MAXDUMPRECOVERTASKS(64)
```

In the above example, the maximum number of concurrent volume recovery from dump tasks is 64. This allows you to issue multiple RECOVER commands and that those commands can be processed concurrently (a maximum of 64 concurrent volume recovery tasks). In addition, it also allows a single FRRECOV command to initiate up to 64 concurrent recovery operations from tape.

To decide how many concurrent volume recovery from dump tasks DFSMSHsm should process, consider how many tape drives are available. If any dump class involved is defined with a STACK value of more than 1, DFSMSHsm writes multiple dump copies sequentially as each tape is mounted for that dump class. The maximum level of parallelism for multi-tasked recovery is limited by how volumes are stacked on the dump tapes. If all volumes being recovered are stacked on a single tape, then only a single recovery task will be started for those volumes. The reduced parallelism might lengthen the total time needed for the volume recovery to complete.

When the SETSYS COMMONQUEUE(RECOVER(CONNECT(*basename*))) command is specified, the preference by the system will be to fully utilize the available recover tasks as defined by MAXDUMPRECOVERTASKS on all of the CVQ group members in the HSMplex. Note that when this SETSYS is *not* specified, only local tasks where the command was issued will be utilized. This reduced parallelism might shorten the total time necessary for the recover operations to complete. In either case, the number of concurrent tasks is further limited by the stacking of the volumes on the dump tapes. For example, recovering 100 volumes from a single tape uses only a single recover task.

If you want a common recover host to be able to submit volume restore work, but not process the work, issue the SETSYS MAXDUMPRECOVERTASKS(0) command on that host. This can be either a master scheduler host or a non-master scheduler host. In the case of a master scheduler host, this does not impact the master scheduler duties for the common recover queue.

Related reading

- For more information about using the MAXDUMPRECOVERTASKS parameter, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,”](#) on page 1057.
- For more information about using the FRRECOV command, see [Chapter 49, “FRRECOV command: Requesting a fast replication recovery,”](#) on page 883.
- For more information about using the RECOVER command, see [Chapter 60, “RECOVER command: Recovering a backup version or dump copy of a data set, or UNIX file, or a volume,”](#) on page 997.
- For more information about parallelism for fast replication volume recovery, see [“Parallelism for fast replication volume recovery from tape”](#) on page 282.
- For more information about recovering SMS-managed data sets, see [“Restoring a data set from a DFSMSHsm dump copy”](#) on page 251.
- For more information about recovering non-SMS-managed data sets, see [“Data set restore from a DFSMSHsm dump copy”](#) on page 314.

Full-volume restore with update



Attention: Most of the DASD that is installed today is permanently mounted (for example, the 3380 or 3390 DASD devices). In addition, DASD can be either online or offline. Therefore, when you are recovering a volume with full-volume restore and incremental backup versions, a possibility exists that one of your users could begin using a data set in the interval between the restore and the incremental recovery. This can be avoided by putting the SMS-managed volume in DISALL status. There are two considerations you need to be aware of before placing the volume in DISALL status:

- With DISALL status, some data sets may be recovered (as directed by your ACS routines) to different volumes in the storage group containing the volume being recovered.
- If you are recovering a catalog, or a volume belonging to a storage group using the guaranteed space attribute, the volume must be in enabled (ENABLE) status during the recovery process.

Ensure that the SMS-managed volume is in disabled (DISALL) status before issuing the RECOVER command, unless you are recovering a catalog, or a volume from a storage group using the guaranteed space attribute.

After the recovery process has completed, be sure to review the messages for failures or data sets that were not recovered. For example, if the volume you are recovering contains part of a multivolume data set, that partial data set is not recovered. After the recovery process has completed, the data set is listed as having not been recovered, giving you the opportunity to recover the entire data set, including the parts of the data set that reside on volumes not being recovered. You can issue individual RECOVER *dsn* commands for data sets such as these.

You can use the following commands to combine full-volume restore with full-volume recovery.

```
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP(DUMPVOLUME(tape_volser) APPLYINCREMENTAL)  
  
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP(DUMPCCLASS(class) APPLYINCREMENTAL)  
  
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP(DUMPGENERATION(dgennum) APPLYINCREMENTAL)
```

When APPLYINCREMENTAL is specified on a RECOVER FROMDUMP command, DFSMSHsm first performs a full-volume restore as described in [“Full-volume restore from a dump copy” on page 298](#). After the volume restore has been completed, DFSMSHsm performs the following steps:

1. If integrated catalog facility catalogs reside on the volume, DFSMSHsm recovers any catalog for which a backup version exists that was created following the dump. All such catalogs must be successfully recovered for the APPLYINCREMENTAL process to continue. Recovery of the catalog will fail and the APPLYINCREMENTAL process will not continue for either or both of the following conditions:
 - Any catalog was backed up by DFSMSHsm operating with a version of MVS/DFP earlier than version 2 release 3.0
 - Any catalog was dumped before being exported or backed up by DFSMSHsm, which uses export
2. DFSMSHsm creates a list of candidate data sets from either or both the dump and backup VTOC copy data sets to control the volume recovery process.
3. DFSMSHsm applies incremental data set backups from the list of candidate data sets with the following exceptions:
 - Catalogs.
 - Data sets cataloged as multiple-volume, including multiple-stripe data sets. If the target volume is SMS-managed, the data set is not currently cataloged on the volume being processed, and the data set has been restored to the volume, DFSMSHsm scratches it from the restored volume.
 - VSAM data sets currently cataloged as MIGRAT. Cluster, index, and data components are scratched from the restored volume.
 - Non-VSAM data sets currently cataloged as MIGRAT but the dump VTOC copy data set is more recent than the backup VTOC copy data set. If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.
 - VSAM volume data sets (VVDS).
 - VTOC index data sets.
 - Data sets that are not on the backup VTOC copy data set but the backup VTOC copy data set is more recent than the dump VTOC copy data set. If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.

- Currently uncataloged VSAM data sets. If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.
- Currently uncataloged non-VSAM data sets targeted to an SMS-managed volume. If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.
- ICF VSAM component data sets.
- Data sets that are currently cataloged as MIGRAT, that were cataloged when backed up, and that are selected for recovery. If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.
- Data sets that are currently cataloged as MIGRAT, that were uncataloged when backed up, that are selected for recovery, and that were migrated after the backup version of the uncataloged data set was made from the same volume. If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.
- Data sets that are currently cataloged on another volume but the selected data set was cataloged when backed up. If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.
- Data sets that were last backed up when they were cataloged, but a backup version of a data set that was uncataloged when it was backed up is selected for recovery. If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.
- Data sets that are cataloged on the non-SMS-managed target volume but the backup version was made when the data set was uncataloged. (If the data set was uncataloged when backed up from the target volume and is now cataloged on the target volume, and the volume is now SMS-managed, recovery is allowed.) If the data set was restored to the volume, DFSMSHsm scratches it from the restored volume.
- Data sets that are not SMS-managed being recovered to SMS-managed volumes.

Note:

1. If the data set was not on the volume when it was dumped, but was on the volume later in the volume backup (backup VTOC is the latest), the latest backup version is applied regardless of the version date.
2. When you are running full-volume restore with update, there is one instance when DFSMSHsm allows a data set backup version (made while the data set was uncataloged) to be recovered for a data set that is currently cataloged. This occurs when the volume being recovered is SMS-managed, and the backup version of a data set was made from this volume when the data set was uncataloged—the volume was converted to SMS-managed—and the data set is currently cataloged on the volume being recovered.

If a data set is cataloged as a multiple-volume or multiple-striped extended format data set, incremental recovery of the data set fails.

3. If a dump VTOC copy data set does not exist for the volume being restored, some of the cleanup described in the preceding steps may not be performed. The extent of the cleanup activity depends on the candidate data set list that is created from the current backup and dump VTOC copies. If you do not have a dump VTOC copy that corresponds to the dump copy being restored, you should verify that the APPLYINCREMENTAL parameter causes the volume and the catalogs to be synchronized.

It is not necessary for the volume to have been processed by the DFSMSHsm volume backup function if a dump VTOC copy data set exists for the dump generation from which the volume is restored. The VTOC copy data sets are used only to build a candidate list. Rely on the current contents of the catalogs and BCDS to verify any action to be taken during the apply incremental process.

4. During volume restore with update, all non-VSAM SMS data sets are recovered to the volume being restored. VSAM SMS data sets may be recovered to other volumes that are in the same storage group as the volume being restored.

The APPLYINCREMENTAL subparameter has some restrictions: in systems running DFSMSdftp, DFSMSHsm can recover catalogs with the APPLYINCREMENTAL process. To be recoverable, these catalogs must have been backed up with MVS/DFP Version 2 Release 3.0, or later. To recover these catalogs, do the following:

- Start DFSMSHsm under DFSMSdftp
- Use DFSMSHsm to back up all catalogs

- Back up new catalogs before dumping

When the preceding steps are taken, DFSMSHsm can recover the integrated catalog facility catalogs during the APPLYINCREMENTAL process. If any of these steps are omitted, the APPLYINCREMENTAL process stops. For this condition, you may need to update aliases manually or with the recommended integrated catalog Forward Recovery Utility (ICFRU), and restart the incremental recovery process.

If you plan to use ICFRU regularly, you should run it before starting the incremental recovery process. Use the following procedure:

1. Use the DFSMSHsm RECOVER command to recover the catalogs that resided on the volume being restored.
2. Run the ICFRU on each catalog. See the instructions for the ICFRU for details.
3. Use the DFSMSHsm BACKDS command to back up the updated catalogs. This updates the backup versions used during the incremental recovery process.
4. Issue the RECOVER FROMDUMP command with the APPLYINCREMENTAL parameter.

If you do not plan to use ICFRU on the system, you must audit the volumes after the recovery process to either delete data sets or delete catalog entries in order to clean up any discrepancies found. However, if ICFRU is used before the incremental recovery process, no discrepancies occur in the catalogs.

Note:

1. If the data set has been migrated, scratched, or moved since the latest backup version was made, it must be scratched after the recovery process has completed. Because the data set has migrated, was scratched, or was moved, it is not the most current copy, and the catalog has no way to point to it.
2. If the data set was recalled since the backup of the catalog, the catalog entry must be removed by using the DELETE NOSCRATCH command. You can attempt to recover the data set from the latest backup version.
3. If a data set was created since the backup of the catalog, the catalog entry must be removed by using the DELETE NOSCRATCH command.

Full-volume restore from a dump copy

Full-volume restore is the process of recovering the entire physical contents of a volume. Full-volume restore, an alternative to the DFSMSHsm volume recover function, invokes Data Facility Data Set Services (DFSMSDss) to perform full-volume restores. DFSMSHsm uses the dump volumes of a dump copy as input to a DFSMSDss full-volume restore request. If the dump copy starts in file 2 or beyond on a stacked dump tape, DFSMSHsm supplies the file block ID for high-speed positioning to the start of the dump copy. The following commands can be used to request a full-volume restore:

```
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP  
  
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP DATE(date)  
  
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP (DUMPVOLUME(tape_volser))  
  
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP (DUMPCCLASS(class))  
  
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP (DUMPCCLASS(class)) DATE(yyyy/mm/dd) -  
  
RECOVER * TOVOLUME(original_volser) UNIT(unittype) -  
FROMDUMP (DUMPGENERATION(dgennum)) -  
TARGETVOLUME(spare_volser)
```

Note:

1. The TARGETVOLUME(*spare_volser*) parameter can be used to supply the restore with a volume that has a volume serial number different from the one that is being restored. This saves you the step of changing the volume serial number (clipping a volume) to the lost volser.
2. A DFSMSdss restriction is that the original source volume of the full-volume dump must be of the same or similar device type (for example, 3390) as the target volume of the full-volume restore.
3. For more information, see [“TARGETVOLUME: Specifying the volume to receive the restored data sets” on page 1009 and Chapter 60, “RECOVER command: Recovering a backup version or dump copy of a data set, or UNIX file, or a volume,” on page 997.](#)
4. The VTOC and the volume serial on the target volume may have changed as a result of the command. Before the volume can be accessed on any remote system, the UCB must be refreshed. The refresh occurs automatically if the volume is online and the device manager REFUCB function is enabled. You enable the REFUCB function through PARMLIB member DEVSUPxx or the MODIFY DEVMAN command. For more information, refer to the description of the REFUCB keyword in [z/OS MVS Initialization and Tuning Reference](#) or [z/OS MVS System Commands](#).

The first form of the command restores the volume from the most recent dump copy. However, DFSMSShsm must search its records to determine the most recent dump copy for the particular volume. You should not use this form of the command because of the extensive searching required.

The second form of the command narrows the search for the correct dump copy. If you know the date that the volume was dumped to the copy that you want to restore from (either the most recent or some other copy), you can specify the date. Then DFSMSShsm searches only for dump copies made on or before the date you have specified.

Similarly, the other forms of the command allow you to narrow the search by specifying:

- The volumes to which the source volume was dumped
- The dump class from which the dump volumes are to be taken
- The dump generation from which the dump volumes are to be taken

When a full-volume restore is requested, DFSMSShsm verifies with the tape operator that the dump volumes are available for mounting. If any dump volume is unavailable, the restore fails. Dump volumes can be stored at a location other than the data processing location.

If the volume being restored is an SMS-managed volume and the dump copy being used to restore the volume was made when the volume was *not* SMS-managed, DFSMSShsm fails the command and issues an error message. Likewise, if an attempt is made to restore a non-SMS-managed volume from a dump copy of the volume when it was SMS-managed, DFSMSShsm fails the command and issues an error message.

The dump volumes on which the dump copy resides must have a retention period other than NOLIMIT unless the DUMPCLASS or DUMPVOLUME parameter is specified. DFSMSShsm assumes the dump volumes are physically available to be mounted when the RECOVER command is issued. DFSMSShsm considers any volume with a retention period of NOLIMIT to be physically not available.

To restore an SMS-managed volume, you must have DFSMSdftp installed at the correct level to support SMS, and SMS must be active.

DFSMSdss restores a volume only if the volume is not allocated to another job. If the volume contains a catalog, the catalog address space in a DFSMSdftp environment may have the catalog (and the volume) allocated. To free the volume before starting the restore, do the following:

1. Issue the F CATALOG,OPEN(*volser*) command to determine which catalogs on the target volume are allocated.
2. For each catalog that is open, issue the F CATALOG,UNALLOCATE (*catalogname*) to cause the catalog address space to close the catalog and deallocate it. When the last catalog on the volume is deallocated, the target volume is also deallocated by the catalog address space.
3. Issue a D U,,,addr,1 command to ensure that the volume is not allocated by another user.
4. If the volume is not now marked as allocated, issue the RECOVER command to DFSMSShsm.

Related reading

For more information about a DFSMSdss full-volume restore, see [z/OS DFSMSdss Storage Administration](#).

Volume recovery from incremental backup versions

Normally volume recovery is run when an entire DASD pack has been lost or damaged or a significant amount of the data has become inaccessible. A copy of the VTOC made during the incremental backup of the volume is used to drive the individual incremental recovery of data sets for which DFSMSShsm has backup versions. DFSMSShsm volume recovery is also referred to as incremental volume recovery. After DFSMSShsm successfully recovers a volume, each supported data set on the recovered volume is at the level of its latest backup. This recovers the data set to the most recent level unless someone changed the data set since the last time DFSMSShsm backed it up.

The following commands cause volume recovery from incremental backup copies:

```
RECOVER * TOVOLUME(volser) UNIT(unitttype)
RECOVER * TOVOLUME(volser) UNIT(unitttype) DATE(date)
```

Volume recovery is similar to data set recovery in that DFSMSShsm recovers each data set; DFSMSShsm does not recover data track by track. First, a single task builds the queue of data sets that need to be recovered, then multitasking is used during the actual recovery of the data sets. If recovering multiple volumes, it is most efficient to first put a hold on tape data set recovery, then create the queue of data sets to be recovered, then release the tape data set recovery processing. Since data sets from different volumes may reside on one tape, recovering data sets for multiple volumes at the same time, rather than volume-by-volume, reduces the number of required tape mounts, and thus speeds processing. However, because DFSMSShsm recovers each data set and because the backup operation uses backup volumes from different days, volume recovery can require access to many backup volumes just to recover one level 0 volume. As a result, volume recovery can be very time-consuming. Consider using the DFSMSShsm full-volume dump restore facility (see [“Full-volume restore with update” on page 295](#)) to reduce the time required for recovery.

You can specify that backup versions be at least as recent as a particular date by specifying the DATE parameter of the RECOVER command. You must use the TOVOLUME parameter to identify an entire volume that you want to recover.

DFSMSShsm uses the *latest* VTOC copy data set to recover the volume unless an error occurs in allocating or opening the latest VTOC copy data set. If the error occurs, DFSMSShsm attempts to recover the volume, using the *next latest* copy of the VTOC copy data set. If a similar error occurs again, the volume recovery fails.

When DFSMSShsm recovers a volume, it recovers the following:

1. Any base entries for a generation data group cataloged in the catalog residing on the volume being recovered.
2. Any integrated catalog facility user catalogs that do not already exist on the volume being recovered. For information about recovering an integrated catalog facility catalog, see [“Backing up and recovering a catalog” on page 487](#).
3. All other data sets except the following:
 - Catalogs
 - VSAM data sets currently cataloged as MIGRAT
 - ICF VSAM components
 - Data sets that are cataloged as multiple-volume
 - Non-VSAM data sets that are currently cataloged as MIGRAT but the selected backup version was made when the data set was cataloged
 - Non-VSAM data sets that are currently cataloged as MIGRAT, the selected backup version was made when the data set was uncataloged, but the migration took place after the backup version was made

- Data sets that are cataloged on a volume other than the one being recovered when the backup version of a cataloged data set is selected for recovery
- Data sets that are currently cataloged on the volume being recovered but the backup version selected for recovery was uncataloged when it was backed up
- Data sets that were cataloged when backed up and but not currently cataloged
- Data sets whose creation date is later than the backup date
- Data sets that are not SMS-managed being recovered to a SMS-managed volume

DFSMSHsm considers data sets uncataloged at the time they are backed up to be unique to the volume on which they reside. This is because multiple uncataloged data sets with the same name may exist on different volumes. Backup versions of uncataloged data sets will only be recovered to the volume from which they were backed up.

DFSMSHsm considers a cataloged data set to be the same data set regardless of the volume on which it is currently cataloged. Therefore, DFSMSHsm will recover the most current backup version of a cataloged data set regardless of the volume on which it was cataloged when it was backed up.

Table 32 on page 301 shows the process of backup version data set selection for volume recovery.

Table 32. Backup Version Data Set Selection for Recovery

First Criterion	Second Criterion	Result
If only backup versions of cataloged data sets exist,	and if the data set is currently cataloged on the target volume,	then the cataloged backup copy is recovered.
	and if the data set is not currently cataloged,	then recovery of this data set fails.
	and if the data set is cataloged but not on the target volume,	then recovery is not attempted.
If only backup versions of uncataloged data sets exist,	and if the data set is currently cataloged on the target volume,	then recovery fails.
	and if the data set is not currently cataloged,	then the uncataloged backup version, made from this volume, is recovered.
	and if the data set is cataloged but not on the target volume,	then the uncataloged backup version, made from this volume, is recovered.
If backup versions of cataloged and uncataloged data sets exist,	and if both backup versions are from the target volume,	then DFSMSHsm recovers the most recent backup copy. Recovery continues, depending on whether the data set was cataloged, as previously discussed.
	and if the backup version of a cataloged data set was made from a different volume, and the data set is currently cataloged on a volume other than the target volume,	then DFSMSHsm attempts to recover the uncataloged backup version.
	and if the backup version of a cataloged data set was made from a different volume, but the data set is currently cataloged on the target volume,	then DFSMSHsm attempts to recover the backup version of the cataloged data set.
	and if the backup version of a cataloged data set was made from a different volume, but the data set is not currently cataloged,	then DFSMSHsm attempts to recover the uncataloged backup version.

For SMS-managed volumes, SMS must be active to recover the data sets that were backed up from that volume.

All non-VSAM SMS data sets are recovered to the volume being recovered. VSAM SMS data sets may be recovered to other volumes that are in the same storage group as the volume being recovered.

If DFSMSHsm encounters a cataloged multiple-volume VSAM data set, it issues a message stating that the data set is multiple volume but does not recover the data set. If the multiple-volume data set is not cataloged or is presently cataloged as a single-volume data set, DFSMSHsm recovers the data set if an eligible backup version exists.

Processing priority of volume recoveries

Programming Interface Information

All volume recover requests are queued to and processed from a queue of management work elements called the volume recovery queue (separate from the recovery queue used for data sets). DFSMSHsm gives all WAIT requests a higher priority on the queue than NOWAIT requests.

You can change the priorities of these requests as they are put on the queue by using the return-priority installation exit ARCRPEXT. ARCRPEXT allows you to assign a relative queuing priority to WAIT requests so they can be prioritized relative to other WAIT requests, and NOWAIT requests prioritized relative to other NOWAIT requests. Refer to [z/OS DFSMS Installation Exits](#) for a description of ARCRPEXT.

End Programming Interface Information

If you are using MVS concatenated PARMLIB support, DFSMSHsm uses the PARMLIB data set containing the ARCCMDxx member and the PARMLIB data set containing the ARCSTRxx member indicated in the startup procedure.

Chapter 7. Availability management of non-SMS-managed storage

DFSMSHsm manages non-SMS-managed storage on a system basis. That is, all the data sets on volumes are managed by a system specification, whether for number of days between backups or number of retained versions. Primary differences in processing for SMS-managed storage and non-SMS-managed storage occur in the following processes:

- Uncataloged data sets can be backed up and recovered.
- Users can specify the volume to which a data set is to be recovered.
- Volumes to be dumped or backed up are individually identified to DFSMSHsm.

Unlike space management, availability management of non-SMS-managed storage has no major differences in the processing techniques from those that are used for management of SMS-managed storage. However, there are some additional tasks that must be accounted for when dealing with the availability management of non-SMS-managed storage.

References to the example system in this topic reflect input needed to support the tasks for non-SMS-managed storage.

Tasks for availability management of non-SMS-managed storage

Because storage groups or management classes are not available to define how to manage the volumes and which volumes to manage for non-SMS-managed storage, the following subtasks provide a way to manage non-SMS-managed storage:

- Defining how frequently data sets that are managed by a system be backed up
- Defining how many versions to keep for each backed up data set
- Defining which volumes are backed up automatically
- Defining which volumes are dumped automatically

Specifying how frequently data sets are backed up

For SMS-managed data sets, you can specify a different backup frequency for each management class. This lets you have as many different backup frequencies as you have management classes. For non-SMS-managed storage, you can specify only one backup frequency for all volumes processed by any one DFSMSHsm host. You can, however, change the backup frequency of individual non-SMS-managed data sets with the (H)ALTERDS command. See “Changing the backup conditions for a data set” on page 308 for more information. Both the backup frequency specified for a DFSMSHsm host and the backup frequency specified for a management class indicate the number of days that must pass between backups of any data set.

The FREQUENCY parameter of the SETSYS command controls the backup frequency for the data sets processed by each DFSMSHsm host. For the example, the command that is added to the ARCCMDxx member for each host is:

```
SETSYS FREQUENCY(1)
```

Specifying how many versions to keep for each backed up data set

For non-SMS-managed storage, the number of backup versions to retain is a DFSMSHsm host-wide specification.

Note: You can use the (H)ALTERDS command to specify a different value from the DFSMSHsm host-wide default for individual data sets.

As with SMS-managed storage, depending on the record length that is used to define the BCDS, DFSMSHsm can maintain up to 29, or up to 100, backup versions of any data set. Within that upper limit, the VERSIONS parameter of the SETSYS command controls the number of backup versions to be retained. For the example system, the following command is added to the ARCCMDxx member for each DFSMSHsm host:

```
SETSYS VERSIONS(5)
```

If you use the RETAIN DAYS keyword when you create a data set backup copies, and active backup copies with an unmet RETAIN DAYS values are rolled off, DFSMSHsm maintains them as retained backup copies. DFSMSHsm can maintain more than enough retained copies for each data set to meet all expected requirements.

Related reading

- For more information about how record length affects the number of backups, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- [“Specifying the number of days to retain a backup copy of a data set” on page 183.](#)
- [“Creating retained backup copies” on page 184.](#)

Specifying which volumes are backed up automatically

For non-SMS-managed storage, you can identify each individual volume that DFSMSHsm backs up automatically. At the same time that you make the volume known to DFSMSHsm, you can specify the kind of unit to back the volume up to.

If multiple DFSMSHsm hosts are requested to process automatic backup, only the first DFSMSHsm host to process a given volume will perform the function. However, should you find it necessary to limit processing of certain volumes to one DFSMSHsm host, you can do so by:

- Specifying the AUTOBACKUP parameter on the ADDVOL command in the DFSMSHsm host you want to perform automatic backup on the volume
- Specifying the NOAUTOBACKUP parameter on the ADDVOL command in the DFSMSHsm hosts you do not want to perform automatic backup on the volume

See [Figure 3 on page 18](#) for the example system to which you will add the volumes. DFSMSHsm host 3 will do the backup processing for the non-SMS-managed volumes, so the following commands are added to the ARCCMDxx member for DFSMSHsm host 3:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL GP0002 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL GP0003 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG1001 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG1002 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG1003 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG2001 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG2002 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL UG2003 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL VP0001 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL VP0002 UNIT(3390) PRIMARY(AUTOBACKUP)
ADDVOL VP0003 UNIT(3390) PRIMARY(AUTOBACKUP)
```

You can add the following commands to the ARCCMDxx member in DFSMSHsm hosts 1 and 2:


```

ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTOBACKUP)
ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTOBACKUP)

```

For non-SMS-managed volumes, you can restrict the kind of unit that is used when backing up the volume. The BACKUPDEVICECATEGORY parameter of the ADDVOL command lets you specify whether to back the volume up to either DASD or tape (if you have identified both DASD and tape backup volumes to DFSMSHsm with the ADDVOL command).

Specifying which volumes are dumped automatically

As with volumes to be backed up, you must identify the non-SMS-managed volumes to be dumped automatically. With the same command, you specify the dump classes to which the volume is to be dumped. If multiple DFSMSHsm hosts are requested to process automatic dump, only the first host to process a given volume performs the function. However, should you find it necessary to limit processing of certain volumes to one host, you can do so by:

- Specifying the AUTODUMP parameter on the ADDVOL command in the host you want to perform automatic dump on the volume
- Specifying the NOAUTODUMP parameter on the ADDVOL command in the hosts you do not want to perform automatic dump on the volume

See [Figure 3 on page 18](#) for the example system to which the volumes are added. You can add the following commands to the ARCCMDxx member for DFSMSHsm host 2:


```
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL GP0002 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL GP0003 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL UG1001 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL UG1002 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL UG1003 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL UG2001 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL UG2002 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL UG2003 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,TWOWEEK))  
ADDVOL VP0001 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,FOURWEEK))  
ADDVOL VP0002 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,FOURWEEK))  
ADDVOL VP0003 UNIT(3390) PRIMARY(AUTODUMP -  
  (ONEWEEK,FOURWEEK))  
ADDVOL ML1001 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 -  
  AUTODUMP(ONEWEEK,TWOWEEK))  
ADDVOL ML1002 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 -  
  AUTODUMP(ONEWEEK,TWOWEEK))  
ADDVOL ML1003 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 -  
  AUTODUMP(ONEWEEK,TWOWEEK))
```

You can add the following commands to the ARCCMDxx member for DFSMSHsm hosts 1 and 3:

```
ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTODUMP)  
ADDVOL ML1001 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 NOAUTODUMP)  
ADDVOL ML1002 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 NOAUTODUMP)  
ADDVOL ML1003 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 NOAUTODUMP)
```

All volumes except VP0001, VP0002, and VP0003 are dumped in accordance with the specifications for dump classes ONEWEEK and TWOWEEK. VP0001, VP0002, and VP0003 are dumped in accordance with the specifications for dump classes ONEWEEK and FOURWEEK. Notice that the ML1 volumes ML1001, ML1002, and ML1003 have been added to the list of volumes to be dumped. Although automatic backup cannot be performed for migration volumes, automatic dump can be performed for ML1 volumes.

Automatic availability management of non-SMS-managed volumes

Processing for availability management of non-SMS-managed volumes is the same as that for SMS-managed volumes with the following exceptions:

- No request is sent to DFSMSdftp for the management classes and storage classes.
- The FREQUENCY and VERSION parameters of the SETSYS command control how often data sets are backed up and how many versions are kept for all of the non-SMS-managed data sets.
- A non-SMS-managed volume can contain uncataloged data sets that can be backed up.

Backing up cataloged and uncataloged data sets with the same name

Although DFSMSShsm and DFSMSdss do not migrate uncataloged data sets, DFSMSShsm can back them up, either automatically or by command. Therefore, DFSMSShsm can back up several different uncataloged data sets that have the same data set name but reside on different volumes. Depending on the record length that is used to define the BCDS, DFSMSShsm can maintain up to 29, or up to 100, backup versions for any specific data set name. After the number of backup versions reaches that limit, backup versions of cataloged data sets take precedence over backup versions of uncataloged data sets of the same name. When DFSMSShsm backs up the data set again, DFSMSShsm deletes the backup version of an uncataloged data set first. In the case of data sets that are indirectly cataloged to SYSRES, those on the inactive SYSRES pack are considered uncataloged by backup processing, but are considered cataloged if backed up while running backup from their own system. For this reason, they cannot be protected by incremental backup. The results of a recover command are unpredictable due to the mixture of indirectly cataloged and uncataloged data sets with the same name.

To avoid potential confusion, do not give cataloged and uncataloged data sets the same data set name.

DFSMSShsm keeps backup versions of a data set name in the following order:

1. All backup versions of a cataloged data set, arranged by date. The most recently created backup version is first. The number of backup versions DFSMSShsm keeps is the limit of versions that existed the last time DFSMSShsm backed up the data set.
2. All backup versions of all uncataloged data sets having the same name, arranged by date. The most recently created backup version is first. The number of backup versions DFSMSShsm keeps is the limit of versions that existed the last time DFSMSShsm backed up the data set.

Although the limit specified by the VERSIONS parameter of the SETSYS command applies to cataloged and uncataloged data sets separately, the maximum number of versions (based on the BCDS record size) applies to the sum of the backup versions of both cataloged and uncataloged data sets with the same name. When the sum of the current backup versions is greater than the BCDS-related limit, DFSMSShsm creates a new version and deletes the oldest backup version of the uncataloged data sets. For example, assume the following:

- The maximum BCDS record size allows DFSMSShsm to maintain 29 backup versions for any data set.
- A cataloged data set and several uncataloged data sets have the same data set name.
- The cataloged data set currently has 24 backup versions.
- The uncataloged data sets currently have four backup versions.
- You specified a limit of 24 with the VERSIONS parameter.
- DFSMSShsm backs up one cataloged data set and two uncataloged data sets with the same name.

Because DFSMSShsm (in this example) tracks only 29 backup versions, here are the results:

- Twenty-four backup versions of the cataloged data set.
- Five backup versions of the uncataloged data sets. DFSMSShsm keeps the five most recently created backup versions of these data sets.

You can identify backup versions of an uncataloged data set by associating the data set name with the volume serial number of the source volume for the backup version. DFSMSShsm considers this association of the data set name and the source volume serial number a unique identification.

Command availability management for non-SMS-managed storage

The activities that were specified as being available by command for SMS-managed storage are available for non-SMS-managed storage. Differences for non-SMS-managed storage are:

- A command that allows changing the backup conditions for individual data sets
- The ability to back up uncataloged data sets
- The ability to back up all non-SMS-managed primary volumes
- The ability to delete backup versions of uncataloged data sets
- The ability to specify a frequency value for a volume backed up by command
- The ability to recover uncataloged data sets
- The ability to recover an SMS-managed data set to a non-SMS-managed volume
- Differences in the way that the EXPIREBV command selects backup versions for deletion

This topic describes the differences between the commands for SMS-managed storage and for non-SMS-managed storage. The forms of the commands specified for SMS-managed storage also are used for non-SMS-managed storage. However, when they are used for non-SMS-managed storage, those forms of the commands use the frequency and versions attribute specified with the system SETSYS command.

Dumping a volume

The commands for dumping a volume are the same for non-SMS-managed storage as for SMS-managed storage. However, the dump process for non-SMS-managed volumes does not request the storage group attributes from DFSMSdfp.

Changing the backup conditions for a data set

SMS-managed storage controls the frequency of backup and the number of versions retained for each data set or UNIX file through the management class attributes. For non-SMS-managed storage or UNIX files contained in a non-SMS-managed zFS, frequency and versions are a DFSMSShsm host-wide specification. However, you can control the frequency and versions for individual data sets with the (H)ALTERDS command. You can control the frequency and the versions independently of each other. The forms for the command are:

```
ALTERDS dsname ... FREQUENCY(days)
ALTERDS dsname ... VERSIONS(limit)
ALTERDS 'pathname' ... VERSIONS(limit)

ALTERDS dsname ... FREQUENCY(days) VERSIONS(limit)
HALTERDS dsname ... FREQUENCY(days)
HALTERDS dsname ... VERSIONS(limit)
HALTERDS 'pathname' ... VERSIONS(limit)
HALTERDS dsname ... FREQUENCY(days) VERSIONS(limit)
```

Non-DFSMSShsm-authorized and DFSMSShsm-authorized users can use the HALTERDS form of the command. Only DFSMSShsm-authorized users can use the ALTERDS form of the command.

The command causes the frequency or versions attributes for the data sets that are identified by *dsname* or UNIX files that are identified by *pathname*, to be changed to the values that are specified by the command parameters. The new values are then used for all volume backup functions that are performed on those data sets until a new ALTERDS or HALTERDS command is issued for the data set. For UNIX files, the new VERSIONS value is used for EXPIREBV, and determining roll-off of old backups.

Security checking with discrete profiles

If your data sets are protected with discrete profiles, HALTERDS processing checks the profile of the original data set for authorization to change the backup conditions for the data set. If the original data set has been deleted, DFSMSHsm checks any backed up discrete profile for the data set. If the second authorization check fails, the HALTERDS processing fails.

Backing up an individual data set

Backing up an individual cataloged data set is performed in the same way as for SMS-managed data sets. However, to back up individual *uncataloged* data sets, issue the following command:

```
BACKDS dsname UNIT(unitttype) VOLUME(volser)
HBACKDS dsname UNIT(unitttype) VOLUME(volser)
```

The HBACKDS form of the command can be used by either non-DFSMSHsm-authorized or DFSMSHsm-authorized users. The BACKDS form of the command can be used only by DFSMSHsm-authorized users. The UNIT and VOLUME parameters are required because DFSMSHsm cannot locate an uncataloged data set without being told where it is.

Inline backup of an individual data set

Backing up individual cataloged data sets is performed the same way as for SMS-managed data sets. (See [“Backing up an individual data set or UNIX file” on page 222.](#)) However, the backup of uncataloged data sets is not supported by inline backup.

Related reading

For more information about how to use inline backup, see [“Inline backup” on page 447.](#)

Backing up all primary volumes

Use the following command to back up all non-SMS-managed primary volumes:

```
BACKVOL PRIMARY
```

This command causes DFSMSHsm to back up *every* volume that has been identified to it as a primary volume by the ADDVOL command, not just those volumes with the AUTOBACKUP attribute.

Note: Volumes selected for processing by BACKVOL PRIMARY are not processed by automatic backup the same day and visa versa.

Backing up all changed data sets on volumes

You can back up all eligible changed data sets on non-SMS-managed volumes, whether the volumes are managed by DFSMSHsm or not. Issue the following additional commands to back up non-SMS-managed volumes:

```
BACKVOL VOLUMES(volser,volser,...volser)
BACKVOL VOLUMES(volser,volser,...volser) FREQUENCY(days)
BACKVOL VOLUMES(volser,volser,...volser) UNIT(unitttype) -
    FREQUENCY(days)
BACKVOL PRIMARY INCREMENTAL
BACKVOL PRIMARY INCREMENTAL FREQUENCY(days)
```


The first two forms of the command are used for backing up volumes that are managed by DFSMSHsm. The first form backs up the changed data sets on the volume in accordance with the frequency for the system as specified by the SETSYS command. The next form uses the frequency specified on the command to replace the attribute specified for the system.

The third form of the command backs up a volume that is not managed by DFSMSHsm.

The last two forms of the command use either the frequency attribute specified for the system or the one specified in the command to back up all the non-SMS-managed primary volumes in the system.

Deleting expired backup versions of non-SMS-managed data sets

To delete expired backup versions from the backup volumes, use the EXPIREBV command as you do for deleting expired SMS-managed backup versions. Issue the following command to delete non-SMS-managed backup versions:

```
EXPIREBV NONSMSVERSIONS(DELETEIFBACKEDUP(days))  
EXECUTE
```

```
EXPIREBV NONSMSVERSIONS(CATALOGEDDATA(days))  
EXECUTE
```

```
EXPIREBV NONSMSVERSIONS(UNCATALOGEDDATA(days))  
EXECUTE
```

Processing for expired backup versions of non-SMS-managed data sets

The EXPIREBV command processing for non-SMS-managed data sets is identical to that for SMS-managed data sets up to the point of determining the expired backup versions. EXPIREBV processing for SMS-managed data sets is described in [“Deleting expired backup versions from the backup volumes” on page 233](#).

The EXPIREBV command has an initial process that checks the retention days value for each active backup version for each data set. If you specified a retention days value when the BACKDS command was issued during data set backup, the number of days since the date when the backup version was created is calculated and compared to the retention days value. If the number of days since the backup version was created is greater than the number of retention days, the backup version is deleted. This process includes both the active backup versions and the retained backup versions. If the retention duration has not been exceeded, the backup version is not processed, even if it meets other expiration criteria. If you did not specify a retention days value, the DELETEDIFBACKUP, CATALOGEDDATA, and UNCATALOGEDDATE parameters provided with the NONSMSVERSION parameter of the EXPIREBV command are used to determine which backup versions are deleted.

Note: If there is only one backup version of a data set, EXPIREBV does not delete the backup version. You must use the BDELETE or HBDELETE command to delete the backup version.

If the command includes (DELETEIFBACKEDUP(*days*)), DFSMSHsm deletes any retired version older than the value specified by the *days* subparameter. It also deletes any additional older cataloged backup versions that exist.

If the command includes (CATALOGEDDATA(*days*)), DFSMSHsm processes only data sets that were cataloged when they were backed up. For deleted data sets, if a date of scratch exists in the MCB record, DFSMSHsm calculates the number of days since the date of scratch, and compares the result to the *days* specified in the command, or to the default of 60 days (if *days* was not specified). If the time since the date of scratch is more than the value specified or more than the default, DFSMSHsm deletes all cataloged backup versions of the data set except for the retired version. For existing data sets, the *days* subparameter is ignored, and all excess backup versions (that result from the (H)ALTERDS command) are deleted.

When (CATALOGEDDATA(*days*)) is used, two executions are required to delete unwanted versions. The first execution determines that the data set was scratched and stores the date of the first EXPIREBV

execution as the scratch date in the DFSMSHsm control data set MCB record. A subsequent execution deletes the unwanted backup versions when the requested time has passed. The scratch date is generally not the date that the data set was actually scratched, but is the date EXPIREBV was next run after the data set was scratched.

Programming Interface Information

The scratch date is stored in the BCDS MCB record in the 4-byte field MCBSCRD (offset X'48', packed decimal). It can be displayed using the FIXCDS command.

End Programming Interface Information

Note: DFSMSHsm honors a scratch date stored by the DFSMSHsm space maintenance and recovery utilities if the date is newer than the most recent backup date for the data set and if EXPIREBV has not already stored the scratch date itself.

If the command includes (UNCATALOGEDDATA(*days*)), DFSMSHsm processes data sets that were uncataloged when they were backed up. DFSMSHsm calculates the number of days since the uncataloged backup version was created and compares the result to the days specified in the command. There is no default value for *days* for uncataloged backup versions. If the time since the uncataloged backup version was created is more than the value specified, DFSMSHsm deletes that version.

Deleting backup versions of specific data sets

Because non-SMS-managed data sets can be either cataloged or uncataloged and because you can have both cataloged and uncataloged backup versions with the same data set name, DFSMSHsm provides the ability to specify that you want to delete the backup versions of the uncataloged data set. Issue the following commands to delete backup versions of uncataloged data sets:

```
BDELETE (dsname ...) ALL
FROMVOLUME(volser)

BDELETE (dsname ...)
VERSIONS(backup version number ...) -
FROMVOLUME(volser)

HBDELETE (dsname ...) ALL
FROMVOLUME(volser)

HBDELETE (dsname ...) -
VERSIONS(backup version number ...) -
FROMVOLUME(volser)
```

DFSMSHsm deletes the uncataloged backup versions that were made from the data sets with the specified name on the specified volume.

The MCB record is not deleted when no existing backup versions exist if the data set is not SMS-managed and if the MCB record contains any ALTERDS information.

Recovering specific data sets

DFSMSHsm recovers specific data sets to non-SMS-managed volumes under the following conditions:

- When the FORCENONSMS parameter is specified on the RECOVER command
- When SMS is not active and the backup version or dump copy being recovered was made from a non-SMS-managed data set
- When SMS is active and ACS does not assign a storage class to the data set (indicating that the data set is not currently SMS-managed)
- When the FROMVOLUME parameter is specified on the RECOVER command
- When the catalog indicates that the data set is not SMS-managed

You can issue the following additional commands to recover a non-SMS-managed data set:

- RECOVER *dsname* FROMVOLUME(*volser*)

This command recovers an uncataloged data set to the volume from which it was backed up.

- RECOVER *dsname* FROMVOLUME(*volser*) REPLACE

This command recovers an uncataloged data set to the volume from which it was backed up, and any existing data set of the same name on the volume is replaced.

- RECOVER *dsname* TOVOLUME(*volser*) - UNIT(*unittype*)

This command recovers a data set to the volume specified.

- RECOVER *dsname* TOVOLUME(*volser*) - UNIT(*unittype*) FORCENONSMS

This command recovers what may have been an SMS-managed data set to the specified non-SMS-managed volume.

- HRECOVER *dsname* TOVOLUME(*volser*) - UNIT(*unittype*)

This is the unauthorized-user form of the command. It recovers a data set to the volume specified.

- RECOVER *dsname* NEWNAME(*newdsname*) - TOVOLUME(*volser*) UNIT(*unittype*)

This command recovers a data set to the volume specified and gives the data set a new name.

- RECOVER *dsname* NEWNAME(*newdsname*) - TOVOLUME(*volser*) UNIT(*unittype*) - FORCENONSMS

This command recovers what may have been an SMS-managed data set to the specified non-SMS-managed volume and gives the data set a new name.

- HRECOVER *dsname* NEWNAME(*newdsname*) - TOVOLUME(*volser*) UNIT(*unittype*)

This command is the unauthorized user form of the command. The data set is recovered to the volume specified and is given a new name.

- RECOVER *dsname* REPLACE

This command recovers a cataloged data set to the volume where the data set currently resides and replaces the existing data set with the recovered data set.

- RECOVER *dsname* REPLACE FORCENONSMS

This command recovers a cataloged data set that was backed up as an SMS-managed data set to the non-SMS-managed volume where the existing data set currently resides. The existing data set is replaced with the recovered data set.

- RECOVER *dsname* REPLACE TOVOLUME(*volser*) - UNIT(*unittype*)

This command recovers a data set to the specified volume and replaces any existing data set with the same name on the specified volume.

- RECOVER *dsname* REPLACE TOVOLUME(*volser*) - UNIT(*unittype*) FORCENONSMS

This command recovers a data set that was backed up as an SMS-managed data set to the specified non-SMS-managed volume. Any existing data set with the same name on the specified volume is replaced.

- HRECOVER *dsname* REPLACE TOVOLUME(*volser*) - UNIT(*unittype*)

This is the unauthorized-user form of the command. It recovers a data set to a specified volume and replaces any existing data set with the same name on the specified volume.

- RECOVER *dsname* FROMVOLUME(*volser*) - NEWNAME(*newdsname*)

This command recovers an uncataloged data set to the volume from which it was backed up and gives the recovered data set a new name.

- RECOVER *dsname* FROMVOLUME(*volser*) - REPLACE TOVOLUME(*volser*) UNIT(*unittype*)

This command recovers an uncataloged data set to the volume specified. The data set with the same name on the specified volume is replaced.

- RECOVER *dsname* NEWNAME(*newdsname*) - REPLACE TOVOLUME(*volser*) UNIT(*unittype*)

This command recovers a data set to the specified volume. The data set is given a new name, and any existing data set on the specified volume that has the same name as the new name is replaced.

- RECOVER *dsname* NEWNAME(*newdsname*) - REPLACE TOVOLUME(*volser*) - UNIT(*unittype*)
FORCENONSMS

This recovers a data set that was backed up as an SMS-managed data set to the specified non-SMS-managed volume. The data set is given a new name, and any existing data set on the specified volume that has the same name as the new name is replaced.

- HRECOVER *dsname* NEWNAME(*newdsname*) - REPLACE TOVOLUME(*volser*) UNIT(*unittype*)

This is the unauthorized-user form of the command. It recovers a data set to the specified volume. The data set is given a new name, and replaces any existing data set on the specified volume that has the same name as the new name.

- RECOVER *dsname* FROMVOLUME(*volser*) - NEWNAME(*newdsname*) TOVOLUME(*volser*) - UNIT(*unittype*)

This recovers an uncataloged data set to a specified volume and gives it a new name.

- RECOVER *dsname* FROMVOLUME(*volser*) - NEWNAME(*newdsname*) TOVOLUME(*volser*) - UNIT(*unittype*)
REPLACE

This recovers an uncataloged data set to a specified volume. The data set is given a new name, and any existing data set that has the same name as the new name on the specified volume is replaced.

The processing for recovering non-SMS-managed data sets is the same as for SMS-managed data sets with the following exceptions:

- When FROMVOLUME is specified, it implies that the data set is not SMS-managed.
- Selection of the target volume is somewhat different:
 - If TOVOLUME is specified, the target volume is the volume specified by TOVOLUME.
 - If TOVOLUME is not specified and FROMVOLUME is specified, the target volume is the volume specified by FROMVOLUME.
 - If neither TOVOLUME nor FROMVOLUME is specified, the target volume is the volume on which the data set is cataloged, unless DFSMSHsm backed up the data set from a migration volume. If DFSMSHsm backed up the data set from a migration volume, DFSMSHsm recovers the data set to the volume from which the data set migrated.
 - If DUMPVOLUME is specified, the target volume is the source volume from which the dump was made.
 - If the specified NEWNAME data set is not currently cataloged, if the device type and volume serial number of the volume from which the incremental backup version was made differ from the device type and volume serial number of the volume on which the original data set is currently cataloged, and if the volume on which the original data set is currently cataloged has a more recent physical dump copy, the target volume is the volume on which the original non-VSAM data set is currently cataloged.
 - If none of the preceding can be applied, the target volume is the volume from which the eligible backup version was made.

To request that DFSMSHsm recover backup versions of data sets that were uncataloged when DFSMSHsm backed them up, specify the FROMVOLUME parameter of the RECOVER command. If you do not specify FROMVOLUME, DFSMSHsm recovers a backup version of a cataloged data set with the same name even though the data set may no longer be cataloged.

When a generation data set is recovered, it is recovered as an ordinary non-GDS data set.

Note:

1. DFSMSHsm fails the recovery of a non-SMS generation data set that is not currently cataloged, and whose corresponding generation data group is no longer defined.
2. DFSMSHsm fails the recovery of a non-SMS generation data set that is not currently cataloged, and whose corresponding generation data group is not found through generic catalog locate.

To request that an SMS-managed data set be recovered to a non-SMS-managed volume, specify the FORCENONSMS and TOVOLUME parameters.

Recovery of VSAM data sets

The FROMVOLUME and TOVOLUME parameters of the RECOVER command, used when recovering non-VSAM data sets, have different results when DFSMSHsm recovers VSAM data sets:

- The FROMVOLUME parameter applies only to uncataloged data sets and, therefore, is not used for VSAM data sets.
- For VSAM data sets not cataloged, the TOVOLUME parameter can specify only a volume owned by the same catalog where the data set being recovered is cataloged or will be cataloged. When you do not specify the TOVOLUME parameter, DFSMSHsm recovers the data set to the volume specified in the current catalog entry. If no catalog entry exists, DFSMSHsm recovers the data set to the same volume from which DFSMSHsm backed up the data set.

Data set restore from a DFSMSHsm dump copy

The results of the commands, in order, are:

1. RECOVER *dsname* FROMDUMP(DUMPCCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*)

This command recovers a data set from the most recent dump copy made in the specified dump class to the specified level 0 volume.

2. RECOVER *dsname* FROMDUMP(DUMPCCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*)
FORCENONSMS

This command recovers a data set that was SMS-managed when it was dumped. The data set is recovered from the most recent dump copy made in the specified dump class to the specified non-SMS-managed level 0 volume.

3. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*)

This command recovers a data set from the specified dump volume to the specified level 0 volume.

4. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*)
FORCENONSMS

This command recovers a data set that was SMS-managed when it was dumped. The data set is recovered from the specified dump volume to the specified level 0 volume.

5. RECOVER *dsname* FROMDUMP(DUMPCCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) REPLACE

This command recovers a data set from the most recent dump copy made in the specified dump class to the specified level 0 volume. Any existing data set with the same name on the specified volume is replaced.

6. RECOVER *dsname* FROMDUMP(DUMPCCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE
FORCENONSMS

This command recovers a data set that was SMS-managed when it was dumped. The data set is recovered from the most recent dump copy made to the specified dump class in the specified level 0 volume. Any existing data set with the same name on the specified volume is replaced.

7. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) REPLACE

This command recovers a data set from the specified dump volume to the specified level 0 volume. Any existing data set with the same name on the specified volume is replaced.

8. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) -
REPLACE FORCENONSMS

This command recovers a data set that was SMS-managed when it was dumped. The data set is recovered from the specified dump volume to the specified level 0 volume. Any existing data set with the same name on the specified volume is replaced.

9. RECOVER *dsname* FROMDUMP(DUMPCCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) -
NEWNAME(*newname*)

This command recovers a data set from the most recent dump copy made in the specified dump class to the specified level 0 volume. The recovered data set is given a new name.

10. RECOVER *dsname* FROMDUMP(DUMPCCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - NEWNAME(*newname*) FORCENONSMS

This command recovers a data set that may have been SMS-managed when it was dumped. The data set is recovered from the most recent dump copy made in the specified dump class to the specified level 0 volume. The recovered data set is given a new name.

11. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - NEWNAME(*newname*)

This command recovers a data set from the specified dump volume to the specified level 0 volume. The recovered data set is given a new name.

12. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - NEWNAME(*newname*) FORCENONSMS

This command recovers a data set that may have been SMS-managed when it was dumped. The data set is recovered from the specified dump volume to the specified non-SMS-managed level 0 volume. The data set is given a new name.

13. RECOVER *dsname* FROMDUMP(DUMPCCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE NEWNAME(*newname*)

This command recovers a data set from the most recent dump copy that was made in the specified dump class to the specified level 0 volume. The recovered data set is given a new name and replaces any existing data set with the same new name on the specified volume.

14. RECOVER *dsname* FROMDUMP(DUMPCCLASS(*class*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE NEWNAME(*newname*) FORCENONSMS

This command recovers a data set that may have been SMS-managed when it was dumped. The data set is recovered from the most recent dump copy that was made in the specified dump class to the specified non-SMS-managed level 0 volume. The recovered data set is given a new name, and replaces any existing data set with the new name on the specified volume.

15. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE NEWNAME(*newname*)

This command recovers a data set from the specified dump volume to the specified level 0 volume. The recovered data set is given a new name and replaces any existing data set with the new name on the specified volume.

16. RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) - TOVOLUME(*volser*) UNIT(*unittype*) - REPLACE NEWNAME(*newname*) FORCENONSMS

This command recovers a data set that may have been SMS-managed when it was dumped. The data set is recovered from the specified dump volume to the specified non-SMS-managed level 0 volume. The recovered data set is given a new name and replaces any existing data set with the new name on the specified volume.

When restoring a non-SMS-managed data set from a dump volume, you can specify a TOVOLUME parameter and a UNIT parameter that cannot be specified for SMS-managed data sets. The TOVOLUME parameter directs that the data set be restored to the specified volume. The UNIT parameter identifies the unit type for restoring the data set to a volume not managed by DFSMSHsm.

If the dump volume specified has more than one valid dump copy stacked on it, you can specify the SOURCEVOLUME(*volser*) subparameter of the FROMDUMP parameter to tell DFSMSHsm which dump copy to use.

Data set recovery of retired data sets

You can restore a retired data set. If the retired data set has a backup version that is marked as a retired version and a physical data set restore is done for the data set without a new name, the retired-version

indication is removed from the backup control data set (BCDS). This allows the data set to be processed again by the incremental backup function.

Combining full-volume restore with volume recovery

When combining a full-volume restore with a volume-recovery, the following considerations apply to non-SMS-managed volumes:

DFSMSHsm verifies that the targeted data set has not been referred to before starting the APPLYINCREMENTAL process and the time that the data set is slated for recovery. If the data set has been referred to, the recovery fails with a return code of 67. Failing the recovery allows you to take action on any application that might have inadvertently used the old data set.

Restrictions:

1. DFSMSHsm does not honor the APPLYINCREMENTAL subparameter if it determines that a catalog that resides on the target volume owns the target volume. This restriction avoids conflicts in the way DFSMSHsm supports catalogs and data sets that are cataloged.
2. DFSMSHsm does not back up catalogs and therefore they cannot be recovered.

Volume recovery from incremental backup versions

In addition to the considerations discussed for volume recovery from backup versions for SMS-managed volumes, the following considerations apply for non-SMS-managed volumes.

If a catalog exists on the volume being recovered, you must recover the catalog before you use DFSMSHsm for the volume recovery. DFSMSHsm catalogs any data sets that were cataloged when DFSMSHsm backed them up, but that are not cataloged when DFSMSHsm recovers them. If a data set was uncataloged when DFSMSHsm backed it up, DFSMSHsm recovers the data set, but does not catalog it.

If a catalog existed on the volume the last time DFSMSHsm backed up the volume, DFSMSHsm recovers the base entry for each generation data group cataloged. Therefore, if you want DFSMSHsm to recover the base entry for each generation data group cataloged, the catalog must exist.

DFSMSHsm recovers the generation data groups that are on the volume being recovered. If the generation data group base was not on the same volume as the generation data sets, you must ensure that the base entry exists before the volume is recovered. You may have to recatalog valid generation data group data sets that are on other volumes.

DFSMSHsm tries only one time to recover generation data group base entries. The recovery is handled the same way as with VTOC copy data sets. If DFSMSHsm uses the next-latest VTOC copy data set to recover the volume because DFSMSHsm cannot use the latest VTOC copy data set, DFSMSHsm recovers the next-latest copy of the generation data group base entries.

DFSMSHsm attempts to recover the volume by using the *latest* VTOC copy data set and the latest copy of the generation data group base entries. DFSMSHsm attempts to recover the volume by using the *next latest* copy of the VTOC and generation data group base entries if one of the following occurs:

- An error occurs in allocating or opening the latest VTOC copy data set.
- The VTOC copy data set resides on a tape backup volume and the operator indicates before starting the actual recovery that the required tape volumes will not be available.

If similar errors occur again, the volume recovery fails.

When DFSMSHsm recovers a volume, it recovers the following:

- Any base entries for a cataloged generation data group residing on the volume being recovered.
- Any user catalogs that do not already exist on the volume being recovered. For information about recovering a catalog, see [“Backing up and recovering a catalog” on page 487](#).
- Unmovable data sets.
- All other data sets.

DFSMSHsm considers non-SMS uncataloged data sets to be unique by reason of the volume on which they reside. This is because multiple uncataloged data sets with the same name may exist on different volumes. Backup versions of uncataloged data sets are only recovered to the volume from which they were backed up.

DFSMSHsm considers a cataloged data set to be the same data set regardless of the volume on which it is currently cataloged. Therefore, DFSMSHsm recovers the most current backup version of a cataloged data set regardless of the volume on which it was cataloged when it was backed up.

Table 33 on page 317 shows the process of backup version data set selection for volume recovery.

Table 33. Backup Version Data Set Selection for Recovery

First Criterion	Second Criterion	Result
If only backup versions of cataloged data sets exist,	and if the data set is currently cataloged on the target volume,	then the cataloged backup copy is recovered.
	and if the data set is not currently cataloged,	then recovery of this data set fails.
	and if the data set is cataloged but not on the target volume,	then recovery is not attempted, and for APPLYINCREMENTAL, the data set is scratched from the volume.
If only backup versions of uncataloged data sets exist,	and if the data set is currently cataloged on the target volume,	then recovery fails.
	and if the data set is not currently cataloged,	then the uncataloged backup version, made from this volume, is recovered.
	and if the data set is cataloged but not on the target volume,	then the uncataloged backup version, made from this volume, is recovered.
If backup versions of cataloged and uncataloged data sets exist,	and if both backup versions are from the target volume,	then DFSMSHsm recovers the most recent backup copy. Recovery continues, depending on whether the data set was cataloged, as previously discussed.
	and if the backup version of a cataloged data set was made from a different volume, and the data set is currently cataloged on a volume other than the target volume,	then DFSMSHsm attempts to recover the uncataloged backup version.
	and if the backup version of a cataloged data set was made from a different volume, but the data set is currently cataloged on the target volume,	then DFSMSHsm attempts to recover the backup version of the cataloged data set.
	and if the backup version of a cataloged data set was made from a different volume, but the data set is not currently cataloged,	then DFSMSHsm attempts to recover the uncataloged backup version.

If a data set is a candidate for recovery, backup versions of the cataloged data set exist, and the data set is currently uncataloged, then DFSMSHsm does not recover the data set because it appears to DFSMSHsm that the data set was deleted after the backup version was created. For recovery of uncataloged data sets, a backup version of each data set is recovered to reflect the VTOC copy at the time of backup. It is up to you to delete any recovered uncataloged data sets that are unwanted.

If a data set is currently cataloged as MIGRAT, and a backup version of the cataloged data set was selected for recovery, then recovery is not performed because the data set migrated since backup.

If a data set was migrated from the target volume after the selected backup version of the uncataloged data set was created, then the recovery is not attempted because the uncataloged data set was deleted before the creation of the cataloged data set that was migrated.

If the volume you are recovering contains both a catalog and unmovable data sets, you could have a problem. After DFSMSHsm recovers the catalog, DFSMSHsm tries to recover the unmovable data sets. However, the allocation for the unmovable data sets can fail because the catalog could be allocated to the space where the unmovable data set must go. To solve this problem, perform the following steps:

1. After the volume recovery completes without recovering an unmovable data set, move the catalog to another volume.
2. Use the RECOVER command for the specific data sets that previously failed recovery.
3. Move the catalog back to the original volume.

After DFSMSHsm has recovered a volume, you can use the VOLUME parameter of the AUDIT command to verify the status of data sets on the volume.

Chapter 8. Aggregate backup and recovery support (ABARS)

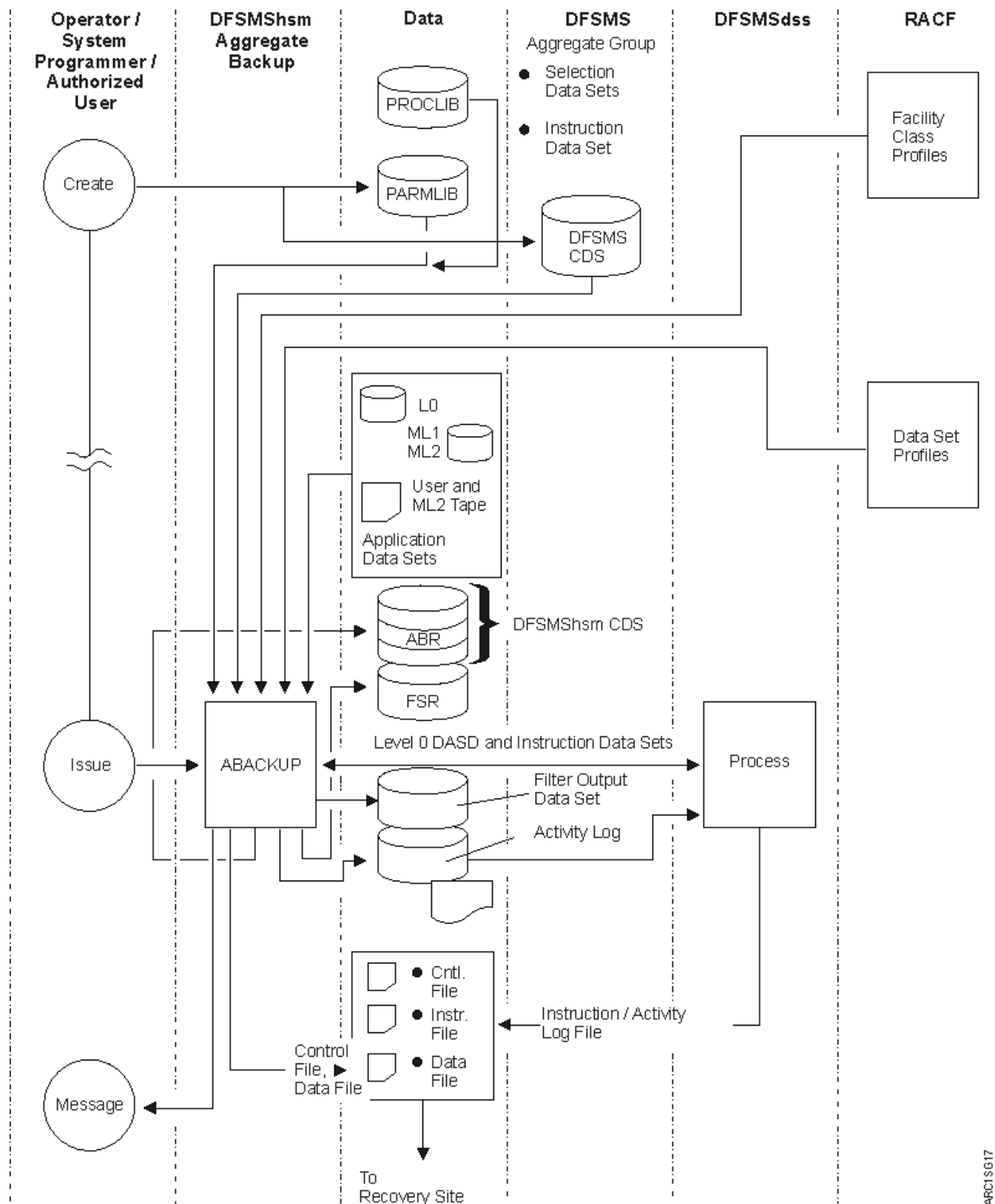
Aggregate backup is the command-driven function that backs up a user-defined group (called an aggregate group) of data sets for recovery at another computer site or at the same site. *Aggregate recovery* is the command-driven function that recovers the data sets that were previously backed up by aggregate backup.

Aggregate backup and recovery support can be used to back up and recover both SMS and non-SMS-managed data.

[Figure 34 on page 320](#) and [Figure 35 on page 321](#) show data movement during aggregate backup and recovery and how these functions interact with other products. The DFSMSHsm aggregate backup and recovery support function is also referred to as ABARS.

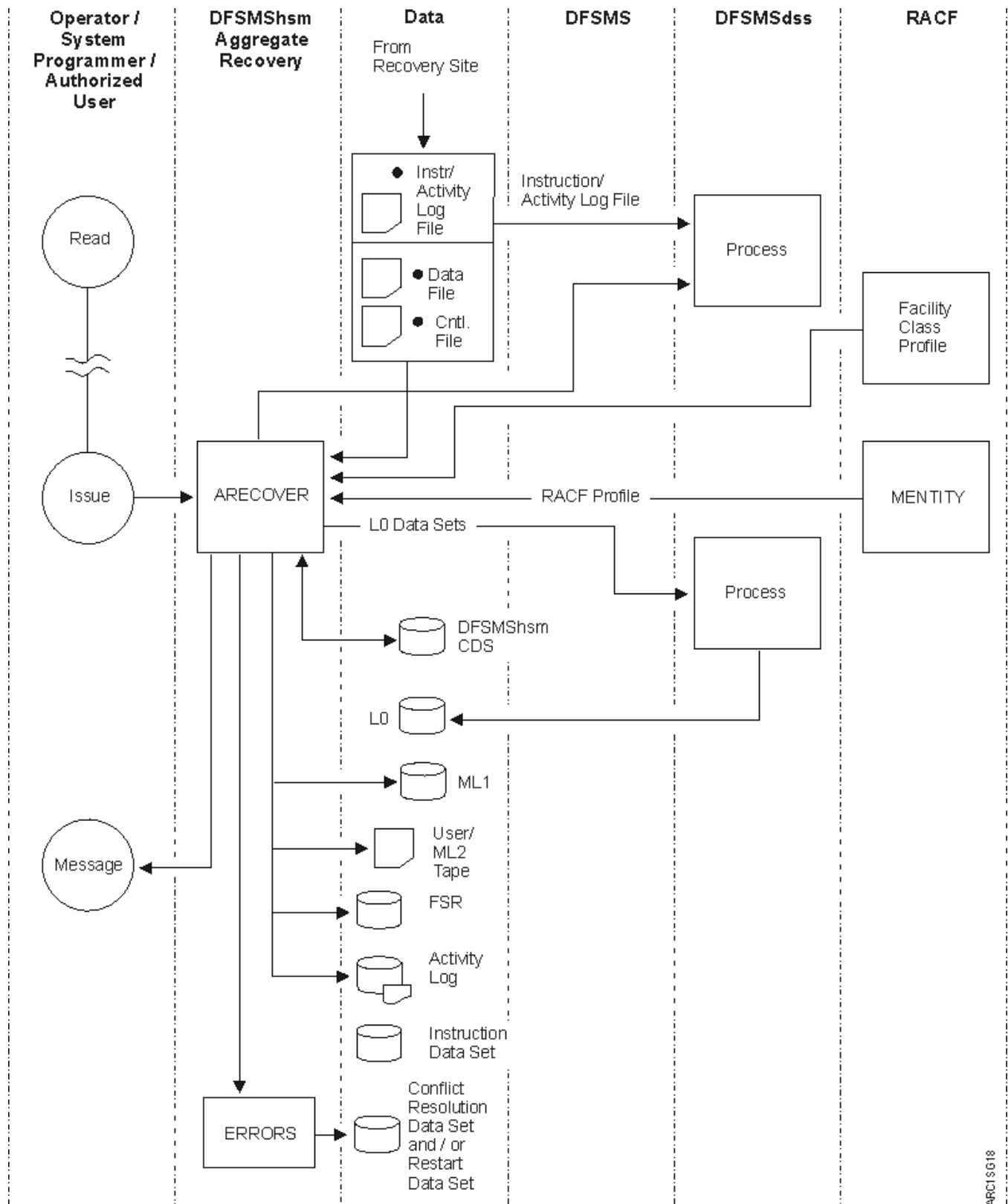
Note: Only a host designated with the HOSTMODE=MAIN attribute can execute aggregate backup or aggregate recovery.

Aggregate backup and recovery support (ABARS)



ARC1SG17

Figure 34. Aggregate Backup and Recovery Support (part 1 of 2)



APR15G18

Figure 35. Aggregate Backup and Recovery Support (part 2 of 2)

The aggregate group and management class used for aggregate backup are defined to DFSMS through Interactive Storage Management Facility (ISMF) panels.

DFSMSdss is used to move level 0 DASD data sets during aggregate backup and recovery processing.

DFSMSHsm can include ML1 and ML2 migration copies of data sets within the aggregate backup. Migration copies that are stored in the cloud (MLC) will not be included and will cause the backup to fail. ABACKUP does not support data sets migrated to the cloud. If the selection criteria results in any of these data sets being selected, then the ABACKUP command fails.

RACF is used to ensure that RACF-protected data sets that are backed up remain RACF-protected when recovered. When RACF facility class is active and the user has restricted command authority, RACF checking of data sets is performed during aggregate backup. See [“Checking authorization during aggregate backup”](#) on page 340 for more information.

Although the backup and recovery sites are referred to as two separate data processing centers, you can choose to perform both functions at the same data processing center. This might be useful if you want to divide your data processing center for work load management.

How aggregate backup works

In general, to use aggregate backup, you first specify the data sets to be backed up, and then use the ABACKUP command or ISMF panels to back up the data sets to tape files. These tape files can then be physically transported or transmitted, using a transmission program like the NetView file transfer program (FTP), to a recovery site where the data sets can be recovered.

The following topics describe how data sets are specified for backup, how the ABACKUP command is used, which tape files are created by aggregate backup, and how to use the Aggregate Backup Assist (ABA) program to produce a selection data set.

Specifying data sets to be backed up

Before you can run aggregate backup, you need to create one or more selection data sets and define an aggregate group and related management class to specify exactly which data sets are to be backed up. You may also decide to create an instruction data set to contain any information you want conveyed to the recovery site. One instruction data set name and up to five selection data set names can be defined within one aggregate group.

Aggregate group

The aggregate group lists the selection data set names, instruction data set name, and additional control information used by aggregate backup in determining which data sets are to be backed up. You create an aggregate group using Interactive Storage Management Facility (ISMF) panels.

The aggregate group optionally contains a management class name that is used to obtain a management class definition that specifies the new aggregate backup parameters.

If a management class is not specified in the aggregate group definition, then DFSMSHsm attempts to retrieve the contents of the default management class during aggregate backup processing. If a default management class has not been defined, then DFSMSHsm defaults are used. See [“Management class considerations”](#) on page 338 for information about these defaults.

Management class

The management class specifies such parameters as the maximum number of aggregate versions to be kept, the aggregate group retention parameters, the serialization option, and whether concurrent copy is requested.

Selection data set

The selection data set lists the names of the data sets to be processed during aggregate backup. You can identify the data set names in a single selection data set, or you can divide the names among as many as five selection data sets.

You can specify six types of data set lists in a selection data set: INCLUDE, EXCLUDE, ACCOMPANY, ALLOCATE, ACCOMPANYEXCLUDE, and ALLOCATEEXCLUDE. The type you specify determines which data sets are backed up and how they are recovered.

Your selection data set must have at least one INCLUDE, ALLOCATE, or ACCOMPANY data set.

An INCLUDE data set list is a list of data sets to be copied by aggregate backup to a tape data file where they can be transported to the recovery site and recovered by aggregate recovery. The list can contain fully qualified data set names or partially qualified names with placeholders. DFSMSHsm expands the list to fully qualified data set names.

An EXCLUDE data set list is a list of fully qualified or partially qualified data set names that are specifically excluded from being backed up. If the expansion of the partially qualified name results in one or more fully qualified names that should not be backed up, the EXCLUDE keyword prevents those data sets from being backed up.

An ACCOMPANY data set list is a list of tape data sets that physically accompany the control and data file from the backup site to the recovery site and that are cataloged as part of an aggregate recovery. These data sets are *not* written to the data file. You have to provide your own copy of the data set to the recovery site. For example, you may want to specify in the ACCOMPANY list any large tape data sets that would take a lot of time to copy to the data file.

An ACCOMPANYEXCLUDE data set list is a list of fully qualified or partially qualified data set names that are specifically excluded from being tracked in the control file during aggregate backup and eventually cataloged during aggregate recovery. If the expansion of the partially qualified name results in one or more fully qualified names that should not be processed, the ACCOMPANYEXCLUDE keyword prevents those data sets from being backed up.

An ALLOCATE data set list is a list of DASD data sets or GDG base names processed by aggregate backup and specified on the control file to be allocated and cataloged at the recovery site. ALLOCATE data sets are *not* copied to a data file. For example, you may want to specify an ALLOCATE data set for aggregate processing that requires the presence of an empty data set.

Note: An ALLOCATE data set list can also contain the names of ML2 migrated tape data sets.

An ALLOCATEEXCLUDE data set list is a list of fully qualified or partially qualified data set names that are to be specifically excluded from having allocation data collected during ABACKUP. If the expansion of the partially qualified name results in one or more fully qualified names that should not be allocated at the recovery site, the ALLOCATEEXCLUDE keyword prevents collecting allocation data for those data sets.

Skipped data sets

Regardless of how you specify data sets for backup, some data sets may be protected by the peer-to-peer remote copy (PPRC) function or by the extended remote copy (XRC) function. In either case, making backup copies is redundant and a waste of storage space. Use the ABACKUP SKIP(PPRC) command to direct ABACKUP to skip backing up any level 0 data sets that are totally contained on PPRC volumes. Use the ABACKUP SKIP(XRC) command to direct ABACKUP to skip backing up any level 0 data sets that are totally contained on XRC volumes. In addition, you can combine the parameters in the ABACKUP SKIP(PPRC XRC) command form.

Note: You can also use the SETSYS ABARSKIP(PPRC) command or the SETSYS ABARSKIP(XRC) command (or a combination of both subparameters) to specify which types of data sets that you want all ABACKUP processing to skip.

Instruction data set

The instruction data set is a data set you create that contains instructions or other information you want to send to the recovery site. The instruction data set is a free-form text file.

Backing up applications

Aggregate backup allows you to back up user-defined groups of data sets to be recovered later as an entity at another computer site. Figure 36 on page 324 shows the information flow of an aggregate backup.

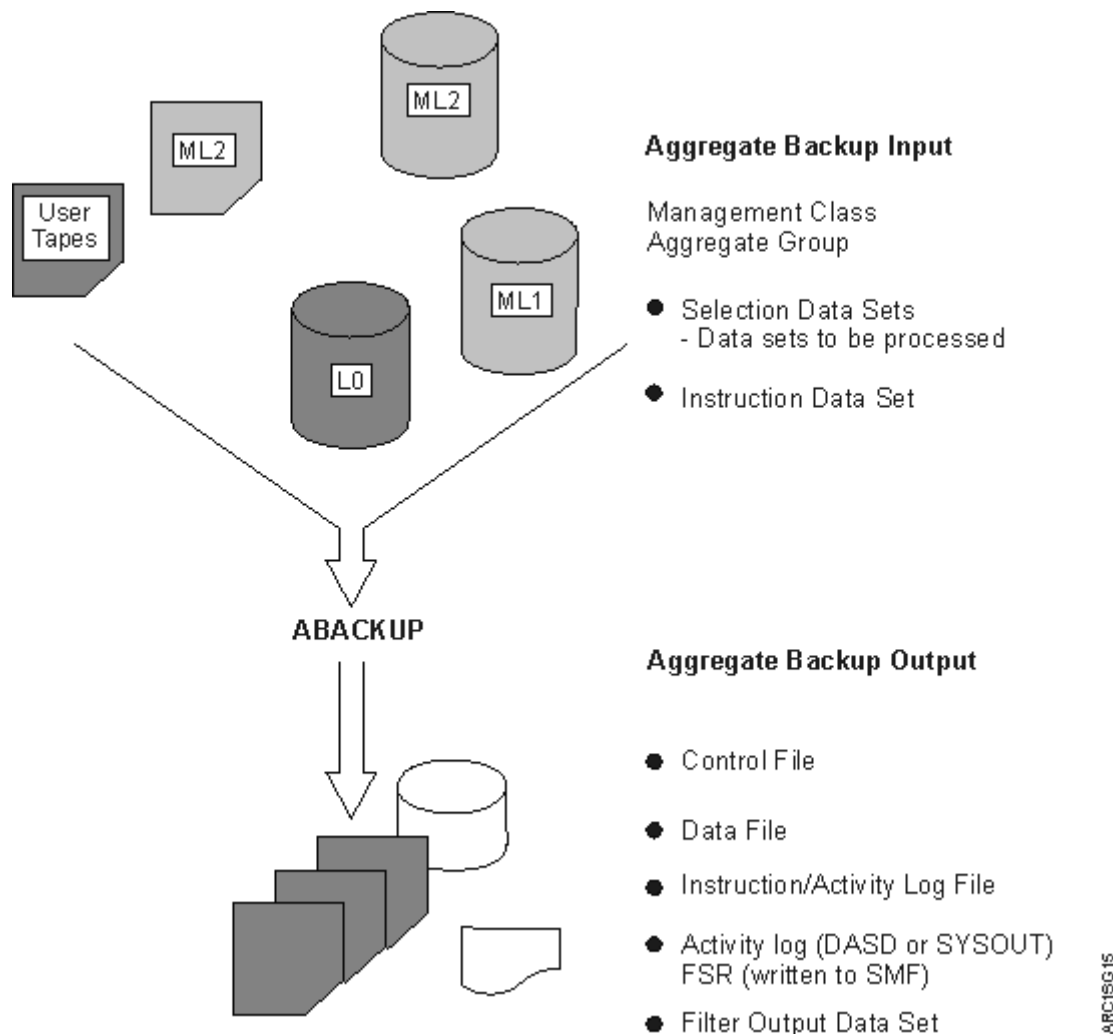


Figure 36. Aggregate Backup Information Flow

ABACKUP command

Two options are available with the ABACKUP command. You can choose to verify the contents of the aggregate backup without actually backing up any data sets, which is the same as performing a test run of aggregate backup. Then, when you are ready, you can perform the actual backup.

When you enter the ABACKUP command with the EXECUTE option, the following tape files are created for later use as input to aggregate recovery:

- Output data set tape files, which contain copies of the backed up data sets, as follows:
 - "D" data file, which contains copies of level 0 data sets that have been backed up.
 - "O" data file, which contains copies of migrated data sets or user tape data sets that have been backed up.

Note: The "D" file is created when you have level 0 disk data sets in the INCLUDE list. The "O" data file is created when you have user tape data sets or migrated data sets in the INCLUDE list. If you requested

stacking for the ABACKUP output files and you do not specify an INCLUDE statement, a dummy "D" and "O" file is created. If you did not request stacking for the ABACKUP output files and you do not specify an INCLUDE statement, no "D" or "O" files are written. For the format of these files, see [“Data files”](#) on page 351.

- A control file. This file contains control information needed by aggregate recovery to verify or to recover the application's data sets. The control file also specifies which ACCOMPANY and ALLOCATE data sets should be cataloged or allocated at the recovery site.
- An instruction/activity log file. This file contains the instruction data set, which is optional. The instruction data set contains the instructions or information that may be needed at the recovery site. A copy of the activity log is also written to this file if the activity log is directed to DASD by the SETSYS ABARSACTLOGTYPE command.

Note: This file is created only if an instruction data set name was specified in the aggregate group, or the ABACKUP activity log was allocated on DASD.

Also, aggregate backup creates the following:

- Filter output data set—Created if requested with the FILTEROUTPUTDATASET parameter of the ABACKUP command.
- Activity log—Contains messages issued during aggregate backup processing.
- Functional statistics record (FSR)—Containing additional aggregate backup processing information. This includes records of the amount of CPU time used for processing ABACKUP and ARECOVER requests that can optionally be written as an SMF record if the installation specifies SETSYS SMF(*smfid*).

Note: If the total number of volumes used for the data files, control file, instruction data set, and accompanying data set exceeds 5000, an extension FSR is created for the aggregate. Each extension FSR contains a maximum of 5000 volume serial numbers. In addition, each extension FSR has the constant "CONT" at offset X'7C'. There are as many extended FSRs as necessary to hold all volume serial numbers processed by a particular aggregate. The extension FSRs repeat all aggregate information except the volume serial number lists.

- An ABARS activity record (ABR) is created in the BCDS. This record contains information related to an aggregate backup and recovery performed on a particular aggregate group.

Rule: You must have a BCDS when performing an aggregate backup or aggregate recovery.

Example of data sets associated with an aggregate group

Table 34 on page 325 shows the data sets that exist in the database at the aggregate backup site. Throughout this topic, this example is used to show each task as you prepare for and perform aggregate backup of some of these data sets.

Table 34. Data Sets at the Example Aggregate Backup Site

Data Set Name	DASD	Tape	SMS	Migrated
PAY2.PY002.SOCSEC	X		X	
PAY2.PY002.STATUS.G0001V00	X			
PAY2.PY002.STATUS.G0002V00	X			
PAY1.PY001.CHECK	X			
PAY1.PY003.CHECK	X			
PAY2.PY001.BENEFIT	X			
PAY1.PY001.PGMLIB	X			
PAY1.PY002.IRA	X			ML1
PAY1.PY001.BONUS	X			ML1

Table 34. Data Sets at the Example Aggregate Backup Site (continued)

Data Set Name	DASD	Tape	SMS	Migrated
PAY1.PY002.LADDER	X			ML1
PAY1.PY001.NAMES		X		
PAY2.PY002.RETIRE		X		
PAY2.PY001.TAXES		X		

You want the following data sets to be copied to the data file:

- PAY2.PY002.SOCSEC
- PAY2.PY002.STATUS.G0002V00
- PAY1.PY001.CHECK
- PAY1.PY001.PGMLIB
- PAY1.PY002.IRA
- PAY1.PY001.BONUS
- PAY1.PY002.LADDER
- PAY1.PY001.NAMES

You want the data set PAY2.PY001.BENEFIT to be identified on the control file for allocation at the recovery site. You also want the data set PAY2.PY001.TAXES to be identified on the control file and to be cataloged at the recovery site.

Filtering for generation data sets

Extra care must be taken when using wildcard filtering for generation data sets. You must use either of the following names:

- The fully qualified data set name
Example: USER1.GDG.G0001V00
- The fully qualified name with relative generation
Example: USER1.GDG(-1)
- The partially qualified base name with the last qualifier (GggggVvv) not specified
Example: USER1.GDG.*

Partial qualification of the last qualifier (GggggVvv) is not supported unless a wildcard (%) or (*) is specified in the first character of the qualifier

Example: USER1.GDG.*V01 is valid but USER1.GDG.G* is not valid.

Another way to get all generations of a GDG using wildcards would be to specify USER1.GDG.%%%%%V%%.

Preparing for aggregate backup

You need to consider the following items as you prepare for aggregate backup:

- Minimizing data set contention on your system.

When ABACKUP is processing data sets on level 0 volumes and the status of a data set changes (for example, it is moved or deleted) after the list of data sets to process has been created but before backup processing has begun, ABACKUP processing fails. Then ABACKUP processing must be entirely rerun.

- Specifying data sets in one or more selection data sets so they can be processed by aggregate backup

- Writing instructions and useful information in an optional instruction data set
- Defining an aggregate group to identify the selection data sets, instruction data set, and other control information for aggregate backup
- Defining a management class
- Defining aggregate backup environmental options with the SETSYS command
- Using RACF facility class to authorize certain users or operators to issue the ABACKUP command

Table 35 on page 327 shows a checklist you might want to follow as you complete each of the steps needed to prepare for and perform aggregate backup.

Table 35. Aggregate Backup Checklist	
Step	Action
___ 1.	Negotiate with the recovery site to provide the support required to recover your data sets.
___ 2.	Review <i>z/OS Planning for Installation</i> .
___ 3.	Define the aggregate group to identify the selection data sets, the instruction data set, and any additional control information.
___ 4.	Define the management class.
___ 5.	<p>Create the selection data sets that list the data sets to be processed by aggregate backup. See <i>z/OS DFSMSHsm Implementation and Customization Guide</i>, Chapter 4, under the heading "Data Mover Support for Availability Management Functions" for a list of data set types that are supported for aggregate backup.</p> <ol style="list-style-type: none"> 1. Allocate the selection data sets. 2. Enter an INCLUDE list of the data sets to be backed up. This step is optional; however, if you omit an INCLUDE list, you must at least specify an ALLOCATE or ACCOMPANY list. 3. Enter an EXCLUDE list of the data sets that should not be backed up. This step is optional. 4. Enter an ACCOMPANY list of the tape data sets that should be taken with the aggregate backup data file and should be cataloged at the recovery site. This step is optional. 5. Enter an ACCOMPANYEXCLUDE list of data sets you want excluded from the ACCOMPANY list. This step is optional. 6. Enter an ALLOCATE list of data sets you want only allocated and cataloged at the recovery site. This step is optional. 7. Enter an ALLOCATEEXCLUDE list of data sets you do not want allocated and cataloged at the aggregate recovery site. This step is optional.
___ 6.	Create an instruction data set to contain any helpful information or instructions for the recovery site. This is optional.
___ 7.	Link-edit the aggregate backup installation exits. This step is optional.
___ 8.	Ensure that all level 0, ML1, and ML2 DASD volumes containing the data sets referred to for aggregate backup are online.
___ 9.	Ensure that the ABARS JCL procedure was installed in the systems JCL library (refer to the <i>z/OS DFSMSHsm Implementation and Customization Guide</i>).

Table 35. Aggregate Backup Checklist (continued)

Step	Action
___ 10.	Determine the SETSYS parameters to be used: <ul style="list-style-type: none"> • Specify maximum number of ABARS address spaces • Specify activity log type, message level, and deletion processing • Specify whether output tapes should be stacked or not • Specify whether you want the ABARS FSR record written as an SMF record • Specify how DFSMSdss backs up level 0 DASD during ABACKUP processing • Indicate ABARS procedure name • Indicate TAPESECURITY(RACF) if you have decided to use RACF TAPEVOL tape security. • Specify TAPESECURITY(EXPIRATION) if RACF DATASET security is to be used. • Activate desired installation exits • Specify ABARS buffer usage
___ 11.	Add the aggregate backup procedure name defined by SETSYS ABARSPROcname to the RACF started task table.
___ 12.	Use RACF FACILITY class to authorize certain users or operators to issue the ABACKUP command.
___ 13.	If you want to RACF protect output tape volumes, you can issue the SETSYS command with the TAPESECURITY(RACF) parameter and issue the RACF command RDEFINE TAPEVOL HSMABR; or you can instead use RACF DATASET class profiles to protect the tapes and create the DATASET profiles using ADDSD
___ 14.	Issue the ABACKUP command with the VERIFY parameter. <ul style="list-style-type: none"> • If aggregate backup verification completes successfully (RC=0), continue to step 16. • If aggregate backup verification fails, correct any errors and repeat this step until the aggregate backup verification completes with a return code of zero.
___ 15.	Issue the ABACKUP command with the EXECUTE parameter.
___ 16.	Review the activity log for messages ARC6060I and ARC6061I that list the aggregate backup control and data files volumes. The control file volume serial numbers are required as input for aggregate recovery.

SMS requirements during aggregate backup

You must have SMS active on your system to perform aggregate backup.

Prior planning is suggested if you want to back up SMS-managed data sets and have them recovered later in an environment compatible to that which existed at the backup site. It is recommended that the SMS classes retain the same attributes both at the backup and at the recovery sites. The ACS routines should also be compatible.

If you do not have identical SMS environments at both sites, SMS-managed data sets can still be successfully recovered, but not necessarily in the environment in which they existed at the backup site. The SMS-managed data sets are recovered in one of the ways described in [“Considering storage management subsystem requirements”](#) on page 378.

Note:

1. You might want to specify the required SMS class names and attributes in the instruction data set when you perform aggregate backup. See [“Creating an instruction data set”](#) on page 334.
2. All SMS class names that are encountered during aggregate backup processing—along with their associated attributes—are listed in the ABACKUP activity log. This information can be retrieved during ARECOVER PREPARE processing and used to define the SMS environment at your recovery site. You should also consider backing up the source control data sets (SCDSs) as a separate aggregate group so that you can recover them as an aggregate before recovering application data.

Creating selection data sets

A selection data set contains the names of the data sets to be processed during aggregate backup. You can identify the data set names in one selection data set, or you can split them up and list them in as many as five selection data sets.

For example, you may want to list a particular group of data sets in a separate selection data set that could then be used as part of more than one aggregate backup. Thus, a particular user's billing application could be listed in a separate selection data set to be backed up by itself. Then, if you decide you also want to back up the billing applications of all the users in your company, you can perform an aggregate backup using this same selection data set in addition to selection data sets for the other users.

A selection data set can be a partitioned data set (PDS) member, a sequential data set, or a partitioned data set extended (PDSE) member.

The Interactive Storage Management Facility (ISMF) panels allow you to allocate, browse, and edit the selection data sets.

If you create a selection data set outside of ISMF panels, you must allocate it as follows:

- Records must be in one of these two formats:
 - Fixed format, block size = 80, length = 80
 - Fixed block format, block size = any multiple of 80, length = 80
- Selection data sets must be cataloged.

Selection data set syntax

You must follow these general rules when creating a selection data set:

- Entries must be entered between columns 1 and 72 inclusive.
- A comment is a string of characters preceded by /* (forward slash and asterisk) and followed by */ (asterisk and backslash) symbols.
- A separator consists of a comma, one or more blanks, or a comment.
- INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE keywords are separated from one another by one or more separators.
- One or more blanks can optionally precede and follow each parenthesis in the pair.
- Continuation characters are not necessary but can be optionally specified by a hyphen (-) or plus (+) as the rightmost nonblank character, optionally preceded by one or more blanks. This follows the syntax conventions for TSO CLIST.
- INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE keywords can be specified in any sequence, but they can be specified only once within a selection data set.
- All keywords are optional. You must, however, specify at least one INCLUDE, ALLOCATE, or ACCOMPANY keyword.
- Data set names specified in the INCLUDE list cannot be specified in the ALLOCATE or ACCOMPANY lists.
- If the same fully qualified data set name is specified in both the INCLUDE list and the EXCLUDE list, the data set is not selected for processing, because EXCLUDE takes precedence over INCLUDE.

If the same fully qualified data set name is specified in both the ACCOMPANY list and the ACCOMPANYEXCLUDE list, the data set is not selected for processing, because ACCOMPANYEXCLUDE takes precedence over ACCOMPANY.

If the same fully qualified data set name is specified in both the ALLOCATE list and the ALLOCATEEXCLUDE list, the data set is not selected for processing, because ALLOCATEEXCLUDE takes precedence over ALLOCATE.

- Fully qualified names are supported.
- PDS member names are not supported.

- Fully or partially qualified data set aliases are not supported.
- Relative GDG names are supported when the GDG name is a fully qualified data set name without placeholders.

Partially qualified relative GDG names are supported in the INCLUDE statement, but not in the EXCLUDE, ALLOCATE, ALLOCATEEXCLUDE, ACCOMPANY, or ACCOMPANYEXCLUDE statements.

The following is an example of a partially qualified relative GDG name:
INCLUDE (USERHLQ.*.GDG.*(-1))

The statement above causes aggregate backup to back up all the minus-one generation data sets that are cataloged and match the specified filtering criteria. No other generations of the data set are backed up unless specifically requested by another data set filter or a fully qualified data set name.

Rule: You cannot specify positive generation (+*nnn*) numbers. Most positive generation numbers refer to data sets that are not yet created. In addition, the integer value within the parentheses must conform to MVS constraints and must be expressed in three digits or less.

- With the exception of relative GDG names (described above), partially qualified data set names are supported in the INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE lists. Use the following examples of partially qualified data set names as guidelines for the use of ** (double asterisk) and * (single asterisk):

PAY1.**

Specifies all data sets whose first or only qualifier is PAY1

USER2..LIST**

Specifies all data sets whose first and last qualifiers are USER2 and LIST, respectively, including USER2.LIST

PAY1.*

Specifies all data sets with two qualifiers, whose first qualifier is PAY1

PAY1.*.LOAD

Specifies all data sets with three qualifiers and whose first and last qualifiers are PAY1 and LOAD respectively

PAY1.UT*.LIST

Specifies all data sets with three qualifiers, whose first and last qualifiers are PAY1 and LIST, respectively, and whose second qualifier begins with UT

When used with other qualifiers, the double asterisk (**) indicates either the nonexistence of leading, trailing, or middle qualifiers, or the fact that they play no role in the selection process.

The single asterisk (*) is used in place of exactly one qualifier. It can also be used to indicate that only part of a qualifier has been specified.

The ignore character % (percent sign) is a placeholder character. Each % represents exactly one character in the data set name. Any character in that position in the name is ignored. Use the following examples as guidelines for the use of ignore character:

PAY1.PY001.I%%

This matches a data set name of PAY1.PY001.IRA.

PAY1.PY001.B%%%S

This matches a data set name of PAY1.PY001.BONUS.

ABARS supports both VSAM and non-VSAM data sets in the selection data set. ABARS allows VSAM base cluster names or VSAM component names in the selection lists. If VSAM component names are specified in the selection data set, ABACKUP processing determines the correct base cluster name that is associated with the component name and backs up the base cluster. If multiple VSAM component names are associated with the same base cluster name, the base cluster is backed up only once.

If a VSAM component name is encountered, ABACKUP processing backs up the associated base cluster and all components that are associated with that base cluster. When ABACKUP is invoking DFSMSdss, it passes DFSMSdss the base cluster name and requests SPHERE processing.

Note: If you are working with an ALLOCATE list, only the base cluster components will be processed.

When a VSAM data set is backed up from the INCLUDE list, all of its associations are backed up, including its components (base cluster, data, and index). However, when a VSAM data set is backed up from the ALLOCATE list, only the base cluster is backed up. This occurs because DFSMSdss dump processing moves actual data for a data set that is backed up from the INCLUDE list. No data (only allocation information that is necessary to allocate the data set at the recovery site) is moved for data sets backed up from the ALLOCATE list. In each instance, all information that is needed to restore the VSAM data set is backed up regardless in which list it appears. For a discussion of different data set types that ABARS supports in the INCLUDE and ALLOCATE lists, refer to [Data set type support for availability management functions in z/OS DFSMSHsm Implementation and Customization Guide](#).

Because of the way that ABARS handles VSAM components and because migrated VSAM data sets are cataloged as non-VSAM, the results of data set filtering can vary, depending on whether the data set is migrated and whether the VERIFY or EXECUTE keywords are specified in the ABACKUP command.

For example, if you have the following statements in your selection data set:

```
INCLUDE(hlq.TEST.CLUSTER.**)  
EXCLUDE(hlq.TEST.CLUSTER)
```

Then you could still back up the base cluster because the data and index components have not been excluded.

These statements work if the VSAM data set is not migrated. However, if the data set is migrated, the data component is located (ABARS does not know at the time it is VSAM) and the base cluster is backed up.

To ensure that all components of the above VSAM cluster are excluded, either exclude using a matching generic data set name; for example,

```
EXCLUDE(hlq.TEST.CLUSTER.**)
```

or exclude all fully qualified component names; for example,

```
EXCLUDE(hlq.TEST.CLUSTER.DATA,hlq.TEST.CLUSTER.INDEX)
```

Related reading

- For more information about how to use the INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE keywords, see [“Selection data set keywords” on page 331](#).
- For more information about the TSO CLIST command, see [z/OS TSO/E Command Reference](#).

Processing user tape data sets in the include list

User tape data sets in the INCLUDE list are supported up to a block size of 256K-1, which is currently the maximum size supported for tape devices. The tape data set is restored to its original size upon recovery, as long as the allocated tape device supports the block size.

Selection data set keywords

The following six keywords can be used to identify data sets in a selection data set:

- INCLUDE
- EXCLUDE
- ACCOMPANY
- ACCOMPANYEXCLUDE
- ALLOCATE
- ALLOCATEEXCLUDE

You must specify at least one INCLUDE, ALLOCATE, or ACCOMPANY.

INCLUDE

Data sets listed by this keyword are written to the aggregate backup data file during ABACKUP processing. The INCLUDE keyword syntax is:

Keyword	Field
Include	(<i>dsn...</i> , <i>dsnmask...</i>)

- **Include** — Enter the whole word or an abbreviation of the word. The letter I is the minimum abbreviation allowed.
- *dsn...*, *dsnmask...* — Specifies the data set names or masks to be backed up. *dsnmask* represents a partially qualified data set name. Data sets to be backed up can be tape or DASD, migrated or nonmigrated, SMS-managed or non-SMS-managed. If a PDS is listed, all members are included in the aggregate backup. If you specify a fully qualified data set name, the data set must exist and must be cataloged, or aggregate backup fails.

Note: DFSMSHsm backs up, as is, any migrated data set in the INCLUDE list, without recalling it first.

EXCLUDE

Data sets listed by this optional keyword are *not* written to the aggregate backup data file. If you specify the same fully qualified data set name in both the INCLUDE and EXCLUDE lists, the data set is not selected, because EXCLUDE takes precedence over INCLUDE.

The EXCLUDE keyword syntax is:

Keyword	Field
Exclude	(<i>dsn...</i> , <i>dsnmask...</i>)

- **Exclude** — Enter the whole word or an abbreviation of the word. The letter E is the minimum abbreviation allowed.
- *dsn...*, *dsnmask...* — Specifies the data set names or masks to be excluded from the aggregate backup. *dsnmask* represents a partially qualified data set name.

Note: Aggregate backup combines the INCLUDE and EXCLUDE lists from all selection data sets for processing. The combination of lists must result in the selection of at least one data set, or aggregate backup fails.

ACCOMPANY

Tape data sets listed with this optional keyword physically accompany the control and data files from the aggregate backup site to the aggregate recovery site and are cataloged as part of an aggregate recovery. These data sets are *not* written to the data file. You must provide your own copy of the data set to be used at the aggregate recovery site.

Example: You can have a large tape data set that would take too much time to copy to the data file. If you can specify this data set in the ACCOMPANY list, and the data set is cataloged at the aggregate recovery site. Then you must transport the tape containing the data set to the aggregate recovery site.

The ACCOMPANY keyword syntax is:

Keyword	Field
ACCOMPANY	(<i>dsn...</i> , <i>dsnmask...</i>)

- **ACCOMPANY** — Enter the whole word, or the alias AC.

- *dsn..., dsnmask...* — Specifies the partially and fully qualified tape data set names or masks to accompany the data file from the aggregate backup site to the aggregate recovery site.

ACCOMPANYEXCLUDE

Data sets listed by this optional keyword are excluded from the list of ACCOMPANY data sets. Data sets in the ACCOMPANY list must not be in the INCLUDE or ALLOCATE lists, and must be cataloged tape data sets.

The ACCOMPANYEXCLUDE keyword syntax is:

Keyword	Field
ACCOMPANYEXCLUDE	<i>(dsn..., dsnmask...)</i>

- ACCOMPANYEXCLUDE — Enter the whole word or an abbreviation of the word. ACCEXCLUDE is the alias, and ACCOMPANYE is the minimum abbreviation.
- *(dsn..., dsnmask...)* — Specifies the data set names or masks to be excluded from the aggregate backup. *dsnmask* represents a partially qualified data set name.

ALLOCATE

DASD data sets listed with this optional keyword have their space allocated and are cataloged at the aggregate recovery site as part of an aggregate recovery. These data sets are *not* written to the data file.

The ALLOCATE keyword syntax is:

Keyword	Field
ALLOCATE	<i>(dsn..., dsnmask...)</i>

- **ALLOCATE** — Enter the whole word or the alias AL.
- *dsn..., dsnmask...* — Specifies the partially and fully qualified DASD data set names or masks to be allocated and cataloged at the aggregate recovery site as part of an aggregate recovery. Data sets in the ALLOCATE list must not be in the INCLUDE or ACCOMPANY lists, and must be cataloged DASD data sets. The DASD data sets must conform to one of the following data set organization types:
 - Physical sequential (PS)
 - Partitioned organization (PO)
 - Direct access (DA)
 - Virtual storage access method (VSAM)
 - Key sequenced data sets
 - Linear data sets
 - Relative record data sets
 - Entry sequenced data sets
 - User catalog
 - GDG base name

All other data set organizations are *not* supported.

To determine how to allocate space at ARECOVER time, DFSMSHsm recalls each migrated data set included in the ALLOCATE list.

Rule: During aggregate recovery, VSAM data sets from the ALLOCATE list must be recovered as SMS-managed data sets. Therefore, SMS must be active at the recovery site, and ACS filtering must direct these data sets to SMS-managed storage.

ALLOCATEEXCLUDE

Data sets listed by this optional keyword are excluded from the allocate list.

The ALLOCATEEXCLUDE keyword syntax is:

Keyword	Field
ALLOCATEEXCLUDE	<i>(dsn..., dsnmask...)</i>

- **ALLOCATEEXCLUDE** — Enter the whole word or an abbreviation of the word. ALLEXCLUDE is the alias, and ALLOCATE is the minimum abbreviation.
- *dsn..., dsnmask...* — Specifies the data set names or masks to be excluded from the aggregate backup. *dsnmask* represents a partially qualified data set name.

Examples of selection data sets

At the example site, two selection data sets are allocated: PAY1SDS1 and PAY1SDS2.

PAY1SDS1 contains the following lines:

```
INCLUDE (PAY1.**)
EXCLUDE (PAY1.PY%%3.**)
```

PAY1SDS1 causes aggregate backup to back up all data sets whose names begin with PAY1, excluding any data sets with the middle qualifier PY%%3. Use of the % character has been explained in a previous topic.

PAY1SDS2 contains the following lines:

```
INC (PAY2.PY002.SOCSEC -
      PAY2.PY002.STATUS.G0002V00 -
      PAY2.PY002.RETIRE)
ACC (PAY2.PY001.TAXES)
ALLOC (PAY2.PY001.BENEFIT)
```

PAY1SDS2 causes aggregate backup to back up all data sets listed. In addition, PAY2.PY001.TAXES is processed as an ACCOMPANY data set and PAY2.PY001.BENEFIT is processed as an ALLOCATE data set.

Creating an instruction data set

Creating the instruction data set is optional. The instruction data set contains instructions or any other information you think is necessary to assist the recovery site in recovering and running the application. It is an informal, free-form text file.

You may find it useful to include the following types of information in the instruction data set:

- SMS attributes
- Description of the application
- RACF environment
- Software requirements
- Hardware requirements
- Unique application execution instructions

If you decide to back up the instruction data set, you must create it before you issue the ABACKUP command. The instruction data set must be a sequential data set and it must be cataloged. ISMF panels allow you to allocate, browse, and edit the instruction data set.

The instruction data set can be recovered during the aggregate recovery prepare step, before actually recovering the aggregate data sets. This can be helpful, especially if people other than the backup site personnel will run the application at the recovery site.

You should include the instruction data set name in the aggregate group definition so that the instruction data set is identified to aggregate backup. During aggregate backup, the instruction data set is backed up to an instruction and activity log file.

Related reading

- For more information about creating the instruction data set, see [z/OS DFSMSdfp Storage Administration](#).
- For more information about the aggregate group, see [“Defining an aggregate group”](#) on page 336.

Example instruction data set

The instruction data set, PAYROLL.INSTRUCT, that was created at the example site is shown in [Figure 37](#) on page 335. PAYROLL.INSTRUCT is used for the example site aggregate group, PAY1.

```
THIS IS THE INSTRUCTION DATA SET FOR AGGREGATE GROUP PAY1.

PERFORMING AGGREGATE BACKUP ON PAY1 BACKS UP THE CRITICAL PAYROLL FILES AT
THE END OF A PAY PERIOD. RELEVANT FILES INCLUDE THE EMPLOYEE ROSTER
(PAY1.PY001.NAMES), AMOUNT OF EACH PAYCHECK (PAY1.PY001.CHECK), AMOUNT FROM
EACH CHECK PLACED IN IRA AND SOCIAL SECURITY DEDUCTION
(PAY1.PY002.IRA AND PAY2.PY002.SOCSEC), AND AMOUNT (IF ANY) OF CHECK
THAT WAS A BONUS IN ADDITION TO SALARY (PAY1.PY001.BONUS).

TO ENSURE PROPER EXECUTION, VERIFY THAT CORRECT RELEASE LEVELS ARE PRESENT FOR
DFSMSHsm, DFSMSdss, AND DFSMSdfp; AND THAT THE CORRECT SMS CLASSES ARE ACTIVE.

YOU WILL NEED THE FOLLOWING SMS CLASSES AND THEIR ASSOCIATED ATTRIBUTES
SET UP TO RECOVER THOSE FILES THAT WERE SMS-MANAGED AT ABACKUP, AND TO
ENSURE THAT THEY WILL BE SMS-MANAGED UPON RECOVERY.

DATA CLASS NAME: PHYSSEQ
DESCRIPTION: PHYSICAL SEQUENTIAL DATA SETS

RECFM          : FB
LRECL          : 80
SPACE
  AVGREC       : U
  PRIMARY      : 50
  SECONDARY    : 50
RETPD          : 175
VOLUME COUNT   : 1

STORAGE CLASS NAME: BATCHDASD
DESCRIPTION: DASD DATA SETS PROCESSED IN BATCH MODE

AVAILABILITY:
  GUARANTEED SPACE           : NO
  GUARANTEED SYNCHRONOUS WRITE : NO

MANAGEMENT CLASS NAME : BATCH
DESCRIPTION            : BATCH DATA SETS

EXPIRATION ATTRIBUTES
  EXPIRE AFTER DAYS NON-USAGE : 150
  EXPIRE AFTER DATE/DAYS      : 366

RETENTION LIMIT :
PARTIAL RELEASE : YES

MIGRATION ATTRIBUTES
  PRIMARY DAYS NONUSAGE : 7
  LEVEL 1 DAYS NONUSAGE : 45
  COMMAND/AUTO MIGRATE  : YES

GDG MANAGEMENT ATTRIBUTES
ROLLED OFF GDS ACTION : EXPIRE

BACKUP ATTRIBUTES
BACKUP FREQUENCY
# OF BACKUP VERSIONS (DS EXISTS) : 3
# OF BACKUP VERSIONS (DS DELETED) : 1
RETAIN DAYS ONLY BACKUP VERSION  : 366
(DS DELETED)
RETAIN DAYS EXTRA BACKUP VERSIONS : 100
ADMIN OR USER COMMAND BACKUP      : YES
AUTO BACKUP                        : YES

EXECUTE ARECOVER WITH THE VERIFY OPTION TO VERIFY THE AGGREGATE RECOVERY.
```

Figure 37. Example of an Instruction Data Set

Defining an aggregate group

The aggregate group identifies the selection data sets, instruction data set, and additional control information required to perform aggregate backup. The aggregate group is an SMS construct you must define through a set of ISMF panels before you can perform aggregate backup. You must activate the newly defined SMS configuration before issuing the ABACKUP command.

Note: If you modify the aggregate group definition, the SMS configuration should be reactivated. This is not required if you are only changing the contents of the selection or instruction data sets.

You must specify the following information in the aggregate group:

Aggregate group name

The name can be up to 8 characters long.

Selection data sets

Insert the names of one-to-five selection data sets you created for this aggregate backup. See [“Creating selection data sets” on page 329](#).

Description

This is an optional description of the aggregate group.

Copies

Insert the number of copies of the ABARS output files to be created. The maximum number of copies is 15, and the default is one copy.

Output data set prefix

The prefix identifies the output data sets created by aggregate backup. These output data sets are created with the following naming conventions:

- *outputdatasetprefix.D.CnnVnnnn* and *outputdatasetprefix.O.CnnVnnnn* for the data file
- *outputdatasetprefix.C.CnnVnnnn* for the control file
- *outputdatasetprefix.I.CnnVnnnn* for the instruction/activity log file

Because they all share a common output data set prefix and version number, it is easier to identify all the output data sets from one aggregate backup.

Example: If you specify the *outputdatasetprefix* as 'PAY1', the data, control, and instruction/activity log files would have the names PAY1.D.C01V0001, PAY1.O.C01V0001, PAY1.C.C01V0001, and PAY1.I.C01V0001, respectively.

Note: *Cnn* is the copy number and *Vnnnn* is the version number generated during the ABACKUP operation.

Since you can make up to 15 copies in a single ABACKUP, the copy number identifies to which copy the file belongs. The version number is the same for all copies made in a single ABACKUP operation.

Example: If an aggregate group definition indicates that two copies are to be made, the first time that you perform an ABACKUP function on that aggregate group, the following files are created:

```
PAY1.D.C01V0001
PAY1.O.C01V0001
PAY1.C.C01V0001
PAY1.I.C01V0001

PAY1.D.C02V0001
PAY1.O.C02V0001
PAY1.C.C02V0001
PAY1.I.C02V0001
```

The next time that you perform an ABACKUP on that aggregate group, you create the following files:

```
PAY1.D.C01V0002
PAY1.O.C01V0002
PAY1.C.C01V0002
PAY1.I.C01V0002

PAY1.D.C02V0002
PAY1.O.C02V0002
```



```
PAY1.C.C02V0002
PAY1.I.C02V0002
```

Related reading

For more information about how to define the aggregate group by using ISMF panels, see [z/OS DFSMSdfp Storage Administration](#).

Example aggregate group

Figure 38 on page 337 and Figure 39 on page 337 show the ISMF panels used to create the aggregate group, PAY1, at the example site.

```

Panel Utilities Scroll Help
-----
DGTDCAG1                      AGGREGATE GROUP DEFINE                      Page 1 of 2
COMMAND ===>

SCDS NAME      . . . . . : PAY1.SCDs
AGGREGATE GROUP NAME : PAY1

TO DEFINE AGGREGATE GROUP, SPECIFY:

DESCRIPTION ===> APPLICATION BACKUP EXAMPLES
=====>

BACKUP ATTRIBUTES
NUMBER OF COPIES . . . . . 1                (1 to 15)
MANAGEMENT CLASS NAME . . . . MCLASS1        (1 to 8 characters, to be
                                                defined in current SCDS)

OUTPUT DATA SET NAME PREFIX . . . PAY1      (1 to 33 characters)

ACCOUNT . . . . .                               (1 to 32 characters)

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 38. Aggregate Group ISMF Panel 1

```

DGTDCAG2                      AGGREGATE GROUP DEFINE                      Page 2 of 2
COMMAND ===>

SCDS NAME:      PAY1.SCDs
AGGREGATE GROUP NAME: PAY1

TO EDIT A DATA SET, SPECIFY NUMBER ===>      (1, 2, 3, 4, 5, or 6)
SELECTION DATA SETS:      (1 to 44 characters)
1 ===> PAY1SDS1
  MEMBER NAME ===>      (1 to 8 characters)
2 ===> PAY1SDS2
  MEMBER NAME ===>      (1 to 8 characters)
3 ===>
  MEMBER NAME ===>      (1 to 8 characters)
4 ===>
  MEMBER NAME ===>      (1 to 8 characters)
5 ===>
  MEMBER NAME ===>      (1 to 8 characters)

INSTRUCTION DATA SET:      (1 to 44 characters)
6 ===> PAY1.INSTRUCT

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 39. Aggregate Group ISMF Panel 2

Notice that the same name was specified for the aggregate group and the output data set prefix. This makes it easier to remember later which control and data files belong to a specific aggregate group.

Management class considerations

The ABARS aggregate group definition can contain a management class name that is used to obtain a management class definition. Use the management class definition to determine the following aggregate backup parameters:

1. The maximum number of aggregate group backup versions to be kept (a specific number of versions up to 9999, or NOLIMIT).

Note: Specifying a limit allows the creation of a new version to roll off (delete) all existing versions that exceed the current limit.

NOLIMIT specifies that no roll-off occurs.

2. Aggregate group backup RETAIN parameters for ABACKUP versions:

```
RETAIN ONLY VERSION = DAYS | WEEKS | MONTHS | YEARS(nnnn) | NOLIMIT
RETAIN EXTRA VERSIONS = DAYS | WEEKS | MONTHS | YEARS(nnnn) | NOLIMIT
```

Note: These values do not affect the expiration date on the actual tape labels.

3. The serialization option:

```
COPY SERIALIZATION = C | F | blank
```

Specify C (continue) for “tolerate an enqueue failure”, and F (fail) for “do not tolerate an enqueue failure”. Blank means a null entry.

4. The ABACKUP COPY TECHNIQUE. You can specify:

PREFERRED (P)

A concurrent copy session is requested for each data set. The aggregate backup continues if a concurrent copy session is not obtained for a particular data set.

REQUIRED (R)

A concurrent copy session is requested for each data set. The entire aggregate backup fails if a concurrent copy session is not obtained for a particular data set.

STANDARD (S)

A concurrent copy session not be requested (default).

VIRTUAL PREFERRED (VP)

A virtual concurrent copy session is requested for each data set. The aggregate backup continues if a virtual concurrent copy session is not obtained for a particular data set.

VIRTUAL REQUIRED (VR)

A virtual concurrent copy session is requested for each data set. The entire aggregate backup fails if a virtual concurrent copy session is not obtained for a particular data set.

CACHE PREFERRED (CP)

A cache-based concurrent copy session is requested for each data set. The aggregate backup continues if a cache-based concurrent copy session is not obtained for a particular data set.

CACHE REQUIRED (CR)

A cache-based concurrent copy session is requested for each data set. The entire aggregate backup fails if a cache-based concurrent copy session is not obtained for a particular data set.

Note: Use these additional considerations when using cache-based concurrent copy to process large data sets or volumes, or when using multiple tasks to process a large number of data sets:

- Use concurrent copy only during periods of light update activity for the data sets or volumes involved.
- Spread multiple concurrent copy jobs across as many LPARs as possible.
- Ensure that jobs and address spaces that will use concurrent copy are assigned a WLM service class with a high execution velocity. Never assign a discretionary goal to concurrent copy work.
- Ensure that volumes that will be processed will not be reserved by another system during the concurrent copy.

In addition:

- DFSMShsm generates a default expiration date of 99365 for the ABARS output files. (Installation exit ARCEDEXT is called, if active, to allow you to specify expiration dates such as 99000 that have special meaning to tape management products).
- When rolling off an ABARS version, DFSMShsm expires tape data sets without regard to the tape expiration date.
- DFSMShsm saves the management class name in the ABR record so that EXPIREBV can obtain the latest retain values.

Note:

1. If the management class specified in the aggregate group definition does not exist, ABACKUP fails.
2. If a management class is not specified in the aggregate group definition, ABACKUP continues using the default management class. If a default management class is not defined, ABACKUP uses the following DFSMShsm defaults:
 - RETAIN ONLY VERSION = NOLIMIT
 - RETAIN EXTRA VERSIONS = DAYS(14)
 - NUMBER OF VERSIONS = 2
 - COPY SERIALIZATION = STATIC
 - ABACKUP COPY TECHNIQUE = STANDARD
3. The default management class name is never stored in the ABR record.

DFSMShsm coexistence considerations for aggregate groups and management classes

In DFSMShsm, the aggregate group definition has been modified from previous versions of DFSMShsm to add the NUMBER OF COPIES and MANAGEMENT CLASS NAME attributes. The DESTINATION attribute is no longer used in DFSMShsm. The TOLERATE ENQUEUE FAILURE attribute has been moved to the management class and is now called the COPY SERIALIZATION attribute.

Use ISMF to redefine the aggregate group definitions when converting to DFSMShsm and DFSMSdfp;. If the aggregate group is not redefined, the existing aggregate group definitions are still valid, but all new aggregate group attributes, as well as all new management class aggregate backup attributes, revert to defaults.

The COPIES attribute in the aggregate group will default to 1.

The defaults for the management class attributes are:

- RETAIN ONLY VERSION = NOLIMIT
- RETAIN EXTRA VERSIONS = DAYS(14)
- NUMBER OF VERSIONS = 2
- COPY SERIALIZATION = FAIL
- ABACKUP COPY TECHNIQUE = STANDARD

The NUMBER OF VERSIONS aggregate backup attribute in the management class is the only attribute that ABARS processing looks at when rolling off excess ABARS versions. It does not consider the RETAIN ONLY VERSIONS or RETAIN EXTRA VERSIONS attributes during this process (these attributes are only considered when the EXPIREBV ABARSVERSIONS command is issued).

The NUMBER OF VERSIONS management class attribute works like the LIMIT parameter used when defining a GDG base. Each time an ABACKUP is performed for a particular aggregate it determines if this will create an excess version; if so, the oldest version is rolled off without regard to the RETAIN ONLY VERSIONS or RETAIN EXTRA VERSIONS attributes.

The RETAIN ONLY VERSION management class attribute is used during EXPIREBV processing when expiring excess ABARS versions. This attribute indicates how long the latest ABARS version should be retained.

The RETAIN EXTRA VERSIONS management class attribute is also used during EXPIREBV processing when expiring excess ABARS versions. This attribute indicates how long any ABARS version other than the latest version should be retained.

EXPIREBV processing for excess ABARS versions does not consider the NUMBER OF VERSIONS attribute when expiring ABARS versions.

The UNIT attributes that apply to the RETAIN ONLY VERSION and RETAIN EXTRA VERSIONS attributes are calculated in the following way:

- WEEKS are multiplied by 7
- MONTHS are multiplied by 30
- YEARS are multiplied by 365

Protecting aggregate backup tape volumes

Before aggregate backup control file, data file, and instruction/activity log file tape volumes can be protected with RACF, you must first define an aggregate backup tape connected set by issuing the following command:

```
RDEFINE TAPEVOL HSMABR
```

RACF must be active and the SETSYS TAPESECURITY(RACF) or TAPESECURITY(RACFINCLUDE) options must be in effect for DFSMSHsm to use RACF to protect aggregate backup output tape volumes and add them to the HSMABR tape connected set. For more RACF information see [“RACF environment for DFSMSHsm”](#) on page 619.

Note: These tapes are not automatically added to RACF unless the above conditions are met at the recovery site. If these conditions are met, the tapes are added to RACF (at the recovery site) when the ARECOVER command is issued for this aggregate.

If TAPESECURITY(RACF) or TAPESECURITY(RACFINCLUDE) is specified and TAPESECURITY(EXPIRATION) TAPESECURITY(EXPIRATIONEXCLUDE) is *not* specified, then the expiration date is not written to the tape header label, and the ARCEEXT installation exit is not invoked.

Tape volumes added to the HSMABR RACF tape connected set are automatically removed by aggregate version rolloff processing during aggregate backup processing, during EXPIREBV processing, or if an aggregate backup failure occurs after the output tapes are opened. A tape volume may be deleted from the HSMABR tape connected set by entering the following RACF command:

```
RALTER TAPEVOL HSMABR DELVOL(volser)  
ADDSD 'outputdatasetprefix.**'
```

If you use RACF DATASET class profiles instead of TAPESECURITY(RACF) to protect the tapes and create the DATASET profiles, you could define ADDSD to specify that ABARS created aggregates are protected. You need a profile for each aggregate group you have defined that uses a different aggregate group prefix.

Checking authorization during aggregate backup

By using RACF FACILITY class profile support, you can permit all console operators or any users, including those who are not DFSMSHsm authorized, to issue the ABACKUP command.

If the RACF FACILITY class profile support is installed and activated, DFSMSHsm checks the RACF profile of the console operators or users for authorization to issue the ABACKUP command. The user ID of OPER must be defined to RACF before console operators can be given authority to issue the ABACKUP command. If the user ID OPER is given authority to the ABACKUP command, then all console operators

are authorized to issue the ABACKUP command. You can also define group names to RACF, using the RACF ADDUSER command.

If the RACF FACILITY class profile is not active, only DFSMSHsm-authorized users and console operators are authorized to issue the ABACKUP command.

You can define the DFSMSHsm ABACKUP command to RACF by defining the following profile names:

Command	Parameter	RACF Profile
ABACKUP		STGADMIN.ARC.ABACKUP
ABACKUP	agname	STGADMIN.ARC.ABACKUP.agname

Through the use of the RACF PERMIT command, you can authorize specific user IDs to issue these commands.

Related reading

- For more information about the RACF FACILITY class control of access to all commands, see [“Controlling command access with RACF FACILITY class” on page 625.](#)
- For more information about the RACF FACILITY class control of access to all commands, see [z/OS DFSMSHsm Implementation and Customization Guide.](#)

Authorizing comprehensive ABACKUP command authority

When users are added to the STGADMIN.ARC.ABACKUP profile, they receive comprehensive command authority. When users are also authorized to the STGADMIN.ARC.ABACKUP.agname profile, they receive RESTRICTED instead of COMPREHENSIVE command authority. When users have COMPREHENSIVE command authority, they may then issue the ABACKUP command for any aggregate group, and no RACF authorization checking is performed on the data sets being processed during aggregate backup.

When you define a RACF FACILITY class profile for the DFSMSHsm ABACKUP command, the Universal Access (UACC) keyword on the RDEFINE command defaults to NONE. This means that the ABACKUP command cannot be issued until a user is authorized to that profile using the RACF PERMIT command.

You can issue the following RACF command to define profiles that prevent or limit the authority of users to issue the ABACKUP command:

```
RDEFINE FACILITY STGADMIN.ARC.ABACKUP
```

The following command authorizes a user to issue the ABACKUP command for all aggregate groups:

```
PERMIT STGADMIN.ARC.ABACKUP CLASS(FACILITY) -  
ID(userID) ACCESS(READ)
```

Related reading

- For more information about the RACF FACILITY class control of access to all commands, see [“Controlling command access with RACF FACILITY class” on page 625.](#)
- For more information about the RACF FACILITY class control of access to all commands, see [z/OS DFSMSHsm Implementation and Customization Guide.](#)

Authorizing restricted ABACKUP command authority

When users are added to the STGADMIN.ARC.ABACKUP.agname profile, they receive RESTRICTED command authority. If users with this RESTRICTED authority issue the ABACKUP command, they must have a minimum of READ access to all RACF-protected data sets in the aggregate group. If they do not have READ or READ/WRITE access to a RACF-protected data set, the ABACKUP command fails for that

data set during verification processing. Each data set that fails authorization checking is listed in the aggregate backup activity log.

If a data set is not RACF protected, it does not fail during verification processing because no RACF authorization checking is performed on it.

You can issue the following RACF command to define profiles that prevent or limit the authority of users to issue the ABACKUP command with a specific aggregate group name:

```
RDEFINE FACILITY STGADMIN.ARC.ABACKUP.PAY1
```

The following RACF command authorizes all console operators to issue the ABACKUP command with a specified aggregate group name:

```
PERMIT STGADMIN.ARC.ABACKUP.PAY1 CLASS(FACILITY) -  
ID(OPER) ACCESS(READ)
```

Effect of activating the RACF OPERCMDS class on restricted ABACKUP command processing

Activation of the RACF OPERCMDS class can affect what user ID ABACKUP uses to check authorization to the data sets being backed up when a restricted FACILITY class profile is defined and the ABACKUP command is issued from an operator's console. Typically, all commands issued from an operator's console (including extended consoles) are considered to be authorized, and no RACF checking of data sets in the aggregate is performed. However, if the RACF OPERCMDS class is active, DFSMSHsm is able to extract the user ID that issued the command and uses that user ID when checking authorization to the data sets that are being backed up.

Related reading

- For more information about the RACF FACILITY class control of access to all commands, see [“Controlling command access with RACF FACILITY class” on page 625](#).
- For more information about the RACF FACILITY class control of access to all commands, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about the ABARS RACF FACILITY and generic profiles, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Using the aggregate backup installation exits

Programming Interface Information

Four installation exits are used by aggregate backup if they are accessible to the LOAD macro and if SETSYS EXITON(ARCM2EXT), SETSYS EXITON(ARCBEXT), SETSYS EXITON(ARCTVEXT), and SETSYS EXITON(ARCEEXT) commands have been issued. (You can use the QUERY SETSYS command to display the status of these exits.)

ARCM2EXT is called by aggregate backup when a data set on an ML2 volume is encountered, and it allows you to specify if the data set should be skipped or included in the aggregate backup.

ARCBEXT is called when:

- A data set encounters an input data set I/O
- A CDS I/O is encountered
- An SDSP allocation error occurs
- An uncataloged data set is encountered
- A data set fails serialization of the MAJOR NAME = ARCDNS, MINOR NAME = *datasetname* resource
- An error occurs during DFSMSdss dump processing

ARCBEXT allows you to specify if the data set should be skipped so that the aggregate backup can complete successfully, or if it should be included in the aggregate backup and cause aggregate backup to fail.

ARCEEXT is called to allow you to specify a tape volume expiration date other than 99365 for the aggregate backup output files.

ARCTVEXT is called to indicate to tape management systems, excluding DFSMSrmm, that ABARS is expiring a tape volume under any of the following conditions:

- When rolling off old ABARS versions during ABACKUP processing
- When expiring ABARS versions (EXPIREBV ABARSVERSIONS)
- When ABACKUP processing fails

End Programming Interface Information

Related reading

For more information about installation exits, see [z/OS DFSMS Installation Exits](#).

Ensuring that the necessary volumes are online

Before performing aggregate backup, ensure that all level 0, ML1, and ML2 DASD volumes containing data sets to be processed by aggregate backup are online. If these volumes are not online during aggregate backup processing, a mount request is issued.

Defining SETSYS parameters for aggregate backup

The SETSYS command allows you to externally define various aggregate backup environmental options.

The system programmer should include SETSYS commands in the DFSMSHsm parmlib member (ARCCMDxx) to ensure that the settings are defined to DFSMSHsm at startup initialization. If you are an authorized DFSMSHsm user, you can issue the SETSYS command before an aggregate backup to modify the SETSYS parameters that control aggregate backup processing.

To specify how many ABACKUP commands can run concurrently, use the MAXABARSADDRESSSPACE parameter. MAXABARSADDRESSSPACE specifies the maximum number of concurrent ABARS secondary address spaces that DFSMSHsm supports, up to 64. When determining the value to specify for MAXABARSADDRESSSPACE, you should factor in such items as system load, resource contention, and aggregate backup and recovery activity. The overriding factor is the number of tape drives available. Aggregate backup uses at least one tape drive for each address space and uses two if backing up user tape data sets or data sets residing on ML2 tape (the default for MAXABARSADDRESSSPACE is 1).

Note: When an ABACKUP command completes, ABARS will determine if there is another ABARS request queued that has not been processed. If there is, DFSMSHsm will reuse the address space to process that command. The address space will remain active until there are no further ABARS commands queued, or the SETSYS MAXABARSADDRESSSPACE command has reduced the number of ABARS address spaces and this address space is no longer required.

Rule: If you issue two ABACKUP commands with the same aggregate group name, they cannot execute concurrently even if MAXABARSADDRESSSPACE is set to a value greater than 1.

The ABARSPROCNAME parameter specifies the procedure name that is used to start an ABARS secondary address space when no such address space is available for reuse.

The EXITON(*exitname*) parameter lets you specify which installation exits (ARCBEXT, ARCEEXT, ARCM2EXT, ARCTVEXT) are active during aggregate backup processing.

Each time you issue the ABACKUP command, an activity log is created. You can use the SETSYS ABARSACTLOGTYPE command to specify whether you want the ABACKUP activity log automatically written to SYSOUT or to DASD. You can use the SETSYS ABARSDELETEACTIVITY(Y | N) command to specify whether you want DFSMSHsm to automatically delete the ABARS activity log that corresponds to

the ABACKUP version being rolled off. The deletion occurs during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing, and removes the need to manually manage the ABARS activity logs.

You can use the SETSYS ABARSBUFFERS(*n*) command to specify the number of ABARS I/O buffers, up to a maximum of nine buffers. DFSMSHsm multiplies the number you specify by two to determine the total number of buffers to use for I/O processing (one-half for input, one-half for output). If the number of buffers is not specified by the SETSYS ABARSBUFFERS command, a default setting of one is used.

Use the ABARSKIP(PPRC XRC) parameter on the SETSYS command to specify that you want the backup to skip level 0 data sets that are protected by the peer-to-peer remote copy function or by the extended remote copy function.

The SETSYS ABARSTAPES(STACK) command allows you to place (stack) the ABACKUP output files from a single aggregate group on a minimum number of tape cartridges (as few as one). The stacking function applies only to tape cartridges. Attempts to redirect ABACKUP output files to DASD when the STACK option is in effect causes ABACKUP to fail.

You can select which DFSMSdss messages should be written to the ABACKUP activity log by using the SETSYS ABARSACTLOGMSG LVL(FULL | REDUCED) command. If you specify SETSYS ABARSACTLOGMSG LVL(FULL), all DFSMSdss messages are intercepted and written to the ABACKUP activity log. If you specify SETSYS ABARSACTLOGMSG LVL(REDUCED), only the DFSMSdss messages ending in a “W” (warning messages) or “E” (error messages) are written to the ABACKUP activity log; no informational messages are written.

ABARS records the CPU time for processing ABACKUP requests, along with any applicable account code information. This information is written to the ABR record and can be obtained using the LIST AGGREGATE command. It is also written to the FSR control block. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*). ABARS then writes the FSR as *smfid* + 1 in the SYS1.MANx or SYS1.MANY system data sets. Installations can use the information to calculate charge backs for ABARS requests.

The SETSYS ABARSOPTIMIZE(*n*) command allows installations to adjust performance when invoking DFSMSdss to back up level 0 DASD data sets that are specified in an ABARS INCLUDE data set list. You can specify that DFSMSdss read one, two, or five tracks at a time, or one cylinder at a time. You can override the SETSYS ABARSOPTIMIZE value on an individual ABACKUP command by issuing the ABACKUP command with the OPTIMIZE(*option*) parameter.

Note: If the SETSYS DEBUG option is in effect when the ABACKUP command begins processing, DFSMSHsm cannot move any data. The DEBUG option causes all ABACKUP commands to be processed as if the VERIFY option had been specified.

Related reading

- For more information about the SETSYS command, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about the SETSYS command, see Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.
- For more information about the ABARSPROCNAME parameter, see “ABARSPROCNAME: Specifying the base procedure name to start the ABARS secondary address space” on page 1096.

Example SETSYS commands for aggregate backup

The following SETSYS commands are entered for the example site before performing aggregate backup:

```
SETSYS MAXABARSADDRESSSPACE(2) ABARSPROCNAME(SAMPLE)
SETSYS ABARSACTLOGTYPE(DASD)
```

The maximum number of secondary address spaces is set to two, which means that only two ABARS commands can run at a time. The procedure name used to start the secondary address space is identified as SAMPLE, and the activity log will be allocated to DASD.

To review the SETSYS parameters in effect for aggregate backup and recovery enter:

```
QUERY ABARS
```

Determining the unit type for the control and data files

Before performing aggregate backup, you must determine the tape unit type to be allocated for the data, control, and instruction/activity log files. The unit type must be a supported tape device, as specified in the *z/OS DFSMSHsm Implementation and Customization Guide*. It is important that the unit type is compatible with the unit type that will be used at the aggregate recovery site.

You can specify the unit type in one of two ways:

1. Use the SETSYS ABARSUNITNAME(*unitname*) command. For *unitname*, substitute the name of a valid unit type.

If you use this method, you can issue ABACKUP and ARECOVER commands without using their UNIT parameters. When DFSMSHsm processes the ABARS control, data, and instruction/activity log files, it uses the unit type specified in the SETSYS ABARSUNITNAME command.

2. Specify the unit type later as part of the ABACKUP command. See [“ABACKUP command” on page 347](#).

Note: If you do not specify the unit type by using the SETSYS ABARSUNITNAME parameter or if you do not use the UNIT parameter with individual ABACKUP or ARECOVER commands, DFSMSHsm selects a default unit type of 3590-1.

Minimizing data set contention

The following are recommendations on how to minimize data set contention during aggregate backup. **If you do not follow these recommendations, aggregate backup may fail.**

- Do not perform ABACKUP processing during AUTOMIGRATION processing.
- Do not perform ABACKUP processing during AUTOBACKUP processing if you are processing VSAM data sets.
- Do not use aggregate backup while the application is in use.
- Use the concurrent copy feature of ABARS.

Performing ABACKUP processing concurrent with volume dumping or while backing up non-VSAM data sets is not a problem.

Data sets cannot be migrated or recalled during aggregate backup processing.

When ABACKUP processes data sets on level 0 volumes and the status of a data set changes (for example, it is moved or deleted) after catalog information about the list of data sets to process has been created (but before serialization), ABACKUP processing fails. The ABACKUP processing must be entirely rerun.

Data sets that DFSMSHsm dynamically allocates during aggregate backup processing

DFSMSHsm dynamically allocates the following data sets (SYSIN, FILTER, and authorization-checking) during aggregate backup processing. These data sets are used while aggregate backup is executing and are dynamically deallocated before completion of aggregate backup processing. These data sets are allocated with a unit type of SYSALLDA. (If they are non-SMS-managed, you must have at least one STORAGE- or PUBLIC-type device online to use the system.) You need to be aware of these data set names so that the high-level qualifiers used by ABARS can be defined to the system. You may want to modify your ACS filters to direct allocation of these data sets to specific volumes.

SYSIN data set

The SYSIN data set is used to pass DUMP parameters to DFSMSdss. The SYSIN data set name is: *authid.SYSIN.agname.Thhmmss.Dyyyyddd*

The variables in the data set name are explained:

authid

DFSMSHsm-authorized user ID.

agname

Aggregate group name that is processed.

hhmmss

Hours, minutes, and seconds.

yyyyddd

Four-digit year followed by the Julian day of the year.

FILTER data set

The FILTER data set is used to pass a list of data set names to DFSMSdss. The FILTER data set name is: *authid.FILTER.agname.Thhmmss.Dyyyyddd*

The variables in the data set name are explained:

authid

DFSMSHsm-authorized user ID.

agname

Aggregate group name that is processed.

hhmmss

Hours, minutes, and seconds.

yyyyddd

Four-digit year followed by the Julian day of the year.

Authorization-checking data set

The authorization-checking data set is used when checking RACF authorization for user tape data sets. The authorization-checking data set name is: *authid.Dyyddd.Thhmmss*

The variables in the data set name are explained:

authid

DFSMSHsm-authorized user ID.

yyddd

Year and Julian day.

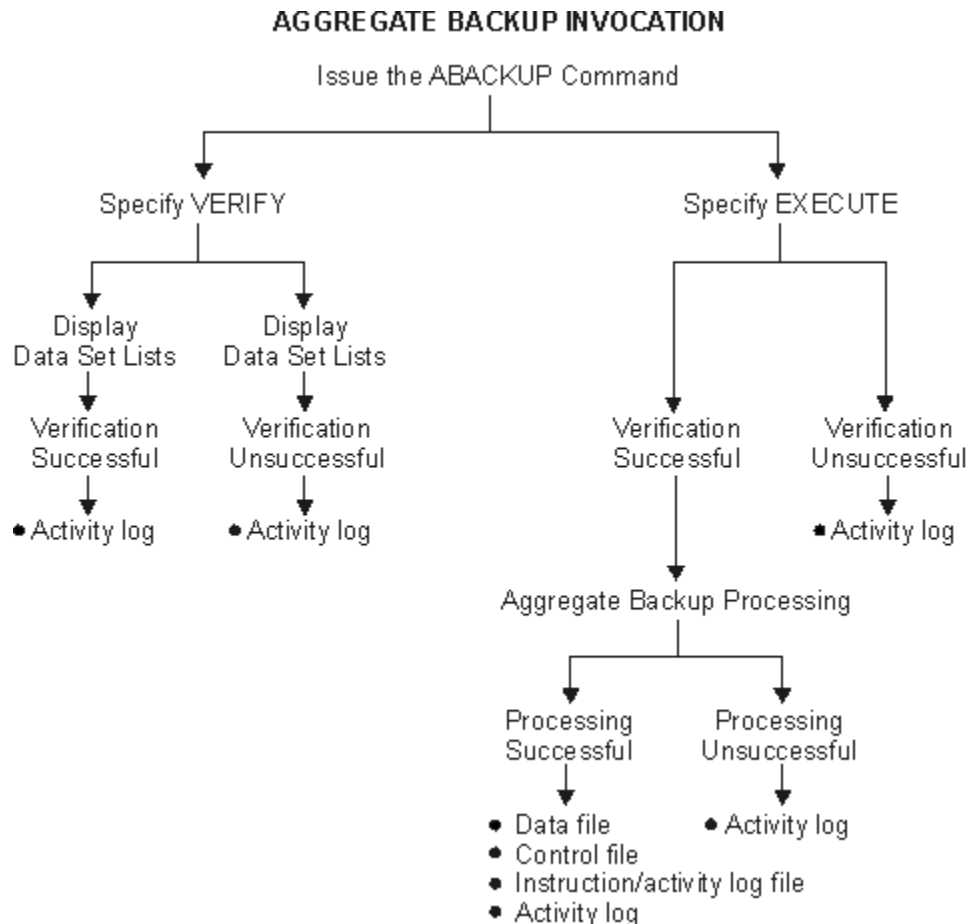
hhmmss

Hours, minutes, and seconds.

Performing aggregate backup

Figure 40 on page 347 shows what occurs when you invoke aggregate backup by issuing the ABACKUP command.

The following topics describe the aggregate backup invocation in more detail.



ARC15627

Figure 40. Aggregate Backup Invocation

ABACKUP command

To perform aggregate backup, enter the ABACKUP command. If you only want to verify the content of an aggregate backup without actually having the data sets backed up, specify the VERIFY parameter. This is like performing a test run of aggregate backup. After aggregate backup verification processing completes, the ABACKUP command stops. No data sets are backed up.

Note: If you have specified the FILTEROUTPUTDATASET parameter on the ABACKUP command, verification processing will place the results of the data set filtering into the data set specified. These results consist of a list of data sets selected for ABACKUP processing. This list can be useful when developing the data set filters in the selection data set.

When you are ready to have your data sets backed up, you can issue the ABACKUP command and specify the EXECUTE parameter. After successful data set verification processing, your data sets are backed up.

The following topics describe in more detail what occurs when you issue the ABACKUP command with the VERIFY or EXECUTE parameter.

The *aname* and EXECUTE | VERIFY parameters are required on the ABACKUP command.

For *aname*, specify the name of the aggregate group you defined earlier. See [“Defining an aggregate group”](#) on page 336.

The VERIFY parameter lists all input tape volumes that are required at the aggregate backup site if a subsequent ABACKUP EXECUTE command is submitted. It also performs verification processing of both migrated and nonmigrated data sets, detecting validation errors before data movement actually takes place.

For *unittype* in UNIT(*unittype*), specify the tape unit type to be allocated for the aggregate backup data, control, and instruction/activity log files.

After successful aggregate backup of an application, the optional MOVE parameter causes ABACKUP to delete all data sets in the application. The different data sets are affected in the following manner:

- DASD data sets are deleted
- Tape data sets are uncataloged
- ACCOMPANY data sets are uncataloged
- ALLOCATE data sets are deleted from DASD with the exception of user catalogs in the ALLOCATE list
- MIGRATED data sets are deleted

Note: If users with restricted RACF command authority issue the ABACKUP command with the MOVE parameter, they must have ALTER access authority for each data set belonging to the aggregate group, or aggregate backup verification fails with a related error message.

If a user has comprehensive RACF command authority and issues the ABACKUP command with the MOVE parameter, no RACF checking of individual data sets is required.

The optional FILTEROUTPUTDATASET(*dsname*) parameter allows DFSMSHsm to direct a fully-qualified list of data set names from the filtering criteria from the INCLUDE/EXCLUDE lists, as well as the names of ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE data sets, to a separate output data set. If the user has restricted ABACKUP command authority, then that user must have a minimum of update authority to append to the existing RACF-protected filter output data set or allocation authority to create a new filter output data set. For *dsname*, substitute a data set name of up to 44 characters. FILTEROUTPUTDATASET can be abbreviated FODS.

When the FILTEROUTPUTDATASET parameter is specified along with the PROCESSONLY parameter, the filter output data set contains the data sets that pass the filtering criteria and also match the category designated in the PROCESSONLY parameter.

When you use the optional OPTIMIZE(*option*) parameter with an ABACKUP command it overrides the value specified by the SETSYS ABARSOPTIMIZE command. The OPTIMIZE parameter allows you to specify the performance level used by DFSMSdss during dump processing of level 0 DASD data sets. You can specify that DFSMSdss read one, two, or five tracks at time, or one cylinder at a time.

The optional PROCESSONLY parameter directs DFSMSHsm to process only the specified categories. You can specify one or more of the categories L0, ML1, ML2, and USERTAPE. The categories can be specified in any order. Specification of all categories will be accepted and is equivalent to not specifying the PROCESSONLY parameter.

Restriction: During ABACKUP, the PROCESSONLY parameter is processed only for data sets that exist (cataloged) during the creation of the control file. If the data set is deleted between the time that data set filtering is complete and the time that there is a successful search (locate) for it during the building of the control file, there is no way to determine where the data set (L0, ML1, ML2, or USERTAPE) resides, nullifying any meaning the PROCESSONLY option would have had.

PROCESSONLY can be abbreviated to PONLY. The following are recognized aliases for category names:

Category	Aliases
----------	---------

L0	
-----------	--

	LEVEL0, PRIMARY
--	-----------------

ML1	
------------	--

	MIGRATIONLEVEL1
--	-----------------

ML2	
------------	--

	MIGRATIONLEVEL2
--	-----------------

The optional, and mutually exclusive, STACK and NOSTACK parameters allow you to override the specified SETSYS ABARSTAPES value, or its default value of STACK. The STACK option causes ABACKUP to place the ABACKUP output files on a minimum number of tape cartridges. The STACK option only applies to

output being directed to tape. Using the STACK option and attempting to direct the ABACKUP output files to DASD causes ABACKUP to fail.

The SKIP option causes DFSMSHsm to skip data sets on volumes that are protected by PPRC or XRC. That is, these data sets are not backed up.

You might find it helpful to use the following DFSMSHsm commands to manage and query an ABACKUP command:

- HOLD ABACKUP prevents all currently queued aggregate backup requests from being processed.
- HOLD ABACKUP(*agname*) prevents currently queued requests for a specific aggregate group from being processed.
- HOLD ABACKUP EOD prevents currently queued requests from being processed, and additionally stops processing of any currently running ABACKUP requests.
- HOLD ABACKUP(*agname*) EOD stops processing of the aggregate group specified, and prevents further requests for this aggregate group from being selected for processing.
- RELEASE ABACKUP releases all queued aggregate backup requests that have been held by using the HOLD ABACKUP command or by DFSMSHsm.
- RELEASE ABACKUP(*agname*) releases all queued requests for the specified aggregate group.
- STOP causes an orderly shutdown of DFSMSHsm after any currently running ABACKUP commands complete processing.
- STOP DUMP causes DFSMSHsm to end processing immediately and to provide a dump. Any ABACKUP command in process is canceled immediately.
- QUERY or HQUERY can display information relevant to ABACKUP.
- CANCEL or HCANCEL can cancel any existing queued ABACKUP.

Related reading

For more information about the commands listed in this topic, see [Part 2, “DFSMSHsm Storage Administration Reference,”](#) on page 677.

Examples of ABACKUP commands

In the example system, the ABACKUP command is first issued with the VERIFY parameter:

```
ABACKUP PAY1 VERIFY
```

Aggregate backup verification is performed in the example system aggregate group, PAY1. Remember, no data sets are actually backed up.

To invoke aggregate backup processing, the ABACKUP command is issued with the EXECUTE parameter:

```
ABACKUP PAY1 EXECUTE
```

Aggregate backup processing is performed in the example system aggregate group PAY1. The data sets are backed up.

Issuing the ABACKUP command with the VERIFY parameter

When you want only to verify the content of your aggregate backup without actually backing up the data sets, issue the ABACKUP command with the VERIFY parameter. This optional test run allows aggregate backup to identify any errors, and it allows you to determine if the data sets that are backed up are what you intended. Remember, no data sets are backed up.

When you specify the VERIFY parameter, aggregate backup automatically performs verification processing. Before it directs all messages to the aggregate backup activity log, verification processing checks to see if the following conditions are met:

- The aggregate group identified in the ABACKUP command is defined.
- The management class identified in the aggregate group is defined.
- The selection data sets named in the aggregate group are allocated and cataloged and meet format restrictions.
- The instruction data set, if named in the aggregate group, is allocated and cataloged.
- Selection data set syntax is correct. See [“Selection data set syntax”](#) on page 329.
- All data sets in the ALLOCATE list are DASD data sets that conform to the supported data set organization types. See [“ALLOCATE”](#) on page 333.
- All data sets in the ACCOMPANY list are tape data sets.
- No data sets in any list, other than the ALLOCATE list, are catalogs.
- All fully qualified data sets are cataloged.
- The same data set name cannot appear in the INCLUDE, ACCOMPANY, or ALLOCATE lists.
- Installation exits, if requested, can be loaded.
- If RACF facility class is active and the user has restricted authority, the user’s authority to process data sets is verified.

When processing completes, determine if the ABACKUP command completed successfully with a return code of zero. You may want to review the activity log for message ARC6166I for a list of the data sets selected for INCLUDE, EXCLUDE, ACCOMPANY, ACCOMPANYEXCLUDE, ALLOCATE, and ALLOCATEEXCLUDE processing. If the list of data sets is acceptable, reissue the ABACKUP command with the EXECUTE parameter so that your data sets will be backed up as described in the following topic. If the list of data sets is unacceptable, modify the selection data set entries or installation exits and reissue the ABACKUP command with the VERIFY parameter until your list of data sets is acceptable.

To aid you in reviewing the results of data set filtering, you can use the FILTEROUTPUTDATASET parameter during verification processing.

If the ABACKUP command failed, as indicated by a nonzero return code, review the aggregate backup activity log for error messages. Correct any errors and reissue the ABACKUP command with the VERIFY parameter. Repeat this process until the ABACKUP command completes successfully with a return code of zero.

Issuing the ABACKUP command with the EXECUTE parameter

Issue the ABACKUP command with the EXECUTE parameter when you are ready to have your data sets actually backed up. Aggregate backup always performs verification processing, as described in the previous topic. The difference is that after verification processing completes successfully, aggregate backup proceeds to back up your data sets.

Aggregate backup directs all messages to the aggregate backup activity log. The aggregate backup starting and ending messages are also listed in the command activity log and displayed to the user who issued the command. When processing completes, determine if the ABACKUP command completed successfully with a return code of zero. Review the activity log for messages ARC6060I and ARC6061I for the volume serial numbers of the aggregate backup control and data files, ARC6071I for the volume serial number for the instruction/activity log file, and ARC6062I for the volume serial numbers for the tapes containing ACCOMPANY data sets. You will need these volumes later when you perform aggregate recovery.

If the ABACKUP command failed, as indicated by a nonzero return code, review the activity log for error messages. Correct any errors and reissue the ABACKUP command. The command always starts processing from the beginning. Repeat this process until the ABACKUP command completes successfully with a return code of zero. If the command fails, the data on the tape is invalid.

Since aggregate backup allows backing up of data sets that have been migrated to ML2 tapes, it is possible that the tape volumes containing the data sets to be backed up reside on an ML2 tape volume that is in use by another DFSMSHsm function. See [“Tape migration level 2 volume contention”](#) on page

136 for more information. If ABACKUP determines that it cannot allocate a tape volume because it is “in use”, it retries the allocation up to 90 times, at 20 second intervals. This takes approximately 30 minutes.

If the preceding retry scenario has been completed and the ML2 tape is still “in-use,” DFSMSHsm issues message number ARC6254A to the operator by way of a write to operator with reply (WTOR). The operator then has the option of canceling the aggregate backup or retrying the allocation retry logic.

Aggregate backup will copy multiple migrated data sets from a specific ML2 tape with a single mount, except when a data set spans to another tape.

Aggregate backup output

This topic describes the data, control, and instruction/activity log files, the activity log created by aggregate backup, and the FSR and ABR records containing ABACKUP CPU times and accounting codes.

Data files

The DFSMSdss data file is an output data set tape file that contains copies of the backed up data sets. They are the data sets you specified to be backed up in the selection data sets.

The format of the data file name is *outputdatasetprefix.D.CnnVnnnn*. This is the same *outputdatasetprefix* you previously specified in the aggregate group.

The name of the DFSMSdss data file in the example is PAY1.D.C01V0001. It contains copies of the following data sets, based on the selection data set entries previously made:

- PAY2.PY002.SOCSEC
- PAY2.PY002.STATUS.G0002V00
- PAY1.PY001.CHECK

The name of the internal I/O data file in the example is PAY1.O.C01V0001. It contains copies of the following data sets, based on the selection data set entries previously made:

- PAY1.PY002.IRA
- PAY1.PY001.BONUS
- PAY1.PY001.LADDER
- PAY1.PY001.NAMES

The format of the data file name is *outputdatasetprefix.O.CnnVnnnn*. This is the same *outputdatasetprefix* you previously specified in the aggregate group.

If you need to direct your ABACKUP internal I/O data file for a unique copy number to DASD, this can be done only if your DFSMSdss data file for the same copy number is directed to DASD.

The "D" data file is created if your INCLUDE statement specifies data sets that reside on Level 0 DASD volumes. The "O" data file is created if your INCLUDE statement specifies user tape data sets or data sets migrated to ML1 or ML2 DASD, or ML2 tape. If you specify the STACK option, a "D" and "O" data file is always created, even when you do not provide an INCLUDE statement. If you omit both the STACK option and the INCLUDE statement, neither data file is created.

If you choose to have any or all of your ABACKUP output files directed to DASD volumes, there are special precautions you must observe for successful recovery of the backed up data:

- The aggregate backup files must exist and be properly cataloged at the recovery site.
- You must use the XMIT and DATASETNAME parameters on your ARECOVER command.

Control file

The control file is the output data set tape file created by aggregate backup that is used later as input to an aggregate recovery. It contains control information necessary to recover the data sets, defined to the aggregate.

Note: The control file can reside on 1 to 15 volumes. If the 15-volume limit is exceeded, aggregate backup fails. If this occurs, either use tapes with a higher density to contain the control file or define multiple aggregate groups for the application so that each aggregate backup does not back up such a large amount of data. Reissue the ABACKUP command.

The format of the control file name is *outputdatasetprefix.C.CnnVnnnn*. This is the same *outputdatasetprefix* you previously specified in the aggregate group.

The name of the control file in the example is PAY1.C.C01V0001. The data set PAY2.PY001.BENEFIT is identified on the control file to be allocated at the aggregate recovery site. The data set PAY2.PY001.TAXES is identified on the control file to be cataloged at the aggregate recovery site.

Instruction and activity log file

If you have previously specified an instruction data set name in the aggregate group definition, or if the ABACKUP activity log is directed to DASD by using SETSYS ABARSACTLOGTYPE(DASD), then the optional instruction/activity log file is written to a different tape volume than the control and data files. See [“Creating an instruction data set” on page 334](#).

The format of the instruction/activity log data set name is *outputdatasetprefix.I.CnnVnnnn*. This is the same *outputdatasetprefix* you previously specified in the aggregate group. See [“Defining an aggregate group” on page 336](#).

The name of the instruction/activity log file in the example is PAY1.I.C01V0001. It contains a copy of the instruction data set, PAYROLL.INSTRUCT, that was previously created.

Note: This data set would not be created unless INSTRUCTION DATASET is specified or the ABACKUP activity log is directed to DASD.

Activity log

An activity log is created each time you issue the ABACKUP command. You might find the activity log useful because it contains messages written during aggregate backup processing. Review messages ARC6060I and ARC6061I for a list of the volume serial numbers for the control and data files, ARC6071I for the volume serial number for the instruction/activity log file, and ARC6062I for the volume serial numbers for the tapes containing the ACCOMPANY data sets. These volumes and their serial numbers are required later as input to aggregate recovery.

The format of the activity log name is: *mvtactn.Hmcthost.ABACKUP.agname.Ddyddd.Thhmmss*

The parts of the activity log name are explained:

mvtactn

Activity log high-level qualifier. The default value is HSMACT.

Hmcthost

DFSMSHsm host identification.

ABACKUP

Constant for aggregate backup.

agname

Name of the aggregate group being backed up.

Ddyddd

The date.

Thhmmss

The time.

The activity log is written to either SYSOUT or a DASD data set; you determine this by using the ABARSACTLOGTYPE parameter of the SETSYS command. If you specify DASD and the allocation attempt fails, an attempt is made to allocate the activity log to SYSOUT. If the activity log is directed to DASD, it is backed up to the instruction/activity log file. See [“Defining SETSYS parameters for aggregate backup” on page 343](#) for more information on setting the activity log output type.

You can use the SETSYS ABARSDELETEACTIVITY(Y | N) command to specify whether you want DFSMSshm to automatically delete the ABARS activity log that corresponds to the ABACKUP version being rolled off. The deletion occurs during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing, and removes the need to manually manage the ABARS activity logs. The default is N, which specifies no automatic deletion.

Figure 41 on page 353 is an example of the aggregate backup activity log.

```

PAGE 0001 DFSMSHSM 1.9.0 DATA FACILITY HIERARCHICAL STORAGE MANAGER 05.350 12:24
ARC6054I AGGREGATE BACKUP STARTING FOR AGGREGATE GROUP PAY1, AT 12:24:33, STARTED TASK = DFHSMABR.ABAR0122
ARC6030I ACTIVITY LOG FOR AGGREGATE GROUP PAY1 WILL BE ROUTED TO SYSOUT=A
ARC6152I THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM THE ALLOCATE LIST DURING AGGREGATE
BACKUP FOR AGGREGATE GROUP PAY1
PAY2.PY001.BENEFIT
ARC6152I THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED FROM THE ACCOMPANY LIST DURING AGGREGATE
BACKUP FOR AGGREGATE GROUP PAY1
PAY2.PY001.TAXES
ARC6004I 0001 ABACKUP PAGE 0001 5695-DF175 DFSMSdss Data Set Services 97.238 12:34
ARC6004I 0001 ABACKUP ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC6004I 0001 ABACKUP DUMP DATASET(FILTERDD(SYS00005)) -
ARC6004I 0001 ABACKUP OUTDDNAME(SYS00004) -
ARC6004I 0001 ABACKUP ) OPTIMIZE(3) SPHERE -
ARC6004I 0001 ABACKUP ALLDATA(*) -
ARC6004I 0001 ABACKUP SHARE
ARC6004I 0001 ABACKUP ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ARC6004I 0001 ABACKUP ADR109I RI01 (01), 97238 12:34:04 INITIAL SCAN OF USER CONTROL STATEMENTS
COMPLETED.
ARC6004I 0001 ABACKUP ADR006I (001)-SETUP(01), 97238 12:34:05 EXECUTION BEGINS
ARC6004I 0001 ABACKUP ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC6004I 0001 ABACKUP PAY2.PY002.SOCSEC
ARC6004I 0001 ABACKUP PAY2.PY002.STATUS.G0002V00
ARC6004I 0001 ABACKUP PAY1.INSTRUCT
ARC6004I 0001 ABACKUP PAY1.PY001.CHECK
ARC6004I 0001 ABACKUP PAY1.PY001.PGMLIB
ARC6004I 0001 ABACKUP ADR013I (001)-CLTSK(01), 97238 12:34:13 TASK COMPLETED WITH RETURN CODE 0000
ARC6004I 0001 ABACKUP ADR012I (SCH)-DSSU (01), 97238 12:34:13 DFSMSdss PROCESSING COMPLETE.
HIGHEST RETURN CODE IS 0000
ARC6064I DATA SET PAY1.PY001.NAMES HAS BEEN SUCCESSFULLY BACKED UP DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP PAY1
ARC6064I DATA SET PAY2.PY002.RETIRE HAS BEEN SUCCESSFULLY BACKED UP DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP PAY1
ARC6064I DATA SET PAY1.PY002.IRA HAS BEEN SUCCESSFULLY BACKED UP DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP PAY1
ARC6064I DATA SET PAY1.PY002.LADDER HAS BEEN SUCCESSFULLY BACKED UP DURING AGGREGATE BACKUP FOR
AGGREGATE GROUP PAY1
ARC6369I STORAGE REQUIREMENTS FOR AGGREGATE GROUP PAY1 ARE: L0=1105K, ML1=98K, ML2=0, TOTAL=1203K
ARC6061I VOLUMES USED FOR CONTROL FILE PAY1.C.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
A00009
ARC6060I VOLUMES USED FOR DATA FILE PAY1.D.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
A00008
ARC6060I VOLUMES USED FOR DATA FILE PAY1.O.C01V0001 DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
A00008
ARC6062I VOLUMES USED FOR ACCOMPANY DATA SETS DURING AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 ARE:
A00001
ARC6051I AN INSTRUCTION DATA SET WAS NOT SPECIFIED FOR AGGREGATE GROUP PAY1
ARC6055I AGGREGATE BACKUP HAS COMPLETED FOR AGGREGATE GROUP PAY1, AT 12:36:53, RETCODE = 000

```

Figure 41. Example of an Aggregate Backup Activity Log

ABACKUP CPU time and accounting codes

ABARS records the CPU time for processing ABACKUP and ARECOVER requests. This information is written to the FSR control block. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*). ABARS then writes the FSR as *smfid* + 1 in the SYS1.MANx or SYS1.MANy system data sets.

The ABACKUP CPU time and ARECOVER CPU time are also recorded in the ABR record.

The ISMF aggregate group definition panel allows you to specify an accounting code for an aggregate group. The accounting code gets written to the FSR control block (and associated SMF record where one exists) along with the ABACKUP CPU processing time for the aggregate. Installations can then use the information to assist in calculating charges to users for ABARS requests.

The accounting code is also written to the ABR record, and the ABACKUP control file. It is written to the ABACKUP control file so the code can be used at the recovery site without requiring an aggregate definition at the recovery site.

The accounting code is optional, and can be up to 32 characters in length. Valid characters are alphanumeric characters, blanks, commas, periods, parentheses, hyphens, slashes, ampersands, and apostrophes (single quotes).

DFSMSHsm coexistence considerations for new ABACKUP output data sets

In DFSMSHsm, the ABACKUP output files are physical sequential data sets, whereas in previous versions of DFSMSHsm these files were generation data sets. This change allows DFSMSHsm to fully manage the ABACKUP output files by using the management class aggregate backup attributes.

Once your installation has converted to DFSMSHsm and is processing aggregate backups, new GDSs are not created for existing ABACKUP output file GDGs. This means that GDG roll-off processing does not take place for the old ABACKUP output file GDSs, and these GDSs and their associated GDG base definitions need to be manually deleted once you have determined that they are no longer needed. You can do this by using an IDCAMS batch job with a DELETE statement for each remaining GDS and a DELETE GDG statement for each remaining GDG base.

Note: DFSMSHsm will not consider old GDSs and GDG bases during roll-off processing for old aggregate backup versions and EXPIREBV ABARSVERSIONS processing.

Expiring ABR records in the BCDS

You can use the EXPIREBV ABARSVERSIONS command to expire ABR records created during aggregate backup or recovery processing.

The ABARSVERSIONS keyword is mutually exclusive from any data set processing performed during EXPIREBV processing. ABARSVERSIONS is also mutually exclusive with the NONSMSVERSIONS parameter. The STARTKEY and ENDKEY parameters can be used with ABARSVERSIONS, as can OUTDATASET | SYSOUT. DISPLAY and EXECUTE are still required parameters and have the same processing affect. .

SETSYS BACKUP | NOBACKUP has no relevance with the ABARSVERSIONS parameter. However, a BCDS must exist or the EXPIREBV ABARSVERSIONS command fails.

When ABARSVERSIONS has been specified, the BCDS is searched and each ABR record is checked to see if the ABARS version has expired. For expired ABARS versions, the control file name is located in the catalog and a list of volume serial numbers that the control file resides on is requested. When necessary, the volumes are deleted from the RACF HSMABR tape volume list. The control file is uncataloged after the volume serial number list has been exhausted. When the file does not exist in the catalog, it is not an error condition, and the ABR record can still be deleted.

EXPIREBV must obtain the management class definition to determine which ABACKUP copies have expired. The management class definition contains the following:

- RETAIN EXTRA VERSIONS—This value is used to expire all versions except the latest version.
- RETAIN ONLY VERSION—This value is used to expire the last remaining version.

If, by using the ARCEDEXT installation exit, you have specified special tape management system expiration dates for the ABACKUP output files, the preceding two values should be set to NOLIMIT. This prevents DFSMSHsm from attempting to delete ABARS versions during ABACKUP and EXPIREBV processing by using management class criteria.

Note: The preceding two values have total control over deletion of ABARS versions, unless the EXPIREBV command is issued with the RETAINVERSIONS keyword.

If the management class specified in the most current version of the ABR record for a specific aggregate group does not exist, expiration of the aggregate group being processed is skipped. If an aggregate group name has been specified on the EXPIREBV command, the command fails for this case.

If it is active, installation exit ARCTVEXT is called for each volume being processed, indicating to the exit that the volume being expired is an ABARS tape volume.

This process is repeated for the data files and the instruction/activity log files.

After the control file, data files, and instruction/activity log file have been uncataloged and removed from the HSMABR tape connected set, the ABR record is deleted and processing continues with the next ABR record available in the BCDS.

Note: It is possible to code SMS ACS routines that will direct ABARS files to DASD. If this is done, and the files are migrated to DASD, an DFSMSHsm DELETE will be issued to remove them during expiration processing. If that delete fails, the file may remain cataloged with no ABR record pointing to it. If this happens, the data set should be manually deleted. See [“Directing ABARS files to DASD” on page 434](#) for more information.

When you use EXECUTE with the ABARSVERSIONS parameter, message ARC6377I is issued to show the status of each file in the ABR record. Message ARC6376I is also issued to show whether the ABR record was deleted.

When you use DISPLAY with the ABARSVERSIONS parameter, a report is generated that lists information about the ABR records that would have been omitted had EXECUTE been specified. The file status shows whether the file will be uncataloged when EXECUTE is run, or whether the file does not exist in the catalog (“Not in Catalog”). If “Catalog Error” is in the file status, an error has occurred when DFSMSHsm retrieved the catalog information, and the ABR record cannot be deleted.

A sample of the output obtained when the ABARSVERSIONS parameter is used with DISPLAY is shown in [Figure 42 on page 355](#).

```
DISPLAY OF ABARS VERSIONS ELIGIBLE FOR EXPIRATION AT 08:36:24 ON 1995/02/07 FOR XA2
```

```
COMMAND INPUT: ABARSVERSIONS
SUBMITTER'S USERID: B999999
AGNAME = agname1    VERSION = 0001

AGGREGATE VERSION KEY = agname1.1995102000101    DELETED*
(* DETERMINED ON 1995/02/07)
AGGREGATE GROUP FILES      FILE STATUS
  prefix.C.C01V0001        Uncataloged
  prefix.D.C01V0001        Uncataloged
  prefix.O.C01V0001        Not in Catalog
  prefix.I.C01V0001        Uncataloged

AGGREGATE VERSION KEY = agname1.1995102000102    NOT DELETED*
(* DETERMINED ON 1995/02/07)
AGGREGATE GROUP FILES      FILE STATUS
  prefix.C.C02V0001        Uncataloged
  prefix.D.C02V0001        Not in Catalog
  prefix.O.C02V0001        Uncataloged
  prefix.I.C02V0001        Catalog Error

AGNAME = agname2    VERSION = 0003    EXP DATE = 1995/05/17

AGGREGATE VERSION KEY = agname2.1995137000301    DELETED*
(* DETERMINED ON 1995/02/07)
AGGREGATE GROUP FILES      FILE STATUS
  prefix.C.C01V0003        Not in Catalog
  prefix.O.C01V0003        Uncataloged
END OF DISPLAY - 00000002 ABARS VERSIONS ELIGIBLE FOR EXPIRATION
```

Figure 42. Example of DISPLAY ABARSVERSIONS Parameter Output

Related reading

For more information about the EXPIREBV command, see Chapter 44, [“EXPIREBV command: Deleting unwanted backup versions of data sets and UNIX files,” on page 851](#).

ABACKUP considerations

If an instruction data set is specified in the aggregate group and DASD logging is specified, DFSMSdss is used to write the instruction data set and the activity log data set into one file on a tape (or tapes). To allow the control file to save the instruction and activity log volume serial number information, the instruction data set and activity log files must be written before the control file. Because of this restriction, the activity log written to tape does not contain volume serial number information for the control file, nor the completion code from ABACKUP processing.

In order to facilitate coexistence with previous releases of ABARS, where the control file is the first file sequence on the tape, the instruction and activity log files are written to a separate tape from the control file, if an instruction data set is specified in the aggregate group or DASD logging is specified.

The log is written to an output tape *only* if:

- The data file has been successfully written to tape.
- DASD logging has been specified at the aggregate backup site using the SETSYS ABARSACTLOGTYPE command, and the DASD activity log has been successfully processed.

If DASD logging is not specified during ABACKUP processing, the following processing takes place:

- A message is written to the log indicating that DASD has not been specified for the ABACKUP process.
- A blank activity log name is written to the ABR record and control file header.

The file containing the activity log data set and instruction data set has a naming convention of:
prefix.I.CnnVnnnn

The parts of the activity log data set and instruction data set name are explained:

prefix

Prefix specified on the Aggregate Group Define/Alter panel.

I

Constant to identify the file as containing the instruction data set and the activity log.

Cnn

The copy number specified for ABACKUP.

Vnnnn

Version number.

If the ABACKUP activity log is being written to SYSOUT and an instruction data set has not been specified in the aggregate group definition, DFSMSdss is not called to DUMP the data sets (there will *not* be a tape mount request for the instruction or activity files).

After the data files are written to tape, the ABACKUP activity log (DASD only) is closed and the instruction data set and activity log are DUMPed to tape using DFSMSdss. All messages are intercepted and stored in memory during DFSMSdss processing.

After the instruction and activity log files are written to tape, the DASD ABACKUP activity log is reopened to allow the stored messages—and any additional messages—to be written to it.

Note: These additional messages are not written to the activity log file that is seen at the ARECOVER site.

The internal I/O data file for a unique copy number can be directed to DASD only if the DFSMSdss data file for the same copy number was directed to DASD, or there is no data file for this invocation of ABACKUP.

If you choose to have any or all of your ABACKUP output files directed to DASD volumes, there are special precautions you must observe for successful recovery of the backed up data:

- The aggregate backup files must be properly cataloged and exist at the recovery site.
- You must use the XMIT and DATASETNAME parameters on your ARECOVER command.

ABARS user catalog and catalog alias ABACKUP support

When DFSMSShsm determines that there are user catalogs in the ALLOCATE list, it provides aggregate backup and recovery support for the catalogs and their associated ALIAS definitions.

When a user catalog is specified in the ALLOCATE list, the necessary data is obtained to define that catalog at the recovery location. The LISTCAT function also provides a list of the aliases defined to the specific catalog.

There is no limit to the number of user catalogs that the user may specify in the ALLOCATE list. If the user specifies any catalog type other than an user catalog, an error message is issued and ABACKUP verification fails.

If a catalog data set name is encountered in either the INCLUDE or ACCOMPANY lists, an error message is issued and ABACKUP verification fails.

The aggregate backup verification step lists the user catalogs specified in the ALLOCATE list and all of the aliases defined to those catalogs. This action is performed when the user specifies the ABACKUP command with the VERIFY parameter.

Note: All errors encountered when processing catalogs in the ALLOCATE list result in error messages.

Aggregate backup assist

Aggregate backup assist (ABA) is a stand-alone program that supports the DFSMSHsm ABARS function. It supports ABARS by producing a selection data set that can be used as input to the aggregate backup function.

How aggregate backup assist works

When supplied with a list of job names associated with an aggregate group (that is, a group of applications, an application, or a subset of an application) the ABA program scans system management facilities (SMF) records that have been collected over a period of time. ABA extracts the names of data sets that have been opened by the jobs on the supplied list. ABA then produces a list of those data set names in the form of a selection data set.

Note: The input SMF records must be collected for a sufficient period of time to capture the processing of all of the applications in the aggregate group.

To shorten ABA's run time, the IFASMFDP utility program may be used before ABA is run to extract the necessary types of SMF records from an installation's SMF data set. See ["Input requirements" on page 358](#).

What aggregate backup assist produces

ABA produces a selection data set consisting of one ABARS INCLUDE statement specifying all of the data set names found during analysis of the SMF records. The data set names are specified in three sections delimited by block comments: cataloged DASD data sets, cataloged tape data sets, and uncataloged data sets. The data set names in each section are sorted in alphabetical order.

Recommendation: Before you issue the ABACKUP command that uses the selection data set that has been created, you should edit the selection data set to delete the names of uncataloged data sets.

Special considerations for ABARS

ABA has been designed to assist you in identifying data sets related to an aggregate group. You need to ensure that all data sets necessary to rebuild the working environment are identified and collected during the aggregate backup process.

Note: Although the output selection data set is in a form that may be usable as input to the aggregate backup function, you need to examine closely and modify the selection data set, if necessary, before actually using it in an ABACKUP command. If you use the data set created by ABA as input to aggregate backup, the data set name must be the same as the selection data set in the aggregate group.

When verifying the contents of the ABA-generated selection data set, look for problems such as the data set:

- Not containing all necessary data set names associated with the aggregate group (data sets not allocated or not opened during the SMF interval)
- Containing data set names that are not part of the aggregate group
- Containing data set names that no longer exist (flagged as uncataloged)
- Containing tape data set names that can be moved to an ACCOMPANY list
- Containing DASD data set names that can be moved to an ALLOCATE list

The following hints assist you in determining which data sets to back up and how to collect them.

Uncataloged data sets: ABARS does not support uncataloged data sets. If you want to back up uncataloged data sets, you should manually catalog the data sets before issuing the ABACKUP command. Any uncataloged data sets you do not choose to catalog should be removed from the ABA-generated selection data set.

Nonaggregate group data sets: The generated selection data set may contain the names of data sets not in the aggregate group; for example, SYS1.LINKLIB. When these data sets are to be excluded, you must delete their names from the generated selection data set.

DASD data sets: The contents of some DASD data sets may not be required for execution of an application. Data set names that are used only for output should be moved to an ALLOCATE list.

Tape data sets: Tape data sets are candidates for the ACCOMPANY list in the selection data set. If you do not want the burden of copying the tapes, the tape data set names should be moved to the ACCOMPANY list.

Authorization and security requirements

For each data set eligible for the selection data set, ABA checks whether the data set is cataloged, and attempts to obtain certain catalog information. ABA should be installed in an APF-authorized library for the following reasons:

- If any of the catalogs contain password protection, ABA may cause the MVS catalog to prompt the system operator for the password. Having APF authorization allows ABA to bypass MVS catalog security checking.
- APF authorization enables ABA to query catalogs that are RACF protected and prevents ABENDs caused by RACF authorization failures.

Input requirements

The ABA program uses two sources of input: SMF data and a list of job names associated with an aggregate group.

SMF data

Before running ABA, you need to ensure that your system is set up to have SMF collect record types 14, 15, and 64. Furthermore, the recorded data must contain sufficient records to span the time frame wherein all desired jobs have been processed and recorded in the SMF data sets. An aggregate group may contain applications that run only monthly, quarterly, or annually, so that you need to plan in advance to collect sufficient SMF data.

The following SMF records are required for ABA:

Type 14

INPUT or RDBACK data set activity

Type 15

OUTPUT/UPDATE/INOUT/OUTIN data set activity

Type 64

VSAM component or cluster status

Aggregate job names

You need to identify all of the jobs that make up the applications in the aggregate group. If your installation has formal naming conventions for job names, then generic job names may be supplied. Substitute an asterisk (*) for a string, and substitute a percent sign (%) for a character. [Table 36 on page 358](#) shows acceptable forms of generic job names.

Table 36. Acceptable Generic Job Names

Specify	ABA collects data set names for:
*AB	Any job name that ends with the string AB.
A*	Any job name that starts with the letter A.
PAY	Any job name with the string PAY in it.

Table 36. Acceptable Generic Job Names (continued)

Specify	ABA collects data set names for:
NET*A	Any job name that begins with NET and ends with A.
NET%%A	Any job name that begins with NET and ends with A with two characters in between. % can be used in place of any character.

- The job names, generic or not, should be one to eight characters long.
- Each job name must be specified on a separate record in the input stream.
- A job name can occur anywhere in the record between columns 1 and 80, inclusive.
- The first non-blank string in each record is assumed to be the job name, and any other information in the record is ignored.

Storage requirements

The amount of storage required to run ABA depends on the number of input job names and generic job names specified, and the number of data set references found to be eligible for the selection data set. ABA has been designed to run in environments with the 24-bit residency mode (RMODE=24) and an addressing mode of ANY (AMODE=31). This enables ABA to take advantage of storage above the 16-megabyte line. To ensure that such storage is available to ABA, especially when you are processing large applications with many jobs and data sets, you may want to specify a large region size on the EXEC statement in the JCL which executes ABA.

DD statements used by ABA

Table 37 on page 359 shows the JCL DD statements used by ABA. If your installation specified DYNALOC=NO in the ICEMAC macro for DFSORT, the SORTWORK DD statements need to be specified in the JCL which executes ABA. DYNALOC=NO tells DFSORT not to dynamically allocate sort work data sets.

Table 37. DD Statements Used by ABA

DD name	Purpose
SMFIN	Defines the input SMF data set.
SORTIN	LRECL=80 and RECFM=FB defines an intermediate data set to be used as input to DFSORT. A temporary data set should be used.
SELECTDS	LRECL=80 and RECFM=F or FB defines the final output selection data set where the data set names are written. It can be either a sequential data set or a member of a partitioned data set. The record length must be 80 and the record format must be fixed or fixed block.
SYSPRINT	LRECL=133 and RECFM=FBA defines a sequential message data set.
SYSOUT	Defines a sequential message data set required by DFSORT.
SORTWK01	Specifies that SYSDA space of 2 cylinders and secondary extents of 3 cylinders each can be used for workspace by DFSORT.
SYSIN	Defines the input data set containing a list of job names in the aggregate group. This data can be in the job stream.
SYSDIAG	Defines an optional diagnostic data set. The DCB parameters are RECFM=FB and LRECL=74. The BLKSIZE must be supplied by the user. The data written to this file is binary and should not be directed to spool. Note: Output from the SYSDIAG data set may be required by IBM Customer Support to diagnose problems with this utility. Therefore, it is highly recommended that you always include this DD statement.

Sample JCL

```
//ABA          JOB ????
//ABARUN      EXEC PGM=ABA,REGION=20M
//SMFIN       DD DSN=CUSTOMER.SMF.DATA,DISP=SHR
//SORTIN      DD DSN=SSORTIN,DISP=(NEW,DELETE),
//              SPACE=(CYL,(2,3)),
//              DCB=(LRECL=80,RECFM=FB),
//              UNIT=SYSDA
//SELECTDS    DD DSN=CUSTOMER.SELECT.DS,DISP=(NEW,CATLG),
//              DCB=(LRECL=80,RECFM=FB),
//              SPACE=(CYL,(1,1),RLSE),UNIT=SYSDA
//SYSPRINT    DD SYSOUT=*
//SYSOUT      DD SYSOUT=*
//SORTWK01    DD UNIT=SYSDA,SPACE=(CYL,(2,3))
//SYSIN       DD *
PAYA*
PAYB*
/*
//SYSDIAG     DD DSN=CUSTOMER.ABA.LOG,DISP=(NEW,CATLG),
//              DCB=(RECFM=FB,LRECL=74,BLKSIZE=7400),
//              SPACE=(CYL,(2,1),RLSE),UNIT=SYSDA
/* ON A 3380, THE PRIMARY ALLOCATION OF 2 CYLINDERS
/* FOR SYSDIAG WILL HANDLE THE DIAGNOSTIC OUTPUT
/* FOR APPROXIMATELY 18,000 UNIQUE, QUALIFIED DATA
/* SETS.
```

Sample output

```
INCLUDE(
/* -----*/
/* Following are cataloged nontape data sets */
/* -----*/
    COLE.MYDATA
    CLR.PLI.LOAD
    CLR.PLI.OBJ
    NET.NETVIEW.REL2.BNJPNL1
    ORDERS.IDS
    RINEY.DFSMS.ABC
/*      6   cataloged nontape data sets found */
/* -----*/
/* Following are cataloged tape data sets */
/* -----*/
    CUSTOMER.CMSBASE.LINKLIB /* VOLSER=TAPE01 */
    PAY0001.MONTHLY.OCT.DATA /* VOLSER=PAYS02 */
    PAY0001.WEEKLY.W0005.DATA /* VOLSER=PAYS02 */
    SYS1.BCDS.BACKUP.V0000007 /* VOLSER=TAPEXX */
/*      4   cataloged tape data sets found */
/* -----*/
/* Following data sets were not found in the */
/* normal catalog search */
/* -----*/
    CUSTOMER.TEMPORARY.LIST
    PAY0001.MONTHLY.OCT.REPORT
    PAY0001.WEEKLY.W0005.REPORT
/*      3   uncataloged data sets found */
/* )
```

Aggregate stacking procedure

Current design of the aggregate backup function allows you to use the STACK parameter on the ABACKUP command to backup only one aggregate to a minimum of one tape. With the restriction of one aggregate backup per tape, performing aggregate backup to a high capacity tape might leave much of the tape empty. The aggregate stacking procedure describes how you can take existing aggregate backup output files, and utilize MVS utilities to stack multiple aggregates or multiple versions of the same aggregate onto a single tape, and then how to recover any of the stacked aggregates from the tape. This topic includes two sample JCL jobs, that you can use as templates to copy (stack) a second aggregate backup onto the same tape as the first version of the aggregate backup, and to restore an aggregate backup onto its original tape after a failure during the stacking procedure. You can modify this template to stack more than two aggregates onto a tape by duplicating the 4 steps and then modifying the data set name (ABACKUP output file) and the LABEL number. This procedure also describes how to:

- prevent roll-off of a backup version on the tape.
- prevent tape expiration.

- protect the stacked tape with RACF.
- handle end of volume (EOV) error during the stacking process.
- recover backup files after an error during the stacking process.

Warning: The steps documented in this procedure are intended to be used as an example of how to stack multiple aggregates or multiple versions of the same aggregate onto the same tape. This is a process that occurs outside of DFSMSHsm's knowledge and control. As DFSMSHsm will not be aware that a tape may contain multiple aggregates or multiple versions of the same aggregate, precautions need to be taken during roll-off and EXPIREBV ABARSVERSIONS processing to ensure that tapes are not returned to scratch that still contain valid aggregate backups on them. See [“Preventing aggregate roll off and tape expiration” on page 365](#) and [“Deleting specific versions of aggregates from a stacked tape” on page 365](#) for a description of these precautions.

Backing up data with ABARS aggregate backup function

To create an aggregate backup, use the STACK keyword on the ABACKUP command. If you specify the STACK keyword, ABARS places all of the output files from the aggregate backup onto one tape. You can then use the ABACKUP command to stack multiple aggregates or multiple versions of the same aggregate onto the same tape.

The example described in this topic shows how to create two versions of the aggregate named AGM1, and then stack them onto one tape. The serial numbers of the tapes used for this purpose are A00040 and A00041.

Step 1: create the first aggregate backup.

- Specify the STACK keyword on the ABACKUP command. The resulting ABACKUP Version 1 files reside on tape volser A00040.

Table 38. The sequence of files associated with ABACKUP Version 1 on the tape volser A00040.

ABACKUP Version 1	File Sequence on tape volser A00040
Data (D)	1
Other Data (O)	2
Instruction and Activity Log (I)	3
Control (C)	4

Step 2: create the second aggregate backup.

- Specify the STACK keyword on the ABACKUP command. The resulting ABACKUP Version 2 files reside on tape volser A00041.

Table 39. The sequence of files associated with ABACKUP Version 2 on the tape volser A00041.

ABACKUP Version 2	File Sequence on tape volser A00041
Data (D)	1
Other Data (O)	2
Instruction and Activity Log (I)	3
Control (C)	4

Note: The output files might span to multiple tape volumes, if the aggregate output files are larger than the tape capacity.

Related reading

For more information about the aggregate output files, see [“Aggregate backup output” on page 351](#).

Stacking multiple aggregates on one tape

To stack multiple aggregates onto one tape, copy the files associated with the second backup onto the tape containing the files from the first backup. You can use the sample JCL provided in this topic, to stack additional aggregate backup copies onto the tape by copying the 4 (JCL) steps, and then modifying the ABARS output file name, the volser, and the tape file sequence number.

Step 3: copy ABACKUP Version 2 files onto the tape containing ABACKUP Version 1 files.

Table 40. File sequence after ABACKUP 2 files are copied onto the tape containing ABACKUP 1 files. Tape's volser is A00040.

File Sequence on tape volser A00040	
ABACKUP Version 1:	
Data (D)	1
Other Data (O)	2
Instruction and Activity Log (I)	3
Control (C)	4
ABACKUP Version 2:	
Data (D)	5
Other Data (O)	6
Instruction and Activity Log (I)	7
Control (C)	8

The sample JCL provided in this example, COPYAGGR JCL, stacks the files resulting from the second aggregate backup (ABACKUP Version 2) from the tape volser A00041 onto the tape volser A00040 which contains the files resulting from the first aggregate backup (ABACKUP Version 1). COPYAGGR JCL uses the IEBGENER program to copy the files . IEBGENER is used to copy the aggregate backup files from ABACKUP Version 2 onto the tape containing the aggregate backup files associated with ABACKUP Version 1, uncataloging the original ABACKUP Version 2 files and cataloging the newly copied versions of ABACKUP Version 2. To use COPYAGGR JCL in your environment, modify the aggregate backup file names, volsers, and labeling information that designates the location of the backup files on the tape (presented here in bold text).


```

//COPYAGGR JOB ,CLASS=A,USER=userid,PASSWORD=pswd,
// MSGCLASS=A
//*****
//*
//* Job Name: COPYAGGR
//*
//* Function: STACK aggregate backup files onto a single tape.
//* *
//* Description of Process: This is a sample JCL batch job
//* that copies the aggregate backup output files from the
//* second ABACKUP of an aggregate named AGM1 onto the tape
//* where the files from the first ABACKUP of aggregate AGM1
//* reside, thus stacking both aggregate backups onto a single
//* tape.
//*
//* STEP 1, in this job, copies the D file, STEP 2 copies the
//* O file. STEP 3 copies the I file and STEP 4 copies the C
//* file. You will have to update the data set names (ABACKUP
//* files) and the tape volsters in each step in order to run this
//* job in your environment.
//*
//* To copy additional aggregate backups onto the stacked tape,
//* duplicate these 4 steps for each additional aggregate
//* backup and then update the ABACKUP output file (D, O, I or
//* C) names and the file sequence numbers.
//*
//* The job will copy the files from the original ABACKUP tape
//* onto the stacked tape, uncatalog the files from the original
//* tape and catalog the copied version onto the stacked tape.
//*
//* In order to recover any of the aggregates from the stacked
//* tape, use the ABARS ARECOVER command with the XMIT and
//* NOSTACK options specified.
//*
//* Example of ARECOVER command:
//*
//* ARECOVER DSN(control-file-name) XMIT NOSTACK EXECUTE
//*
//*****
//*
//*****
//* Copy the D file from the 2nd ABACKUP of aggregate AGM1 onto
//* the same tape containing the output files from the 1st
//* ABACKUP of aggregate AGM1. This D file is copied at file
//* sequence number 5. If the copy is successful uncatalog the
//* original version of the D file and catalog the new stacked
//* version.
//*****

```

Figure 43. Stacking Aggregates - Sample JCL (Part 1 of 2)


```

//-----
//STEP1 EXEC PGM=IEBGENER
//-----
//*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=AGM1.D.C01V0002,DISP=(OLD,UNCATLG,KEEP)
//SYSUT2 DD DSN=AGM1.D.C01V0002,DISP=(NEW,CATLG,DELETE),
// UNIT=3480,LABEL=(5,SL),VOL=SER=A00040,
// DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)
//SYSIN DD DUMMY
//*
//-----
//*****
//* Copy the 0 file from the 2nd ABACKUP of aggregate AGM1 onto
//* the same tape containing the output files from the 1st
//* ABACKUP of aggregate AGM1. This 0 file is copied at file
//* sequence number 6. If the copy is successful uncatlog the
//* original version of the 0 file and catalog the new stacked
//* version.
//*****
//-----
//STEP2 EXEC PGM=IEBGENER
//-----
//*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=AGM1.O.C01V0002,DISP=(OLD,UNCATLG,KEEP)
//SYSUT2 DD DSN=AGM1.O.C01V0002,DISP=(NEW,CATLG,DELETE),
// UNIT=3480,LABEL=(6,SL),VOL=SER=A00040,
// DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)
//SYSIN DD DUMMY
//*
//-----
//*****
//* Copy the I file from the 2nd ABACKUP of aggregate AGM1 onto
//* the same tape containing the output files from the 1st
//* ABACKUP of aggregate AGM1. This I file is copied at file
//* sequence number 7. If the copy is successful uncatlog the
//* original version of the I file and catalog the new stacked
//* version.
//*****
//-----
//STEP3 EXEC PGM=IEBGENER
//-----
//*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=AGM1.I.C01V0002,DISP=(OLD,UNCATLG,KEEP)
//SYSUT2 DD DSN=AGM1.I.C01V0002,DISP=(NEW,CATLG,DELETE),
// UNIT=3480,LABEL=(7,SL),VOL=SER=A00040,
// DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)
//SYSIN DD DUMMY
//*
//-----
//*****
//* Copy the C file from the 2nd ABACKUP of aggregate AGM1 onto
//* the same tape containing the output files from the 1st
//* ABACKUP of aggregate AGM1. This C file is copied at file
//* sequence number 8. If the copy is successful uncatlog the
//* original version of the D file and catalog the new stacked
//* version.
//*****
//-----
//STEP4 EXEC PGM=IEBGENER
//-----
//*
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=AGM1.C.C01V0002,DISP=(OLD,UNCATLG,KEEP)
//SYSUT2 DD DSN=AGM1.C.C01V0002,DISP=(NEW,CATLG,DELETE),
// UNIT=3480,LABEL=(8,SL),VOL=SER=A00040,
// DCB=(RECFM=U,LRECL=0,BLKSIZE=32760)
//SYSIN DD DUMMY
//-----
//*****
//* To stack additional versions of the aggregate onto this tape,
//* copy the 4 steps that make up this JCL batch job, then update
//* the ABACKUP (D, 0, I and C) file names and the file sequence
//* number.
//*****
//*

```

Figure 44. Stacking Aggregates - Sample JCL (Part 2 of 2)

Aggregate recovery from the stacked tape

To recover aggregates from the stacked tape, specify the XMIT and the NOSTACK parameters on the ARECOVER command. The following example shows the syntax of the ARECOVER command needed to recover an aggregate that has gone through the stacking process. To use this sample command, replace

The example below shows the format of the EXPIREBV command that you can use to expire aggregate backup versions, where *aggregate_name* is the name of the aggregate group and *n* is the number of versions of that aggregate group that you want to keep.

```
EXPIREBV EXECUTE ABARSVERSIONS (AGNAME(aggregate_name) RETAINVERSIONS(n)
```

Note: If you specify SETSYS EXITON(ARCTVEXT), DFSMSHsm calls the installation exit ARCTVEXT for each volume that is being processed, indicating to the exit, that this is an ABARS tape volume. If you install DFSMSrmm, DFSMSHsm calls the EDGTVEXT installation exit instead.

The example below shows how you can use the EXPIREBV command to delete the oldest version of the aggregate, if four versions of the aggregate AGM1 are stacked on four separate tapes. After the command executes, the oldest version is deleted and the tape that it is on is returned to the tape installation exit (even if there are other valid aggregate backups on the tape). The three remaining versions that were backed up after the oldest version was backed up, are kept:

```
EXPIREBV EXECUTE ABARSVERSIONS (AGNAME(AGM1) RETAINVERSIONS(3)
```

As the tape installation exit (if active) will be called for any tape that contains the version being expired and deleted, you must either code your tape installation exit not to release ABARS tapes to scratch or do not run EXPIREBV ABARSVERSIONS unless you are expiring all the versions on a particular tape. If a tape is returned to scratch that still contains valid aggregates on it and the tape is reused by another job, future roll-off or EXPIREBV ABARSVERSIONS processing may again release the tape to scratch as ABARS still has record of the previously unexpired versions on the tape.

RACF protection of tapes used for stacking

You can use RACF protection for tapes with stacked aggregate backup files. In these examples *volser* is the serial number of the stacked tape's volume.

- If you are using a new tape to stack aggregates, use these commands to protect it with RACF:

```
SETSYS TAPESECURITY(RACF)  
RDEFINE TAPEVOL HSMABR  
RALTER TAPEVOL HSMABR ADDVOL(volser)
```

- If you are using a tape with existing ABACKUP version to stack additional aggregates, use the these commands to protect them with RACF:

```
SETSYS TAPESECURITY(RACF)  
RDEFINE TAPEVOL HSMABR
```

RACF protection of tapes after EXPIREBV processing of versions

When you enter the EXPIREBV ABARSVERSIONS command to expire specific aggregate backup versions from a stacked tape, the DFSMSHsm automatically removes the tape from the RACF tape volume set. If there are still valid aggregate backup files on the stacked tape, enter the RACF RALTER command to add the tape's *volser* back to DFSMSHsm's RACF tape volume set. In this example, *volser* is the serial number of the stacked tape.

```
RALTER TAPEVOL HSMABR ADDVOL(volser)
```

To remove a stacked tape from DFSMSHsm's RACF tape volume set, enter the RACF RALTER command:

```
RALTER TAPEVOL HSMABR DELVOL(volser)
```


End of volume during the stacking process

If you try to maximize the number of aggregate backup versions you can stack onto a single tape, you might reach the end of volume (EOV). If EOV occurs before all of the ABACKUP files associated with the specific version are copied, DFSMSHsm issues the error message IEC151I:

```
IEC151I A13-rc,mod,jjj,sss, ddname .-#. ,dev,ser,dsname
```

For example, if the sample COPYAGGR JCL, described in [“Stacking multiple aggregates on one tape”](#) on page 362 fails during the copy of the O file, in JCL STEP2, DDNAME SYSUT2, using output device address 0B31, writing to volser A00040, writing the data set AGM1.O.C01V00020, DFSMSHsm issues, this message:

```
IEC151I A13-18,IFG0195B,COPYAGGR,STEP2,SYSUT2,0B31,A00040,AGM1.O.C01V0002
```

If an EOV error occurs during an aggregate copy (stacking), you must follow the procedure described in the [“Cleanup after stacking errors”](#) on page 367 to recover that aggregate.

Cleanup after stacking errors

If a problem occurs during the stacking procedure, you must uncatalog the ABACKUP files associated with the bad aggregate version on the stacked tape, and recatalog the files on the original backup tape to avoid loss of data. The example described in [“End of volume during the stacking process”](#) on page 367 shows the EOV error that occurred during the writing of the O file. In the sample COPYAGGR JCL, described in [“Stacking multiple aggregates on one tape”](#) on page 362, the D file is copied first, followed by the O file, the I file, and the C file. After the EOV error, version 2 of the aggregate AGM1 has all of the D file, and part of the O file on the stacked tape volser (A00040). The I file and the C file are on the original tape volser (A00041). To recover the data associated with version 2 of the aggregate AGM1 or attempt to stack this version onto a new tape, you must:

1. Uncatalog the D file and the O file that reside on the stacked tape A00040.
2. Recatalog the D file and the O file onto their original tape A00041.

This example shows a sample JCL that you can use to recatalog an entire aggregate onto the original ABACKUP file. This JCL recatalogs all of the ABACKUP files associated with version 2 of aggregate AGM1 onto the original ABACKUP volser.


```
//DEFNVSAM JOB ,CLASS=A,USER=userid,PASSWORD=pswd,
// MSGCLASS=A
//*****
// *
// * Job Name: DEFNVSAM *
// *
// * Function: Use this job to catalog the ABACKUP output files, *
// * onto their original tape volser, when an error occurs during *
// * the IEBGENER stacking JCL job. *
// *
// * Before running this job, make sure the ABACKUP files, *
// * associated with the failed aggregate copy, are uncataloged. *
// *
//*****
// *
// *-----
//STEP0001 EXEC PGM=IDCAMS Catalog D file
// *-----
//SYSPRINT DD SYSOUT=A
//SYSABEND DD DUMMY
//SYSIN DD *
DEFINE NONVSAM -
(NAME(AGM1.D.C01V0002) -
DEVICETYPES(3480) -
FILESEQUENCENUMBERS(1) -
VOLUMES(A00041)) -
CATALOG(HSMLIBC.USERCAT)
/*
// *-----
//STEP0002 EXEC PGM=IDCAMS Catalog O file
// *-----
//SYSPRINT DD SYSOUT=A
//SYSABEND DD DUMMY
//SYSIN DD *
DEFINE NONVSAM -
(NAME(AGM1.O.C01V0002) -
DEVICETYPES(3480) -
FILESEQUENCENUMBERS(2) -
VOLUMES(A00041)) -
CATALOG(HSMLIBC.USERCAT)
/*
// *-----
//STEP0003 EXEC PGM=IDCAMS Catalog I file
// *-----
//SYSPRINT DD SYSOUT=A
//SYSABEND DD DUMMY
//SYSIN DD *
DEFINE NONVSAM -
(NAME(AGM1.I.C01V0002) -
DEVICETYPES(3480) -
FILESEQUENCENUMBERS(3) -
VOLUMES(A00041)) -
CATALOG(HSMLIBC.USERCAT)
/*
// *-----
//STEP0004 EXEC PGM=IDCAMS Catalog C file
// *-----
//SYSPRINT DD SYSOUT=A
//SYSABEND DD DUMMY
//SYSIN DD *
DEFINE NONVSAM -
(NAME(AGM1.C.C01V0002) -
DEVICETYPES(3480) -
FILESEQUENCENUMBERS(4) -
VOLUMES(A00041)) -
CATALOG(HSMLIBC.USERCAT)
```

Figure 46. Sample JCL to recatalog an entire aggregate onto the original ABACKUP file

Copying ABARs aggregates to another device type

If you want to copy individual aggregates from one device type, for example from a 3490 to a 3590, use the following procedure.

Note that the current design of the aggregate backup function allows you to use the STACK parameter on the ABACKUP command to back up only one aggregate, to a minimum of one tape.

ABARs uses the prefix to define the aggregate. It identifies the output data sets created by aggregate backup. These output data sets are created with the following naming conventions:

- *outputdatasetprefix.D.CnnVnnnn* and *outputdatasetprefix.O.CnnVnnnn* for the data file
- *outputdatasetprefix.C.CnnVnnnn* for the control file

- *outputdatasetprefix.I.CnnVnnnn* for the instruction/activity log file

The sequence of files that are associated with ABACKUP:

1. Data (D)
2. Other Data (O)
3. Instruction and Activity Log (I)
4. Control (C)

When you copy individual aggregates from one device type to another, you should review your tape management retention rules to ensure that the newly copied aggregates are retained permanently. You may also need to expire the old tapes manually.

Aggregate backup can be performed in NOSTACK or STACK mode. The following sample JCL shows how to copy a C-, D-, I-, and O-file from an existing tape to a new scratch tape and a new device type, assuming that ABACKUP NOSTACK has been issued:


```
//*****
//* COPY THE D FILE FROM EXISTING TAPE TO A NEW SCRATCH TAPE
//* FILE SEQUENCE IS 1.
//* IF THE COPY IS SUCCESSFUL, THE D FILE NEW VERSION
//* WILL BE CATALOGED
//*****
//STEP1 EXEC PGM=IEBGENER,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=AGM1.D.C01V0001,DISP=(OLD,UNCATLG,KEEP),
//      UNIT=3590-1,VOL=SER=A22999,LABEL=(1,SL)
//SYSUT2 DD DSN=AGM1.D.C01V0001,
//      UNIT=3480,
//      DCB=*.SYSUT1,LABEL=(1,SL,,EXPDT=99365),
//      DISP=(,CATLG,DELETE)
//SYSIN DD DUMMY
//*****
//* COPY THE O FILE FROM EXISTING TAPE AND STACK TO NEW TAPE
//* CREATED IN STEP 1. FILE SEQUENCE IS 2,
//* IF THE COPY IS SUCCESSFUL, THE O FILE NEW VERSION
//* WILL BE CATALOGED
//*****
//STEP2 EXEC PGM=IEBGENER,TIME=1439
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=AGM1.O.C01V0001,DISP=(OLD,UNCATLG,KEEP),
//      UNIT=3590-1,VOL=SER=A22999,LABEL=(2,SL)
//SYSUT2 DD DSN=AGM1.O.C01V0001,
//      UNIT=3480,
//      VOL=(,RETAIN,,9,REF=*.STEP1.SYSUT2),
//      DCB=*.SYSUT1,LABEL=(2,SL,,EXPDT=99365),
//      DISP=(,CATLG,DELETE)
//SYSIN DD DUMMY
//*****
//* COPY THE I FILE FROM EXISTING TAPE TO NEW TAPE
//* IF THE COPY IS SUCCESSFUL, THE I FILE NEW VERSION
//* WILL BE CATALOGED
//*****
//STEP3 EXEC PGM=IEBGENER,REGION=0M
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=AGM1.I.C01V0001,DISP=(OLD,UNCATLG,KEEP),
//      UNIT=3590-1,VOL=SER=A22998,LABEL=(1,SL)
//SYSUT2 DD DSN=AGM1.I.C01V0001,
//      UNIT=3480,
//      DCB=*.SYSUT1,LABEL=(1,SL,,EXPDT=99365),
//      DISP=(,CATLG,DELETE)
//SYSIN DD DUMMY
//*      VOL=(,RETAIN,,9,REF=*.STEP2.SYSUT2),
//*****
//* COPY THE C FILE FROM EXISTING TAPE TO NEW TAPE
//* IF THE COPY IS SUCCESSFUL, THE C FILE NEW VERSION
//* WILL BE CATALOGED
//*****
//STEP4 EXEC PGM=IEBGENER,TIME=1439
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DSN=AGM1.C.C01V0001,DISP=(OLD,UNCATLG,KEEP),
//      UNIT=3590-1,VOL=SER=A22997,LABEL=(1,SL)
//SYSUT2 DD DSN=AGM1.C.C01V0001,
//      UNIT=3480,
//*      VOL=(,RETAIN,,9,REF=*.STEP3.SYSUT2),
//      DCB=*.SYSUT1,LABEL=(1,SL,,EXPDT=99365),
//      DISP=(,CATLG,DELETE)
//SYSIN DD DUMMY
//*
```

Figure 47. Sample JCL to copy files from an existing tape to a new scratch tape and new device type, NOSTACK mode

The following sample JCL shows how to copy a C-, D-, I-, and O-file from an existing tape to a new scratch tape and a new device type, assuming that ABACKUP STACK has been issued (note the file sequence number in the LABEL= statement):


```

//*****
//* COPY THE D FILE FROM EXISTING TAPE TO A NEW SCRATCH TAPE
//* FILE SEQUENCE IS 1.
//* IF THE COPY IS SUCCESSFUL, THE D FILE NEW VERSION
//* WILL BE CATALOGED
//*****
//STEP1 EXEC PGM=IEBGENER,REGION=0M
//SYSPRINT DD SYSOUT=H
//SYSUT1 DD DSN=AGM1.D.C01V0001,DISP=(OLD,UNCATLG,KEEP),
//      UNIT=3590-1,VOL=SER=A22999,LABEL=(1,SL)
//SYSUT2 DD DSN=AGM1.D.C01V0001,
//      UNIT=3480,
//      DCB=*.SYSUT1,LABEL=(1,SL,,EXPDT=99365),
//      DISP=(,CATLG,DELETE)
//SYSIN DD DUMMY
//*****
//* COPY THE O FILE FROM EXISTING TAPE AND STACK TO NEW TAPE
//* CREATED IN STEP 1. FILE SEQUENCE IS 2,
//* IF THE COPY IS SUCCESSFUL, THE O FILE NEW VERSION
//* WILL BE CATALOGED
//*****
//STEP2 EXEC PGM=IEBGENER,REGION=0M
//SYSPRINT DD SYSOUT=H
//SYSUT1 DD DSN=AGM1.O.C01V0001,DISP=(OLD,UNCATLG,KEEP),
//      UNIT=3590-1,VOL=SER=A22999,LABEL=(2,SL)
//SYSUT2 DD DSN=AGM1.O.C01V0001,
//      UNIT=3480,
//      VOL=(,RETAIN,,9,REF=*.STEP1.SYSUT2),
//      DCB=*.SYSUT1,LABEL=(2,SL,,EXPDT=99365),
//      DISP=(,CATLG,DELETE)
//SYSIN DD DUMMY
//*****
//* COPY THE I FILE FROM EXISTING TAPE AND STACK TO NEW TAPE
//* CREATED IN STEP 1. FILE SEQUENCE IS 3,
//* IF THE COPY IS SUCCESSFUL, THE I FILE NEW VERSION
//* WILL BE CATALOGED
//*****
//STEP3 EXEC PGM=IEBGENER,REGION=0M
//SYSPRINT DD SYSOUT=H
//SYSUT1 DD DSN=AGM1.I.C01V0001,DISP=(OLD,UNCATLG,KEEP),
//      UNIT=3590-1,VOL=SER=A22999,LABEL=(3,SL)
//SYSUT2 DD DSN=AGM1.I.C01V0001,
//      VOL=(,RETAIN,,9,REF=*.STEP2.SYSUT2),
//      UNIT=3480,
//      DCB=*.SYSUT1,LABEL=(3,SL,,EXPDT=99365),
//      DISP=(,CATLG,DELETE)
//SYSIN DD DUMMY
//*****
//* COPY THE C FILE FROM EXISTING TAPE AND STACK TO NEW TAPE
//* CREATED IN STEP 1. FILE SEQUENCE IS 4,
//* IF THE COPY IS SUCCESSFUL, THE C FILE NEW VERSION
//* WILL BE CATALOGED
//*****
//STEP4 EXEC PGM=IEBGENER,REGION=0M
//SYSPRINT DD SYSOUT=H
//SYSUT1 DD DSN=AGM1.C.C01V0001,DISP=(OLD,UNCATLG,KEEP),
//      UNIT=3590-1,VOL=SER=A22999,LABEL=(4,SL)
//SYSUT2 DD DSN=AGM1.C.C01V0001,
//      UNIT=3480,
//      VOL=(,RETAIN,,9,REF=*.STEP3.SYSUT2),
//      DCB=*.SYSUT1,LABEL=(4,SL,,EXPDT=99365),
//      DISP=(,CATLG,DELETE)
//SYSIN DD DUMMY
//*

```

Figure 48. Sample JCL to copy files from an existing tape to a new scratch tape and new device type, after ABACKUP STACK

After the copy has completed, you can verify that the copies have been created and cataloged to the new device/volume by issuing LIST AGGREGATE(agname).

```
LIST AGGREGATE(agname)
```

If the new tape that was created does not show in LIST AGGREGATE(agname) output (this may also occur if you are at a disaster recovery site and the BCDS does not have the ABR record), you can instruct DFSMSHsm not to use the volser and unit information in the control file to drive the recovery by first performing an ARECOVER PREPARE with the XMIT parameter:

```
ARECOVER DSNAME(AGM1.C.C01V0001) XMIT PREPARE
```


ARECOVER will search the catalog for the control and data files and, if found, use the unit and volser from the catalog when allocating these data sets. Note that UNIT and VOLUMES are invalid if the XMIT parameter is used. If specified, you will receive an ARC6002E error message.

After ARECOVER PREPARE has successfully completed, you may perform aggregate recovery of data sets. Note that user tape data sets listed in the ACCOMPANY list will be recovered to the new device type.

This procedure does not release/scratch the source aggregate backup tape. The source aggregate backup tape must be manually released to the scratch tape pool through your tape management system. Please refer to the tape management system documentation to determine the steps that are required to release the tape to the scratch pool.

How aggregate recovery works

To perform aggregate recovery, you must have the control file, data files, and instruction/activity log file created by a previous aggregate backup. To use the simplified ARECOVER command, you must also have an ABARS activity (ABR) record at the recovery site. You can then use the ARECOVER command to recover the data sets from the tape files.

Figure 49 on page 372 shows the information flow of an aggregate recovery.

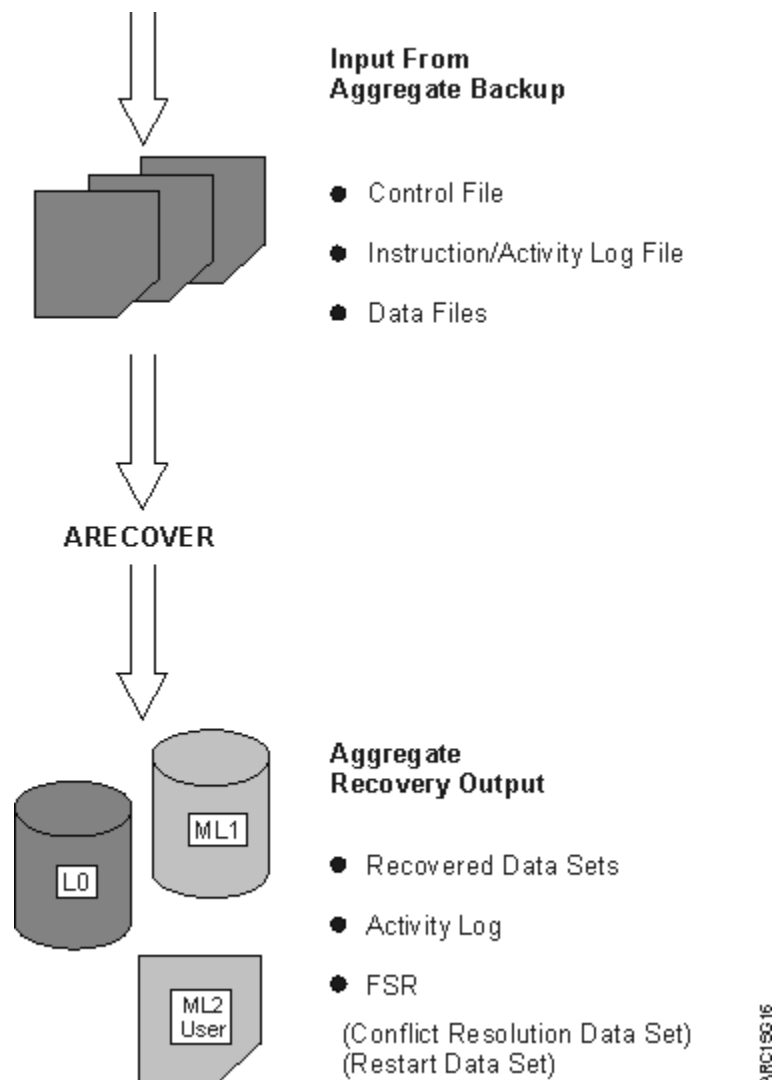


Figure 49. Aggregate Recovery Information Flow

You can choose to prepare the recovery site for the eventual recovery of the aggregate by issuing the ARECOVER command with the PREPARE parameter. You can also choose to verify the aggregate recovery

without actually recovering the data sets. Then, when you are ready, you can perform the actual recovery of your data sets.

The control file and data file were created as a result of aggregate backup and are used as input for aggregate recovery.

When you issue the ARECOVER command, DFSMSHsm creates:

- The recovered data sets, which are those data sets recovered from the aggregate backup tape data file. [Table 41 on page 373](#) shows the manner in which aggregate recovery recovers, allocates, and catalogs data sets.
- An activity log, which contains messages written during aggregate recovery processing.
- The FSR, which contains the statistics for aggregate recovery processing. This includes recording the CPU time for processing ARECOVER requests. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*).
- Updates to the ABR record, which shows the CPU time for processing ARECOVER requests. The ABR shows the accumulated CPU time of each restart until the recovery of the aggregate is successful.

Aggregate recovery can also create the following:

- A conflict resolution data set that is used as input to a subsequent ARECOVER command.

The *conflict resolution data set* contains names of data sets that the ARECOVER verification process determines to have like-named conflicts that were not resolved by the RECOVERNEWNAME parameters, the DATASETCONFLICT parameter, or the ARCCREXT installation exit. For each such data set encountered, an entry is placed into the conflict resolution data set, with a default action of BYPASS.

You can edit the conflict resolution data set before issuing the next ARECOVER command, and indicate which conflict resolution actions are to be taken by ARECOVER verification processing for each data set entry.

You then issue a subsequent ARECOVER command. The conflict resolution data set is read and used as input to the ARECOVER verification process.

- A restart data set is created when the recovery of data sets does not complete successfully. The restart data set contains the names of all the data sets recovered successfully. When an aggregate recovery is reissued, the restart data set ensures that data sets already processed are not recovered a second time. If you want to start aggregate recovery again from the beginning, delete the restart data set.
- Accounting code records that are written to the ABACKUP control file, FSR, and ABR. You can assign accounting codes to aggregates, which allows you to associate ABARS CPU time to specific aggregates. The codes are written to the ABACKUP control file so they can be propagated at the recovery site.

Related reading

- For more information about creating the ABR record, see [“How to create the ABR record” on page 733](#).
- See [“EAV considerations for recovery functions” on page 15](#).

Recovered aggregate data sets

A data set is recovered differently if it is an INCLUDE, ACCOMPANY, or ALLOCATE data set, as specified in a selection data set during aggregate backup. [Table 41 on page 373](#) summarizes how these different types of data sets are recovered.

Table 41. Aggregate Recovery Data Set Processing

Selection Data Set Keyword	Data Set Recovered	Data Set Allocated	Data Set Cataloged
INCLUDE	X	X	X
ALLOCATE		X	X

Table 41. Aggregate Recovery Data Set Processing (continued)

Selection Data Set Keyword	Data Set Recovered	Data Set Allocated	Data Set Cataloged
ACCOMPANY			X

Additionally, the type of device on which an INCLUDE data set is recovered is directly based on its disposition at the aggregate backup site.

Example aggregate recovery site

In “[BACKUPPREFIX: Specifying the prefix for the backup version name or the dump copy data set name](#)” on page 1107, an example aggregate backup site was used to show backup procedures. This same example is used to show how to recover applications at the example recovery site.

The following INCLUDE data sets are recovered to DASD:

- PAY2.PY002.SOCSEC
- PAY2.PY002.STATUS.G0002.V00
- PAY1.PY001.CHECK
- PAY1.PY002.IRA
- PAY1.PY001.BONUS
- PAY1.PY002.LADDER

The following INCLUDE data set are recovered to tape:

- PAY1.PY001.NAMES

The following ALLOCATE data set are allocated on a DASD:

- PAY2.PY001.BENEFIT

The following ACCOMPANY tape data set are cataloged:

- PAY2.PY001.TAXES

Preparing for aggregate recovery

You need to consider the following in order to prepare for aggregate recovery:

- Retrieving instructions or information from the instruction data set that may have been optionally created at the aggregate backup site.
- Issuing ARECOVER with the PREPARE parameter to:
 - Determine the amount of storage needed to recover aggregate groups.
 - Catalog ABACKUP outfiles.
 - Add volumes to RACF, if SETSYS TAPESECURITY(RACF) or SETSYS TAPESECURITY(RACFINCLUDE) is specified, RACF is active and the HSMABR tape volume set is defined.
- Issuing ARECOVER with the INSTRUCTION and ACTIVITY parameters
- Creating an ABR record
- Determining the aggregate recovery site constraints.
- Planning for aggregate recovery requirements for SMS.
- Defining the target volumes to be used for aggregate recovery of non-SMS data with the DEFINE ARPOOL command.

If you have multivolume, non-VSAM, non-SMS-managed data sets, ensure that you define an adequate number to level 0 volumes for this aggregate group, using the LOVOLS subparameter of the DEFINE ARPOOL command. If no ARPOOL is defined for this aggregate group, you must have an adequate

number of volumes ADDVOled as primary volumes, or you may have allocation failures during ARECOVER processing of this aggregate group.

- Defining aggregate recovery environmental options with the SETSYS command.
- Defining a model entity to define discrete data set profiles to RACF.
- Determining if the RACF facility class is active and at least one profile is defined, so that the user authority to use the ARECOVER command is verified.
- Determining if the RACF facility class is active and if there is a REPLACE action in either the conflict resolution data set, the DSCONFLICT parameter of the ARECOVER command, or the ARCCREXT exit. If so, define the following facility class profiles and add the user to at least one of them:

STGADMIN.ARC.ARECOVER

STGADMIN.ARC.ARECOVER.*agname*.REPLACE

STGADMIN.ARC.ARECOVER.*agname*.RENTGT

- Determining if the aggregate backup output has been sent by a transmit program such as the NetView FTP. If you are using NetView FTP, catalog the data file data sets (using the disposition parameter) and any accompanying data sets that have been transmitted. If you are using another file transfer program, ensure that the files are cataloged.
- If you choose to have any or all of your ABACKUP output files directed to DASD volumes, there are special precautions you must observe for successful recovery of the backed up data:
 - The aggregate backup files must be properly cataloged and exist at the recovery site.
 - You must use the XMIT and DATASETNAME parameters on your ARECOVER command.

Table 42 on page 375 shows a checklist you may want to follow as you complete each of the steps needed to prepare for and perform aggregate recovery.

Table 42. Aggregate Recovery Checklist	
Step	Action
___ 1.	Verify receipt of the aggregate backup tapes containing the control, data, and instruction/activity log files.
___ 2.	Issue the DEFINE ARPOOL command to specify the target volumes to be used by aggregate recovery for non-SMS-managed data.
___ 3.	Run ARECOVER with PREPARE, INSTRUCTION, and ACTIVITY parameters
___ 4.	Determine the aggregate recovery constraints. <ol style="list-style-type: none"> 1. Determine the DASD resources needed to perform the aggregate recovery. 2. Determine the tape unit name to be used for output data sets to be recovered. 3. Determine the tape unit name if recovering to ML2 tape.
___ 5.	Ensure that the ABARS JCL procedure has been installed in the systems JCL library (refer to the z/OS DFSMSshm Implementation and Customization Guide).

Table 42. Aggregate Recovery Checklist (continued)

Step	Action
___ 6.	<p>Determine the SETSYS parameters to be used:</p> <ul style="list-style-type: none"> • Specify the number of ABARS address spaces • Specify activity log type, message level, and deletion processing • Indicate ABARS procname • Indicate target unit name for recovering user tape data sets • Indicate target unit name if recovering to ML2 tape • Specify space utilization for non-SMS level 0 DASD • Specify ABARS buffer usage • Specify whether the ABACKUP output tapes were stacked • Specify whether you want the ABARS FSR record written as an SMF record (SETSYS SMF) • Indicate how SMS-managed GDSs backed up from level 0 DASD should be restored. • Indicate the desired ABARSVOLCOUNT environment
___ 7.	Add the aggregate recovery procedure name defined by SETSYS ABARSPROCNAME to the RACF started task table (ICHRIN03).
___ 8.	Define the model entity to define discrete data set profiles to RACF. This step is optional.
___ 9.	Link-edit the aggregate recovery installation exits. This step is optional.
___ 10.	If the aggregate backup output has been transmitted, ensure all aggregate backup output files and any ACCOMPANY data sets are cataloged. See “Preparing for aggregate recovery” on page 374.
___ 11.	Use RACF facility class to authorize certain users or operators to issue the ARECOVER command.
___ 12.	<p>Issue the ARECOVER command with the VERIFY parameter. This step is optional.</p> <ul style="list-style-type: none"> • If aggregate recovery verification completes successfully with a return code of zero, continue to step 13. • If aggregate recovery verification fails, use the conflict resolution data set to assist in resolving any conflicts, and repeat this step until the recovery verification completes successfully with a return code of zero.
___ 13.	<p>Issue the ARECOVER command with the EXECUTE parameter.</p> <ul style="list-style-type: none"> • Determine if aggregate recovery completed successfully with a return code of zero. If not, correct any errors and issue the ARECOVER command with the VERIFY parameter. If verification is successful, reissue the ARECOVER command with the EXECUTE parameter. • If aggregate recovery verification fails, use the conflict resolution data set to assist in resolving any conflicts, and repeat this step until the recovery verification completes successfully with a return code of zero. • Issuing the ARECOVER command with the VOLCOUNT(ANY) parameter can help overcome some SMS allocation failures.

Authorization checking during aggregate recovery

You can permit all console operators or any user, including those who are not DFSMSHsm authorized, to issue the ARECOVER command by using RACF FACILITY class profile support.

If the RACF FACILITY class profile support is installed and activated, DFSMSHsm checks the console operator or user RACF profiles for authorization to issue the ARECOVER command. The user ID of OPER must be defined to RACF before console operators can be given authority to issue the ARECOVER command. If the user ID OPER is given authority to the ARECOVER command, then all console operators are authorized to issue the ARECOVER command. You can also define group names to RACF, using the RACF ADDUSER command.

If the RACF FACILITY class profile is not active, only DFSMSHsm-authorized users and console operators are authorized to issue the ARECOVER command.

You can define the DFSMSHsm ARECOVER command to RACF by defining the following profile names:

Command	RACF Profile
ARECOVER	STGADMIN.ARC.ARECOVER
ARECOVER	STGADMIN.ARC.ARECOVER. <i>agname</i>
ARECOVER	STGADMIN.ARC.ARECOVER. <i>agname</i> .REPLACE

Specific user IDs can then be authorized to issue these commands through the use of the RACF PERMIT command.

You can restrict use of DSCONFLICT(REPLACE), REPLACE as conflict resolution data set action, or REPLACE specified by ARCCREXT, through the use of RACF facility class profile STGADMIN.ARC.ARECOVER.*agname*.REPLACE.

Related reading

For more information about the RACF FACILITY class and control of access to all commands, see “Controlling command access with RACF FACILITY class” on page 625 and [z/OS DFSMSHsm Implementation and Customization Guide](#).

Authorizing comprehensive ARECOVER command authority

When users are added to the STGADMIN.ARC.ARECOVER profile they receive comprehensive command authority. A user with comprehensive command authority for the ARECOVER command is authorized to issue any ARECOVER command.

You could use the following commands to prevent or limit the authority of users to issue the ARECOVER command:

```
RDEFINE FACILITY STGADMIN.ARC.ARECOVER
PERMIT STGADMIN.ARC.ARECOVER CLASS(FACILITY) -
  ID(user ID) ACCESS(READ)
```

The PERMIT command authorizes a user to issue the ARECOVER command for all aggregate groups.

Related reading

For more information about the RACF FACILITY class and control of access to all commands, see “Controlling command access with RACF FACILITY class” on page 625 and [z/OS DFSMSHsm Implementation and Customization Guide](#).

Authorizing restricted ARECOVER command authority

When users are added to the STGADMIN.ARC.ARECOVER.*agname* or STGADMIN.ARC.ARECOVER.*agname*.REPLACE profiles, the users have restricted command authority.

You can use the following commands to prevent or limit users authority to issue the ARECOVER command with a specific aggregate group name:

```
RDEFINE FACILITY STGADMIN.ARC.ARECOVER.PAY1
PERMIT STGADMIN.ARC.ABACKUP.PAY1 CLASS(FACILITY) -
  ID(OPER) ACCESS(READ)
```

The PERMIT command gives all console operators authority to issue the ARECOVER command with a specified aggregate group name.

Related reading

For more information about the RACF FACILITY class and control of access to all commands, see “Controlling command access with RACF FACILITY class” on page 625 and [z/OS DFSMSHsm Implementation and Customization Guide](#).

Considering storage management subsystem requirements

Prior planning is necessary if you want to recover storage management subsystem (SMS) managed data sets in an environment comparable to that which existed at the backup site. SMS must be active to recover VSAM data sets specified in the ALLOCATE list.

SMS class names and attributes are displayed in an ARC6119I message during the aggregate recovery verification. This may help you ensure that the SMS-managed data sets are recovered as you intend.

Note:

1. SMS status at the recovery site determines the status of the recovered data set. A data set that was SMS-managed at the backup site may not necessarily be recovered as SMS-managed and vice versa.
2. You may want to create an aggregate that includes the SMS control data sets from the aggregate backup site. This aggregate could then be restored before restoring the aggregates for application data. This would be the most efficient way to create an SMS environment at the recovery site that matches the SMS environment at the backup site.
3. When restoring an SMS-managed user catalog that was defined in the ALLOCATE statement, ABARS passes the volume serial number where the catalog resided at the backup site. If the target volume exists at the recovery site, the user catalog is allocated on that volume. If the original volume does not exist at the recovery site, the ACS routines at the recovery site direct the allocation. In the event that SMS determines the target volume to be invalid, ABARS requests SMS to relax the guaranteed space rules so that user catalogs can be directed to other volumes in a storage group associated with a guaranteed space storage class.

Rule: If the original volume does not exist at the recovery site, you must modify your ACS routines to remove that volume from any storage group definitions associated with a guaranteed space storage class. If you do not, the recovery will fail.

4. When restoring an SMS-managed NONVSAM data set which belonged, at the time it was backed up, to a storage class with the guaranteed space attribute, ABARS first attempts to allocate the data set to the same volumes that it was backed up from. If the original volumes are unavailable or no longer belong to a storage group that is associated with a guaranteed space storage class, the data set is allocated to different volumes existing in a storage group associated with a guaranteed space storage class.

Note: If the original volumes do not exist at the recovery site, modify the ACS routines to remove the volumes from any storage group definitions associated with a guaranteed space storage class. Otherwise the recovery fails.

Nonmigrated, SMS-managed data sets in an INCLUDE list are directed to the appropriate volumes during aggregate recovery by ACS routines. The following information is passed to ACS routines when restoring data sets backed up by DFSMSdss:

- Output volume serial numbers
- &ACSENVIR = RECOVER (environment in which ACS was invoked)
- Default data class name (if RACF installed)
- Default management class name (if RACF installed)
- Default storage class name (if RACF installed)
- Data set name
- Data set organization
- RACF owner or group (if RACF installed)
- Data set type

- Expiration date
- High-level qualifier of the data set name
- Low-level qualifier of the data set name
- Maximum size of the data set in kilobytes
- Number of qualifiers in the data set name
- Number of volumes specified by the user
- Data set record organization
- Retention period
- Size of data set in kilobytes
- Unit name

If the data sets are not directed by an ACS routine or if SMS is not active, the data sets are recovered to level 0 volumes defined with the DEFINE ARPOOL command. If no level 0 volumes are defined to an ARPOOL, aggregate recovery uses the primary volumes that have been ADDVOLD to DFSMSHsm.

The SETSYS ABARSVOLCOUNT(NONE |ANY) command affects the way in which SMS-managed data set allocations are performed by DFSMSdss for L0 data sets dumped from primary volumes.

Determining aggregate recovery site constraints

Before performing aggregate recovery, you need to determine the DASD and tape unit resources needed to recover the control file, data files, and user tape data sets. You need an equivalent number of tapes during ARECOVER as were used during ABACKUP, assuming the same technology. You can find this information using the LIST AGGREGATE command. For the most part, this information is determined at the backup site and may have been included in the instruction data set during aggregate backup. See [“Retrieving the instruction data set”](#) on page 379.

The DASD space requirements depend in part on the residency of the data sets at the backup site and to which device type they will be recovered.

DASD space requirements are displayed in the ARECOVER activity log, the ABACKUP activity log (if recovered), the ABR record created during the PREPARE step, and in the FSR.

Tip: During ABACKUP processing of an aggregate group, you can use the SETSYS ABARSKIP command to skip data sets on volumes that are protected by the peer-to-peer remote copy function or by the extended remote copy function. Ensure that those volumes are online before issuing an ARECOVER command against that aggregate group.

During aggregate backup, user tape data sets with a block size equal to or larger than 32K bytes can be included in the selection data set. User tape data sets with block sizes up to 256K-1 bytes can also be included in the selection data set. DFSMSHsm Version 2 Release 10 or a subsequent release is required to recover these data sets. In addition, your tape hardware must be capable of processing these large block sizes.

Related reading

For more information about the LIST AGGREGATE command, see [“AGGREGATE: Listing the aggregate backup and recovery activity information”](#) on page 925.

Retrieving the instruction data set

An instruction data set is optionally created for an aggregate backup. It may contain instructions or other information to help you perform that application's recovery. It is contained in the instruction data set, which is the second file on the control file tape.

The instruction data set may also include the following important information:

- Required SMS attributes

- Sample ACS routines
- Description of the application
- RACF environment
- Software requirements
- Hardware requirements
- Unique application execution instructions

To help you perform aggregate recovery, you should first retrieve the instruction data set. The instruction data set is contained in the instruction/activity log file, *outputdatasetprefix.I.CnnVnnnn*. The *outputdatasetprefix* was specified in the aggregate group during aggregate backup. You can retrieve the instruction data set by issuing the ARECOVER command with the INSTRUCTION parameter.

Example instruction data set

The instruction data set, PAYROLL.INSTRUCT, created at the example backup site and identified in the example aggregate group, PAY1, is shown in [Figure 50 on page 381](#).


```

THIS IS THE INSTRUCTION DATA SET FOR AGGREGATE GROUP PAY1.

PERFORMING AGGREGATE BACKUP ON PAY1 BACKS UP THE CRITICAL PAYROLL
FILES AT THE END OF A PAY PERIOD. RELEVANT FILES INCLUDE THE EMPLOYEE
ROSTER (PAY1.PY001.NAMES), AMOUNT OF EACH PAYCHECK (PAY1.PY001.CHECK),
AMOUNT FROM EACH CHECK PLACED IN IRA AND SOCIAL SECURITY DEDUCTION
(PAY1.PY002.IRA AND PAY2.PY002.SOCSEC), AND AMOUNT (IF ANY) OF CHECK
THAT WAS A BONUS IN ADDITION TO SALARY (PAY1.PY001.BONUS).
TO ENSURE PROPER EXECUTION, VERIFY THAT CORRECT RELEASE LEVELS ARE PRESENT
FOR DFSMSHsm, DFSMSdss, AND DFSMSdép; AND THAT THE CORRECT SMS CLASSES ARE ACTIVE.
YOU WILL NEED THE FOLLOWING SMS CLASSES AND THEIR ASSOCIATED ATTRIBUTES
SET UP TO RECOVER THOSE FILES THAT WERE SMS-MANAGED AT ABACKUP, AND TO
ENSURE THAT THEY WILL BE SMS-MANAGED UPON RECOVERY.

DATA CLASS NAME: PHYSSEQ
DESCRIPTION: PHYSICAL SEQUENTIAL DATA SETS

RECFM      : FB
LRECL      : 80
SPACE
  AVGREC   : U
  PRIMARY  : 50
  SECONDARY : 50
RETPD      : 175
VOLUME COUNT : 1

STORAGE CLASS NAME: BATCHDASD
DESCRIPTION: DASD DATA SETS PROCESSED IN BATCH MODE

AVAILABILITY:
  GUARANTEED SPACE      : NO
  GUARANTEED SYNCHRONOUS WRITE : NO

MANAGEMENT CLASS NAME : BATCH
DESCRIPTION            : BATCH DATA SETS

EXPIRATION ATTRIBUTES
  EXPIRE AFTER DAYS NON-USAGE : 150
  EXPIRE AFTER DATE/DAYS      : 366

RETENTION LIMIT :
PARTIAL RELEASE : YES

MIGRATION ATTRIBUTES
  PRIMARY DAYS NONUSAGE : 7
  LEVEL 1 DAYS NONUSAGE : 45
  COMMAND/AUTO MIGRATE  : YES

GDG MANAGEMENT ATTRIBUTES
  ROLLED OFF GDS ACTION : EXPIRE

BACKUP ATTRIBUTES
  BACKUP FREQUENCY
  # OF BACKUP VERSIONS (DS EXISTS) : 3
  # OF BACKUP VERSIONS (DS DELETED) : 1
  RETAIN DAYS ONLY BACKUP VERSION  : 366
  (DS DELETED)
  RETAIN DAYS EXTRA BACKUP VERSIONS : 100
  ADMIN OR USER COMMAND BACKUP      : YES
  AUTO BACKUP                        : YES

YOU SHOULD EXECUTE ARECOVER WITH THE VERIFY OPTION TO VERIFY THE AGGREGATE RECOVERY.

PAYROLL.INSTRUCT was copied to the instruction/activity log file,
PAY1.I.C01V0001, during aggregate backup.

```

Figure 50. Example of a Retrieved Instruction Data Set

Creating an ABR record in the BCDS

Before performing an ARECOVER execute or verify for a particular set of ABACKUP output tapes, issue an ARECOVER PREPARE command to create an ABR record for this recovery. This record contains an inventory of aggregate recovery activity for the aggregate version and allows you to use the simplified ARECOVER command syntax described in the [Chapter 29, “ARECOVER command: Recovering a specified aggregate group,”](#) on page 723 when you actually perform aggregate recovery.

The DATASETNAME(*dsname*) parameter determines the control file data set name. You can use DSNAME as an abbreviation for DATASETNAME.

VOLUMES and XMIT are mutually exclusive parameters. VOLUMES (UNIT is optional) specifies that the volume list be used to allocate the control file to be read. XMIT specifies that the required files must already be cataloged, and the system supplies the volume list.

PREPARE, VERIFY, and EXECUTE are mutually exclusive parameters. When PREPARE is specified with the ARECOVER DATASETNAME(*dsname*) command, the ABR record is created without verification processing or data movement. When VERIFY or EXECUTE are specified, an ABR record is created if it does not exist.

Related reading

For more information about the [Chapter 29, “ARECOVER command: Recovering a specified aggregate group,”](#) on page 723.

Defining ABARS tapes to RACF

The output tape volumes generated by aggregate backup processing can be defined to RACF during ARECOVER processing. This is done when the ABR record is created by the PREPARE, VERIFY, or EXECUTE functions and when the following conditions are met:

- RACF is active at the recovery site
- You have defined TAPEVOL resource class HSMABR at the recovery site
- You have specified SETSYS TAPESECURITY(RACF) or SETSYS TAPESECURITY(RACFINCLUDE)

Note: Tapes containing DFSMSHsm output files that are recovered by DFSMSHsm are also RACF-protected at the recovery site when the preceding conditions are met.

```
ADDSD 'outputdatasetprefix.**'
```

If you use RACF DATASET class profiles instead of TAPESECURITY(RACF) to protect the tapes and create the DATASET profiles, you could define ADDSD, which specifies that tapes created during the aggregate backup process are protected. You need a profile for each aggregate group you have defined that uses a different aggregate group prefix.

Issuing the DEFINE ARPOOL command

The DEFINE ARPOOL command allows you to determine the target DASD level 0 and ML1 volumes to be used for aggregate recovery. These are the volumes used as target volumes for migrated and DASD data sets that have not had their allocation directed to other volumes by SMS. Data sets can have their allocations directed by SMS only when it is installed and active at the recovery site.

To ensure that the settings are defined to DFSMSHsm at startup initialization, you should include the DEFINE command in the DFSMSHsm parmlib member (ARCCMDxx). If you are an authorized DFSMSHsm user, you can issue the DEFINE command before an aggregate recovery to modify existing pool configurations or add new pools.

To specify a pool of volumes to be used as target volumes for data sets recovered by aggregate recovery, use the ARPOOL(*agname*) parameter. The aggregate group name (*agname*) identifies a pool and relates that set of volumes to an aggregate group that was previously backed up with the ABACKUP command. These volumes are used as output volumes when the application is recovered. If an aggregate group name of * is specified, the volumes are assigned to a general pool of aggregate recovery volumes. A general pool is a group of volumes defined for recovering those aggregate groups for which no specific pool has been defined. If no aggregate group name is specified, the DEFINE command fails.

Note: Even though ABARS supports a general pool of volumes for all aggregates, you should define individual aggregate recovery pools for each aggregate. Define these pools so that the order of the volumes is varied, especially for those volumes defined with the ML1VOLS parameter. For example:

```
DEFINE ARPOOL(AGG1 ML1VOLS(VOL1,VOL2,VOL3) -  
  L0VOLS(VOLA,VOLB,VOLC))  
  
DEFINE ARPOOL(AGG2 ML1VOLS(VOL2,VOL3,VOL1) -  
  L0VOLS(VOLB,VOLC,VOLA))  
  
DEFINE ARPOOL(AGG3 ML1VOLS(VOL3,VOL1,VOL2) -  
  L0VOLS(VOLC,VOLA,VOLB))
```


The volumes defined with the ML1VOLS parameter are serially loaded in the order in which they are defined. If several aggregate recoveries are processing with ARPOOLS that have the same volume serial numbers in the same order, then volume and channel contention are increased. Varying volume order prevents different aggregate recoveries from attempting to use the same volumes. It is apparent that this is especially critical if multiple ARECOVER commands are processing simultaneously.

ALL specifies that all currently mounted ADDVOled level 0 and ML1 volumes are to be defined to the volume pools for aggregate recovery. It is equivalent to specifying the LOVOLS(*) and ML1VOLS(*) parameters.

ML1VOLS(*volser1,...volsern* | *) specifies a list of non-SMS-managed volumes to be used as target volumes to recover migrated data sets. If * is specified, all currently mounted ADDVOled ML1 volumes are used as the pool of ML1 volumes for aggregate recovery.

LOVOLS(*volser1,...volsern* | *) specifies a list of non-SMS-managed volumes to be used as target volumes to recover nonmigrated, non-SMS-managed data sets. If * is specified, all currently mounted ADDVOled primary volumes are used as the pool of level 0 volumes for aggregate recovery.

Note: If insufficient space is defined, the data set recovery fails. If the level 0 or ML1 volumes specified in the pool are not online and mounted, aggregate recovery fails.

If the DEFINE ARPOOL command is issued without the ALL, ML1VOLS, and LOVOLS parameters, the aggregate recovery pool specified by *agname* is deleted. If an aggregate group name of * is specified, the general pool of aggregate recovery volumes is deleted.

If a DEFINE ARPOOL command was not previously issued before an ARECOVER command, DFSMSHsm uses all currently mounted ADDVOled primary (59 maximum) and ML1 volumes as the ARPOOL.

The aggregate recovery pools that have been defined can be displayed by using the QUERY ARPOOL command. If LOVOLS have been specified, these volume serial numbers are passed to ACS by DFSMSdss. ACS routines may then use these volume serial numbers to determine the classes and placement of the data sets.

If ACS does not assign storage classes to the data sets, then DFSMSdss attempts to direct allocation to the volume serial number where the data set resided when backed up. If that particular volume serial number does not exist at the recovery site, recovery of the data set in question fails. In a non-SMS environment, all level 0 data sets are directed to LOVOLS volumes.

The ARPOOL volumes need not be added to DFSMSHsm using the ADDVOL command before issuing the DEFINE ARPOOL command; however, the following restrictions apply to the relationship between DFSMSHsm-owned and -managed volumes and volumes assigned to the aggregate recovery pool:

- A volume previously added to DFSMSHsm as anything other than a PRIMARY volume cannot be defined with the LOVOLS parameter of the DEFINE ARPOOL command.
- A volume previously added to DFSMSHsm as anything other than an ML1 volume cannot be defined with the ML1VOLS parameter of the DEFINE ARPOOL command.
- A volume previously defined to aggregate recovery with the ML1VOLS parameter of the DEFINE ARPOOL command can subsequently be added to DFSMSHsm only as an ML1 volume.
- A volume previously defined to aggregate recovery with the LOVOLS parameter of the DEFINE ARPOOL command can subsequently be added to DFSMSHsm only as a PRIMARY volume.
- A volume cannot be SMS managed.
- A volume defined as LOVOLS cannot also be defined as ML1VOLS. Conversely, a volume defined as ML1VOLS cannot also be defined as LOVOLS. This is true whether it conflicts with a previously defined pool or whether a conflict exists within the pool being defined.

You should use the ADDVOL command to add all aggregate recovery volumes to DFSMSHsm as soon as possible after an ARECOVER command has successfully completed processing. If you do not do this, failures can occur when an attempt is made to refer to those data sets.

Related reading

- For more information about the DEFINE command, see [Chapter 40, “DEFINE command: Defining control structures for use by DFSMSHsm,”](#) on page 809.
- For more information about the QUERY command, see [Chapter 58, “QUERY command: Displaying the status of parameters, statistics, and requests,”](#) on page 981.
- For more information about the DEFINE ARPOOL command, see [“ARPOOL: Identifying the volumes for aggregate recovery”](#) on page 813.

Example DEFINE ARPOOL command for aggregate recovery

The following DEFINE ARPOOL command is entered at the example site before performing aggregate recovery:

```
DEFINE ARPOOL(PAY1 ML1VOLS(MIG102) L0VOLS(PRIM02))
```

This command specifies that the migration volume (MIG102) and the primary volume (PRIM02) are used for the aggregate recovery of data sets identified in the example aggregate group, PAY1.

To review the aggregate recovery pools that have been defined, enter

```
QUERY ARPOOL
```

Note: Define ARPOOLS to minimize volume contention when running more than one aggregate recovery at the same time. If two or more aggregate recoveries are attempted using the same level 0 volume, one or more of the aggregate recoveries can fail because of the inability of DFSMSdss to allocate the receiving volume.

Defining SETSYS parameters for aggregate recovery

The SETSYS command allows you to externally define various aggregate recovery environmental options.

To ensure that the settings are defined to DFSMSHsm at startup initialization, the system programmer should include the SETSYS command in the DFSMSHsm parmlib member (ARCCMDxx). If you are an authorized DFSMSHsm user, you can issue the SETSYS command before an aggregate recovery to modify the SETSYS parameters that control aggregate processing.

To specify how many ABARS commands can run concurrently, use the MAXABARSADDRESSSPACE parameter. MAXABARSADDRESSSPACE specifies the maximum number of concurrent ABARS secondary address spaces that DFSMSHsm supports (up to 64). When determining the value to specify for MAXABARSADDRESSSPACE, factor in such items as system load, resource contention, aggregate backup and recovery activity. The overriding factor is the number of tape drives available (the default for MAXABARSADDRESSSPACE is 1).

When an ARECOVER command completes, ABARS determines if there is another ABARS request queued that has not been processed. If there is, DFSMSHsm reuses the address space to process that command. The address space remains active until there are no further ABARS commands queued, or the SETSYS MAXABARSADDRESSSPACE command has reduced the number of ABARS address spaces and this address space is no longer required.

The tape unit type used for the control and data file must be compatible with the tape unit type used during aggregate backup. You will specify this unit type later as part of the UNIT parameter of the ARECOVER command or in the SETSYS ABARSUNITNAME parameter.

The ABARSPROCNAME parameter specifies the procedure name used to start an ABARS secondary address space when no such address space is available for reuse. The default is DFHSMABR.

The EXITON(*exitname*) parameter lets you specify which installation exits (ARCCREXT, ARCSKEXT, ARCTVEXT) are active during aggregate recovery processing.

The OPTIMUMDASDBLOCKING parameter allows efficient utilization of DASD when ABARS performs aggregate recovery at your recovery site. When ARECOVER recovers a data set to ML1 DASD and SETSYS OPTIMUMDASDBLOCKING has been specified, ARECOVER determines the maximum block size for the data set on the ML1 DASD. This information is used for the allocation block size of the migrated data set. The 2K records of the input migrated data set are reblocked to the optimum block size for the output data set.

If SETSYS NOOPTIMUMDASDBLOCKING is specified, DFSMSHsm does not define optimum block size for the output data set.

The ARECOVERUNITNAME parameter specifies a default tape unit name for user tape data sets being recovered to a tape unit during recovery processing. If the TARGETUNIT parameter of the ARECOVER command is specified, its value takes precedence over any value specified for the ARECOVERUNITNAME parameter. However, if neither parameter is specified, a default of 3590-1 is assumed. This parameter can be overridden if SMS is active and ACS routines assign a tape storage group to the data set.

The ABARSVOLCOUNT(NONE | ANY) command allows you to affect the method of invoking DFSMSdss for restoring L0 data sets that were dumped from primary volumes. If NONE is specified, the DFSMSdss VOLCOUNT parameter is not passed to DFSMSdss during the restore. If ANY is specified, the VOLCOUNT(ANY) parameter is passed to DFSMSdss during the restore. Specifying VOLCOUNT(ANY) to DFSMSdss results in the allocation of the target data set on as many volumes as required, up to a maximum of 59 volumes.

The ARECOVERPERCENTUTILIZED(*nnn*) parameter allows you to specify to what percent DFSMSHsm allows DFSMSdss to fill non-SMS DASD recovery volumes during any ARECOVER command processing. For *nnn*, substitute an integer number between 1 and 100. The default is 080 (80%).

Tip: When you issue any ARECOVER command, you can temporarily override the SETSYS ARECOVERPERCENTUTILIZED value by using the PERCENTUTILIZED parameter of the ARECOVER command.

Each time you issue the ARECOVER command, an activity log is created. You can use the SETSYS ABARSACTLOGTYPE parameter to specify whether you want the activity log automatically written to SYSOUT or to DASD. You can use the SETSYS ABARSDELETEACTIVITY(Y | N) command to specify whether you want DFSMSHsm to automatically delete the ABARS activity log that corresponds to the ABACKUP version being rolled off. The deletion occurs during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing, and removes the need to manually manage the ABARS activity logs.

The default, if you do not specify the SETSYS ABARSTAPES command, is the STACK option.

When SETSYS ABARSTAPES(STACK) is specified, ARECOVER attempts to process the ABACKUP output files as if they are stacked. If the files are not stacked, ARECOVER detects the error and attempts to process the files as if ARECOVER were issued with the NOSTACK option. When SETSYS ABARSTAPES(NOSTACK) is specified, ARECOVER attempts to process the ABACKUP output files as if they are not stacked. If the files are stacked, ARECOVER detects the error and attempts to process the files as if ARECOVER were issued with the NOSTACK option.

Specifying the XMIT parameter with the ARECOVER command when using stacked output tapes has the following effect on ARECOVER processing:

- ARECOVER relies on information in the catalog to determine the file sequence numbers during the allocation of ABACKUP files. If the ABACKUP files, with the exception of the control file, are not cataloged, ARECOVER obtains the file sequence numbers from the control file.
- XMIT processing for STACKED output remains basically the same as for data that is not stacked, meaning that deallocation and allocation occur for each ABACKUP file processed. This can result in several mount requests for the same tape volume.
- If the ABACKUP output is transmitted, it is possible for the data to be received on a device type and geometry different from that on which it was originally written. Because of this, you cannot assume that the files will still be stacked when received by the network transmission software.

ABARS records the CPU time for processing ARECOVER requests, along with any applicable account code information. This information is written to the ABR record and can be obtained using the LIST

AGGREGATE command. This information is written to the FSR control block. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*). ABARS then writes the FSR as *smfid* + 1 in the SYS1.MANx or SYS1.MANY system data sets. Installations can use the information to calculate charge backs for ABARS requests.

The new SETSYS ARECOVER TGTGDS(*option*) command provides greater flexibility for managing SMS-managed generation data sets that are being restored to level 0 DASD. You can specify the following options:

DEFERRED

Specifies that the target data set is to be assigned the DEFERRED status.

ACTIVE

Specifies that the target data set is to be assigned the ACTIVE status, for example, rolled into the GDG base.

ROLLEDOFF

Specifies that the target data set is to be assigned the ROLLEDOFF status.

SOURCE

Specifies that the target data set is to be assigned the same status it had when it was backed up.

The default option is SOURCE, which is consistent with prior releases of DFSMSHsm.

Installations are able to override the SETSYS ARECOVER TGTGDS(*option*) specification with a new ARECOVER command TGTGDS(*option*) keyword parameter. The options for ARECOVER TGTGDS are the same options as for SETSYS ARECOVER TGTGDS.

Note: If the SETSYS DEBUG parameter is specified before aggregate recovery processing, DFSMSHsm does not move any data. DEBUG causes all ARECOVER commands to be processed as if the VERIFY option is specified.

Related reading

- For more information about the ABARSPROCNAME parameter, see [“ABARSPROCNAME: Specifying the base procedure name to start the ABARS secondary address space”](#) on page 1096.
- For more information about the ABARSVOLCOUNT parameter, see [“ABARSVOLCOUNT: Specifying the method for determining volume count for SMS target data set allocations done by DFSMSdss”](#) on page 1097.

Example SETSYS commands for aggregate recovery

The following SETSYS commands are entered at the example site before performing aggregate recovery:

```
SETSYS MAXABARSADDRESSSPACE(2)
SETSYS ARECOVERUNITNAME(3490)
SETSYS ABARSPROCNAME(SAMPLE)
```

The maximum number of secondary address spaces is set to two, which means that two ABARS commands can run at a time. If the TARGETUNIT parameter is not specified when the ARECOVER command is issued, user tape data sets are recovered to 3490 tape units as specified by the ARECOVERUNITNAME(3490) parameter. The procedure name used to start the secondary address space is SAMPLE.

To review the SETSYS parameters in effect for aggregate processing, enter:

```
QUERY ABARS
```

or

```
QUERY SETSYS
```


Defining the model entity

If you specify the MENTITY parameter of the ARECOVER command, you must define a model entity. The model entity is used to define discrete data set profiles for RACF protection. The model entity is applicable only for level 0 DASD and migrated data sets that have been discretely RACF-protected at the backup site.

For more information on the MENTITY parameter, see [“ARECOVER command” on page 392](#).

Using the aggregate recovery installation exits

Programming Interface Information

If installation exits ARCSKEXT and ARCCREXT have been previously link-edited and if the SETSYS EXITON(ARCSKEXT) and SETSYS EXITON(ARCCREXT) commands have been issued (you can use the QUERY SETSYS command to display the status of these exits), ARCSKEXT and ARCCREXT are called during aggregate recovery verification. Installation exit ARCSKEXT is called for each data set to be recovered, allowing you to specify if that data set should be skipped or included in the aggregate recovery.

Installation exit ARCCREXT is called whenever a data set naming conflict is detected. This exit allows you to specify that the data set is to be:

- Skipped—The data sets are not recovered.
- Renamed—You give the source data sets on the data file new names. Data sets must not already exist with the new names you specify.

The rename process does not apply to migrated VSAM data sets. If you specify that a migrated VSAM data set should be renamed, it is skipped.

Note: DFSMSdss renaming of a VSAM sphere does not cause any alternate indexes to be renamed. If a data set naming conflict exists for alternate indexes, the alternate indices are not recovered.

- Replaced—The like-named data sets at the recovery site are replaced with the corresponding aggregate backup data sets. (This is the same result as if you had specified the DSCONFLICT(REPLACE) parameter on the ARECOVER command.)

Note: When using the DSCONFLICT(REPLACE) parameter, the data set is deleted before attempting the recovery of the data set from the aggregate backup output files. If the recovery fails, the data set remains deleted. Causes of recovery failure include:

- I/O errors on the input tape
- Failure to allocate the new data set
- Recovering at a site that does not support the type of data set being recovered
- Renamed target—The like-named target data sets at the recovery site are renamed using the new high-level qualifier specified. This is similar to the use of the DSCONFLICT RENAMETARGET(*level*) parameter.

If installation exit ARCTVEXT has been previously link-edited and if the SETSYS EXITON(ARCTVEXT) command has been issued (you can use the QUERY SETSYS command to display the status of this exit), ARCTVEXT is called during aggregate recovery. This exit is called when rolling off old ABARS versions to indicate to tape management systems, excluding DFSMSrmm, that ABARS is expiring a tape volume.

End Programming Interface Information

Related reading

For more information about the ARCCREXT and ARCTVEXT installation exits, see [z/OS DFSMS Installation Exits](#).

Processing generation data sets

The following topics discuss how backed-up generation data sets (GDSs) are restored during ARECOVER processing.

GDSs from level 0 DASD in the include list

Each generation data set (GDS) in the INCLUDE list that has been backed up from level 0 is recovered as a level 0 non-SMS-managed GDS with a status of ACTIVE if (1) ACS directs it to be recovered as not managed by SMS or if (2) SMS cannot manage the GDS because SMS is not active.

If the GDS is directed by ACS to be recovered as SMS-managed, it is recovered with the status specified by the SETSYS ARECOVER TGTGDS(*option*) command. The status can be specified as DEFERRED, ACTIVE, ROLLEDOFF, or SOURCE. SOURCE indicates that the SMS-managed data set being restored is to be assigned the same status it had when backed up. If the SETSYS ARECOVER TGTGDS command is not specified, the default is SOURCE.

If the ARECOVER command is issued with its optional TGTGDS(*option*) parameter, the value specified by the TGTGDS parameter overrides the value specified by the SETSYS ARECOVER TGTGDS command. The TGTGDS parameter, like the ARECOVER TGTGDS parameter, can be specified with the options of DEFERRED, ACTIVE, ROLLEDOFF, or SOURCE. [Table 43 on page 388](#) shows how level 0 GDSs are processed.

Table 43. Expected Results for Level 0 Data Sets in the INCLUDE List (Processed by DFDSS)

ABACKUP	ARECOVER			
	NON-SMS, NON-GDS	NON-SMS, ACTIVE GDS	SMS, Status as Specified or Default of SOURCE	SMS, NON-GDS
NON-SMS, NON-GDS	*			X
NON-SMS, ACTIVE GDS		*	X	
SMS, ACTIVE GDS		*	X	
SMS, ROLLED-OFF GDS		*	X	
SMS, DEFERRED GDS		*	X	
SMS, NON-GDS	*			X

* = Recovered as NON-SMS X = Recovered as SMS

Note: Non-SMS rolled-off GDSs are not processed.

Migrated GDSs in the include list

Non-SMS-managed GDSs in the INCLUDE list that were migrated data sets with a status of ACTIVE are always recovered as non-SMS-managed migrated GDSs, and retain a status of ACTIVE.

If SMS-managed migrated GDSs with ACTIVE status are directed by ACS to be recovered as non-SMS-managed or become non-SMS-managed because SMS is not active, they are recovered with a status of ACTIVE.

If an SMS-managed migrated GDS with a status of ACTIVE is directed by ACS to be recovered as SMS-managed, it is recovered with a status of ACTIVE.

SMS-managed migrated GDSs with a status of DEFERRED or ROLLED OFF that are directed by ACS to be recovered as SMS-managed maintain their original GDS status. If ACS directs these same data sets to be recovered as non-SMS-managed, or if they become non-SMS-managed because SMS is not active, they are recovered as uncataloged non-SMS GDSs, and DFSMSHsm issues message ARC6131I informing you of this status. You can issue the following Access Method Services command to catalog the generation data sets:

```
DEFINE NONVSAM(name(dsname) VOLUMES(MIGRAT) -
  DEVICETYPES(device) NORECATALOG)
```

The generation data sets are cataloged as non-SMS-managed GDSs with a status of ACTIVE. [Table 44 on page 389](#) shows how migrated GDSs are processed.

Table 44. Expected Results for Migrated Data Sets in the INCLUDE List

ABACKUP	ARECOVER						
	NON-SMS, NON-GDS	NON-SMS, ACTIVE GDS	NON-SMS, UNCATA- LOGED	SMS, ACTIVE GDS	SMS, ROLLED-OFF GDS	SMS, DEFERRED GDS	SMS, NON- GDS
NON-SMS, NON-GDS	*						
NON-SMS, ACTIVE GDS		*					
SMS, ACTIVE GDS		*		X			
SMS, ROLLED-OFF GDS			* (ARC6131I)		X		
SMS, DEFERRED GDS			* (ARC6131I)			X	
SMS, NON-GDS	*						X

* = Recovered as NON-SMS X = Recovered as SMS

Note: Non-SMS rolled-off GDSs are not processed.

When processing migrated data sets that are SMS-managed, ACS routines are invoked to determine if the data set should be recovered as SMS-managed or non-SMS-managed. If the ACS routines encounter a nonzero return code, they recover the migrated data sets that were SMS-managed at the backup site as non-SMS-managed data sets at the recovery site.

GDSs in the ALLOCATE list

Generation data sets in the ALLOCATE list are always recovered with a status of ACTIVE. You should not include SMS generation data sets with a status of ROLLED OFF or DEFERRED in the allocate list, because they will be recovered with a status of ACTIVE. This results in other GDSs with a status of ACTIVE being rolled off.

GDS processing order

The order in which the GDSs are processed determines which GDSs have an ACTIVE status after the recovery process has ended. The data sets in the INCLUDE list are processed first, and in the following order: the level 0 data sets are processed, followed by the migrated and user tape data sets. The data sets in the ALLOCATE list are processed next. The data sets in the ACCOMPANY list are processed last. If the limit is exceeded during the recovery process for a generation data group, then the oldest GDSs roll off.

Processing migrated data sets in the INCLUDE list

Migrated non-SMS-managed data sets in the INCLUDE list are always recovered as non-SMS-managed data sets. You can convert the data sets (with the exception of migrated GDSs) to SMS-managed data sets by recalling them with an ACS routine that does this conversion.

Processing VSAM data sets in the ALLOCATE list

During aggregate recovery, VSAM data sets from the ABACKUP ALLOCATE list must be recovered as SMS-managed data sets. Therefore, SMS must be active at the recovery site, and ACS filtering must direct these data sets to SMS-managed storage.

The following verification is performed during aggregate recovery. Error messages are issued for verification failures. If EXECUTE is specified on the ARECOVER command, only those data sets not receiving verification errors are recovered.

- If SMS is not active, an error message is issued that lists the VSAM data sets from the ALLOCATE list, indicating that these data sets will not be recovered.

- If SMS is active, ACS filtering is invoked to determine if the data set can be recovered as SMS-managed. If the data set will not be SMS-managed, or SMS errors occur, messages are issued, and the indicated data set is not recovered. The following ACS variables are passed to the ACS routines:
 - &ACSENVIR = ALLOC (data set recovery environment)
 - &DSN = VSAM cluster name. If the data set was renamed during ARECOVER processing, the new name is passed.
 - &DSTYPE = PERM (permanent data set)
 - &UNIT = Unit name of ABACKUP source data set
 - &DSORG = VS (VSAM data set organization)
 - &RECORG = ES, KS, RR, LS (VSAM type - ESDS, KSDS, RRDS, LINEAR)

If the ABACKUP source data set had associated SMS class names, these class names are passed as initial values to ACS routines. ACS may override these class names. If ACS does not override the SMS class name, and the name is not defined in the currently active configuration, an SMS ACS error occurs. The associated messages are written to the ARECOVER activity log, and verification fails.

 - &DATACLAS = source data class name
 - &MGMTCLAS = source management class name
 - &STORCLAS = source storage class name
- A data class should be defined and selected during aggregate recovery ACS filtering. This is particularly important when these data sets are allocated to a device with a different geometry than the source (that is, source=3380, target=3390). The source space allocation quantity is always recorded as an amount of cylinders or tracks, even if RECORDS was specified when the data set was defined. As a result, allocation to an unlike device may encounter over-allocation or under-allocation problems if the allocation quantity is not set by the data class.
- The storage group selected at the aggregate recovery site should contain enough volumes to support multiple volume allocation of those data sets that resided on multiple volumes at the aggregate backup site.
- Only base cluster information will be processed during ABACKUP for allocation and cataloging at the time of the ARECOVER. Users must define any alternate index (AIX) path associations at the ARECOVER site.

Note: If an AIX/PATH, as well as base cluster allocations, are needed at the time of the ARECOVER, place the entry in the INCLUDE list where the SPHERE processing is performed. When it is placed in the INCLUDE list, data for all components of the base cluster and the associations (AIX/PATH) are backed up and placed in the data file. This data file will be fully recovered during ARECOVER processing.

Catalog considerations during ARECOVER processing

There are at least three states that the catalog environment can be in at the recovery location:

1. The master and user catalogs exist at the recovery site with valid data set entries. You can issue the ARECOVER command with or without the DATASETCONFLICT(REPLACE) parameter. When you specify DATASETCONFLICT(REPLACE) parameter, be aware that you can unintentionally delete existing production data sets. If you do not specify DATASETCONFLICT(REPLACE), the data sets being recovered that have name conflicts are skipped during the recovery process.

Note: If a catalog entry exists for a VSAM RLS data set, existing attributes are maintained even if DSCONFLICT(REPLACE) was specified on the ARECOVER command. However, there is a PATCH that can be turned on to allow DFSMSHsm ABARS ARECOVER to use the backup version RLS attributes instead of the target's. See [Tuning DFSMSHsm](#) in *z/OS DFSMSHsm Implementation and Customization Guide*.

2. The Master and user catalogs exist at the recovery site but the data set entries are invalid. This means that either the volumes that are associated with a data set entry are not online or the volume does not contain a valid data set entry in the VTOC. This situation typically exists when catalogs have been

previously recovered as part of a full volume dump and restore of the operating system volumes containing these catalogs.

You can choose to issue the ARECOVER command with or without specifying the DATASETCONFLICT(REPLACE) parameter. If you do not specify DATASETCONFLICT(REPLACE), the data sets being recovered that have name conflicts are skipped during the recovery process. If you do specify DATASETCONFLICT(REPLACE), ARECOVER processing detects whether the catalog entry is valid or invalid. If the catalog entry is valid, a delete is issued for data sets in conflict before recovering the data set from the ABACKUP tapes. If the catalog entry is invalid, a DELETE NOSCRATCH is performed for data sets in conflict.

3. You can choose to DEFINE a new catalog and associated aliases before you issue the ARECOVER command. Doing so allows ARECOVER processing to catalog data set entries when data sets are being recovered, and avoids any data set naming conflicts.

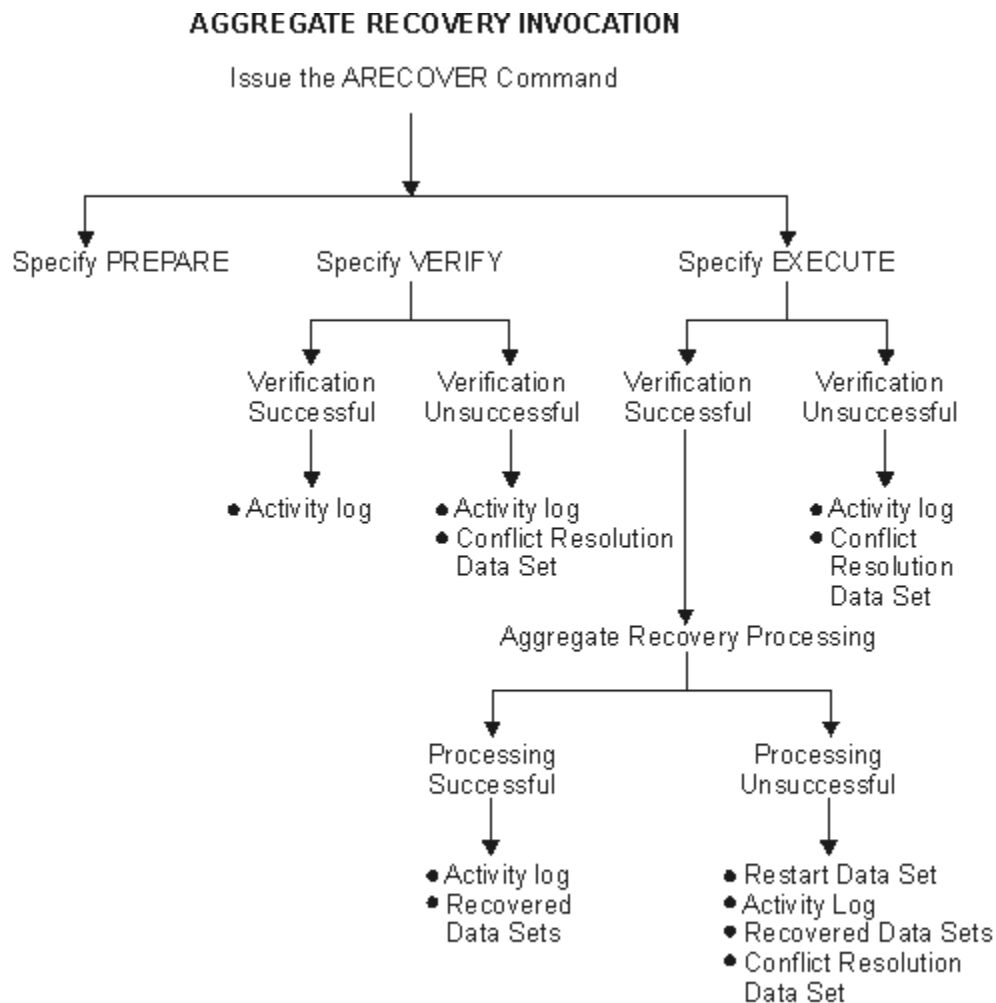
A situation can occur where migrated data sets in the catalog but the migration volume does not exist at the recovery site. ARECOVER processing deletes conflicting data sets when the DATASETCONFLICT(REPLACE) parameter is specified, and DFSMSHsm processing handles the situation where the migration volume is not online or the VTOC entry does not exist. If an ARC0184I message is issued when attempting to retrieve the MCV record for the migration volume, issue an ADDVOL command for the migration volume and reissue the ARECOVER command.

It is possible that the catalog has entries for migrated data sets, but the MCD record does not exist in the MCDS. ARECOVER processing detects this situation and issues a DELETE NOSCRATCH for the migrated data set if the DATASETCONFLICT(REPLACE) parameter was specified on the ARECOVER command. This allows the migrated data set to be deleted and aggregate recovery processing to successfully recover the migrated data set encountering this situation.

4. If you plan to recover the user catalogs specified in the ALLOCATE statement by using ARECOVER, and you have recovered the master catalog from the ABACKUP site that has alias entries for these user catalogs, it is best to first issue IDCAMS EXPORT DISCONNECT command before issuing the ARECOVER command to remove this connection.

Performing aggregate recovery

Figure 51 on page 392 shows what occurs when you invoke aggregate recovery by issuing the ARECOVER command.



ARC15G28

Figure 51. Aggregate Recovery Invocation

The following topics describe the aggregate recovery invocation in more detail.

ARECOVER command

Issue the ARECOVER command when you are ready to perform aggregate recovery.

You can specify the ARECOVER command with the PREPARE parameter to create the ABR record. You can specify the ARECOVER command with the VERIFY parameter to verify the accuracy of an aggregate recovery without actually having the data sets recovered. This is like performing a test run of aggregate recovery. After aggregate recovery verification processing completes, the ARECOVER command stops. No data sets are recovered. When you are ready to have aggregate recovery recover your data sets, you can issue the ARECOVER command and specify the EXECUTE parameter. After successful verification processing, your data sets are recovered.

Note: If SETSYS DEBUG is in effect, the VERIFY parameter is forced, regardless of other settings.

The following two topics describe in more detail what occurs when you issue the ARECOVER command with the EXECUTE parameter.

The DATASETNAME and VOLUMES | XMIT parameters are required if the ABR record was not previously created using the PREPARE parameter of the ARECOVER command. The AGGREGATE parameter is required if the DATASETNAME parameter is not used. Either the PREPARE, EXECUTE, or VERIFY parameters must be specified on the ARECOVER command.

For *dsname* in DATASETNAME(*dsname*), specify the data set name of the control file.

For *volser1,...volsern* in VOLUMES(*volser1,...volsern*), specify the serial numbers of the control files. The volume serial numbers must be specified in the same order as in message ARC6061I, issued during the aggregate backup.

When files are transferred by using a transmission program such as NetView FTP, issue the ARECOVER command using the XMIT parameter. This tells DFSMSHsm that the aggregate backup output has been transmitted. Do *not* use the VOLUMES and UNIT parameters on the ARECOVER command when XMIT is specified.

Note:

- Specify the XMIT parameter with the ARECOVER command only if all aggregate backup output files have been cataloged. For DFSMSHsm hosts that are running DFSMS Version 2 Release 10 or subsequent releases, only the control file needs to be cataloged before ARECOVER, as long as you are recovering the original ABACKUP files and as long as all of these files have been backed up to tape.
- If installations use a utility to convert their existing aggregate backup output files from one tape technology to another, ensure that the newly created tapes are cataloged at the recovery site with the newly created unit name and volume serial number, and use the XMIT parameter on the ARECOVER command. The XMIT parameter ensures that ARECOVER processing retrieves the allocation information for the ABARS files from the catalog rather than using the volser and unit stored on the control file tape.

If you choose to have any or all of your ABACKUP output files directed to DASD volumes, there are special precautions you must observe for successful recovery of the backed up data:

- The aggregate backup files must be properly cataloged and exist at the recovery site.
- You must use the XMIT and DATASETNAME parameters on your ARECOVER command.

For *agname* in AGGREGATE(*agname*), specify the aggregate group name. Do *not* use the AGGREGATE parameter if you use the DATASETNAME parameter.

The EXECUTE parameter allows the actual ARECOVER data movement to take place when the ARECOVER command is processed. Before you can use ARECOVER AGGREGATE, one of the following actions must occur:

- You must be performing aggregate recovery on the same system where ABACKUP has been issued, and where the ABR record still exists, or
- You must have previously issued the ARECOVER DATASETNAME(*dsname*) PREPARE command.

Specify the PREPARE parameter to build the ABARS CDS record (ABR record) at the recovery site if it does not already exist, without further verification and without data movement. The PREPARE parameter also catalogs ABACKUP output files and adds volumes to RACF, if required.

The optional ACTIVITY parameter specifies that the ABACKUP activity log is to be recovered.

The optional INSTRUCTION parameter specifies that the instruction data set is to be recovered.

When the ARECOVER command is issued with the VERIFY or PREPARE parameters, DFSMSHsm retrieves this information from the ABR record and issues messages during recovery processing indicating the amount of storage required to recover an aggregate group. The amount of storage required for level 0, ML1, and ML2 recovery is displayed separately, along with the combined total storage requirement. This information is maintained in the ABR record and is written to both the ARECOVER activity log and the FSR record.

Note:

1. If the aggregate recovery is being made from DFHSM V2.5.0 or V2.6.0 ABACKUP output tapes, no storage requirements are displayed.
2. When ARECOVER EXECUTE is issued, the storage requirements of data sets not recovered is written in the ARECOVER activity log. The amounts of level 0, ML1, and ML2 storage are kept separately, as well as the combined total storage requirement.

3. When ARECOVER VERIFY is issued, and DFSMSHsm determines that a RESTART data set exists, a message is written in the ARECOVER activity log indicating the storage requirements of data sets not successfully recovered in the previous pass. The amounts of level 0, ML1, and ML2 storage are kept separately, as well as the combined total storage requirement.

The optional, and mutually exclusive, STACK and NOSTACK parameters are only valid with the DATASETNAME parameter. They allow you to override the specified SETSYS ABARSTAPES value, or its default value of STACK. The option you select indicates to ARECOVER DATASETNAME processing whether the ABACKUP output you are going to recover is stacked. This information is needed before mounting the control file tape. When the control file tape is mounted, ARECOVER forward spaces to file sequence number 4 if the output is stacked, but does not if the output is not stacked.

The optional UNIT(*unittype*) parameter is only valid if the DATASETNAME and VOLUMES parameters are used. For *unittype*, specify the unit type to be allocated for the control file.

The DATE or VERSION optional parameters are valid only when the AGGREGATE parameter is used. For DATE(*yyyy/mm/dd*), specify the date of the aggregate group version to be recovered. *yyyy* is the 4-digit year, *mm* is the month (01–12), and *dd* is the day of the month (01–31). If more than one aggregate version has been created on the date you specify, the most current version is selected.

For *nnnn* in VERSION(*nnnn*), specify the specific version of the aggregate group to be recovered. If neither DATE nor VERSION is specified, the most current aggregate group version is recovered.

Use the optional DATASETCONFLICT parameter to specify how you want to handle data set name conflicts.

- The RENAMESOURCE(*level*) subparameter specifies that the ABACKUP data set be recovered with the new high-level qualifier specified by *level*.
- The RENAMETARGET(*level*) subparameter specifies that the data set at the recovery site be renamed with the high-level qualifier specified by *level*, and the ABACKUP data set be recovered with its original name.

RENAMETARGET(*level*) performs an ALTER NEWNAME for the existing data set name. Special care should be taken when you change the high-level qualifier so that the existing data set does not become uncataloged, which may occur when no catalog alias exists for the new high-level qualifier.

- The BYPASS subparameter specifies that the ABACKUP data set be skipped and not recovered.
- The REPLACE subparameter specifies that the data set at the recovery site be deleted and replaced by the ABACKUP data set.

Note:

1. When you use the DSCONFLICT(REPLACE) parameter, the data set is deleted before attempting the recovery of the data set from the aggregate backup output files. If the recovery fails, the data set remains deleted. Recovery failure results from the following causes:
 - I/O errors on the input tape
 - Failure to allocate the new data set
 - Recovering at a site that does not support the type of data set being recovered
2. The DATASETCONFLICT parameter may be specified in conjunction with the RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL parameters.
3. The DATASETCONFLICT parameter may be abbreviated DSCONFLICT, RENAMESOURCE may be abbreviated RENSRC, and RENAMETARGET may be abbreviated RENTGT.

If you want to specify a predefined model entity to be used for defining discrete profiles to RACF, use the optional MENTITY parameter. Only those data sets previously protected by discrete profiles and defined to RACF before aggregate backup are protected by using discrete profiles. If the model entity is specified in the ARECOVER command and is not predefined, aggregate recovery fails.



Attention: If MENTITY is not specified and data sets exist that have been discretely RACF-protected at the backup site, the data sets are recovered as RACF indicated, and a default discrete

profile is created. The profile owner is set to the high-level qualifier of the data set, and UACC is set to NONE.

If a RACF profile has been predefined for the data sets, they are recovered with this predefined profile.

Use the optional MIGRATEDDATA parameter to specify the migration level to be used when recovering migrated data sets:

- ML1 specifies that all migrated data sets are recovered to migration level 1 DASD (this is the default if you do not use MIGRATEDDATA).
- ML2 specifies that all migrated data sets are recovered to nonspecific tape volumes which are then ADDVOLed as ML2 volumes.

Note: You must specify the tape unit name to be used when recovering migrated data sets to nonspecific volumes by use of the ARECOVERML2UNIT parameter of the SETSYS command.

- SOURCELEVEL specifies that migrated data sets that have been ML1 at the ABACKUP site are recovered to ML1 DASD; migrated data sets that have been ML2 DASD or ML2 tape are recovered to nonspecific volumes which are then ADDVOLed as ML2 volumes.

Note: If the duplex tape option is specified for ML2 tapes, a TCN record is created to cause an automatic TAPECOPY during secondary space management. ARECOVER of ML2 data is not duplexed.

DFSMSHsm automatically creates MCL records for all data sets that are successfully recovered to ML1 DASD volumes unless directed otherwise. This causes these data sets to be backed up during automatic backup processing.

Note: MCL records are not created for data sets recovered to ML2 tape volumes.

The optional NOBACKUPMIGRATED parameter allows you to specify that recovered migrated data sets are *not* to be backed up during automatic backup. If you do not specify NOBACKUPMIGRATED, DFSMSHsm backs up recovered migrated data sets residing on ML1 DASD the next time automatic backup is run.

The optional ONLYDATASET parameter allows you to recover individual data sets from an aggregate backup version. You can specify a single fully qualified data set name using the NAME(*dsname*) subparameter of ONLYDATASET to recover a particular data set. To recover multiple data sets, you can supply the name of a data set that contains a list of data set names to be recovered on the LISTOFNAMES(*listdsname*) subparameter of ONLYDATASET. The LISTOFNAMES data set must be a sequential data set, fixed-block, record size of 80, and be cataloged. The LISTOFNAMES data set should contain one fully qualified data set name per record. The abbreviation for ONLYDATASET is ONLYDS and the abbreviation for LISTOFNAMES is LON.

The optional PERCENTUTILIZED(*nnn*) parameter allows you to change the percentage to which DFSMSHsm allows DFSMSdss to fill non-SMS DASD recovery volumes when recovering data sets to level 0 volumes. For *nnn*, substitute an integer number between 1 and 100. If SETSYS ARECOVERPERCENTUTILIZED has not been specified, the default value is 080 (80%). The value that you specify overrides any default value specified by the SETSYS ARECOVERPERCENTUTILIZED parameter.

The optional RECOVERNEWNAMEALL(*level*) parameter specifies that all data sets being recovered be renamed with the high-level qualifier specified by *level*.

The optional RECOVERNEWNAMELEVEL(*olevel1,nlevel1,...*) parameter specifies that all data sets that have a high-level qualifier *olevelx* are recovered with a new high-level qualifier *nlevelx*. The maximum number of *olevelx,nlevelx* qualifier pairs is 30.

The RECOVERNEWNAMEALL and RECOVERNEWNAMELEVEL parameters allow data sets to be renamed before verification checking for like-named conflicts. These parameters apply to data sets in the INCLUDE and ALLOCATE lists (except for user catalogs in the ALLOCATE list), but do not apply to ACCOMPANY data sets. In addition, migrated source VSAM data sets may not be renamed. If the rename criteria applies to a migrated source VSAM data set, a message is issued, and the data set is not renamed.

If the rename criteria applies to a nonmigrated source VSAM data set, the new component names are not generated by ARECOVER processing, but are generated by a) DFDSS if the data set is from the INCLUDE list, or b) SMS VSAM allocation if the data set is from the ALLOCATE list. As a result, conflict resolution checking does not check for new conflicts for individual components, since the new component names are unknown.

If RECOVERNEWNAMEALL is specified, a message is issued indicating that all data sets will be renamed with the specified level. If RECOVERNEWNAMELEVEL is specified, a message is issued listing the level pairs specified, and indicating that each data set matching the old level will be renamed to the corresponding new level. Each data set meeting the rename criteria is recovered with the new name during data movement. Like-named conflicts created by the new name are evaluated during the conflict resolution process.

If renaming a data set creates a data set name with more than 44 characters, the data set name is truncated to 44 characters, and if the last character is now a '.', it is also truncated. A message is issued indicating that truncation occurred and the resulting new name.

Programming Interface Information

Whether RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL is specified, if the ARECOVER installation exit ARCSKEXT (which allows any data set to be skipped) is active, it is called, before any conflict resolution, with the original data set name of each data set to be recovered. If a data set has been renamed by the RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL parameter, but was not selected to be skipped in ARCSKEXT by the original data set name, ARCSKEXT is called again with the new data set name. This allows a data set to be skipped based on either its original or new name.

End Programming Interface Information

The optional TARGETUNIT parameter is used to specify the unit type to be allocated to user tape data sets being recovered to a tape device. If TARGETUNIT is not specified, the unit type defaults to the current value of the SETSYS ARECOVERUNITNAME parameter.

ACS routines can override this parameter by assigning a tape storage group. If this is the case, the unit name specified by these parameters is ignored.

When you use the optional TGTGDS(*option*) parameter with the ARECOVER command, it overrides the value specified by the SETSYS ARECOVERTGTGDS command. The TGTGDS parameter allows you to specify DEFERRED, ACTIVE, ROLLEDOFF, or SOURCE for the SMS-managed generation data set that you are restoring to level 0 DASD. SOURCE causes the data set to be assigned the status it had at the time it was backed up.

You may find it helpful to use the following DFSMSHsm commands to manage and query an ARECOVER command:

- HOLD ARECOVER prevents all currently queued ARECOVER commands from being processed.
- RELEASE ARECOVER releases queued ARECOVER commands that have been held by using the HOLD command or by DFSMSHsm.
- STOP causes an orderly shutdown of DFSMSHsm after any currently running ABACKUP and ARECOVER commands complete processing.
- STOP DUMP causes an immediate shutdown of DFSMSHsm. Any ABACKUP and ARECOVER commands in process are ended and the secondary address space is released.
- QUERY or HQUERY displays information relevant to aggregate recovery.
- CANCEL or HCANCEL cancels any existing queued aggregate recovery.
- LIST displays information from existing ABR records.

Related reading

- For more information about the ABR record, see [“How to create the ABR record”](#) on page 733.
- For more information about the ABARS activity log, see [“Backing up and recovering the ABARS activity log”](#) on page 401.

- For more information about the ABARS Instruction data set, see [“Recovering the ABARS instruction data set” on page 400.](#)
- For more information about the ALTER NEWNAME function, see [z/OS DFSMS Access Method Services Commands.](#)
- For more information about conflict resolution, see [“ARECOVER data set name conflict resolution” on page 402.](#)
- For more information about the RACF profile, see [z/OS Security Server RACF Command Language Reference.](#)
- For more information about ABARS functions, see [“Holding and releasing ABARS functions” on page 408.](#)
- For more information about the ARECOVER command, see [Chapter 29, “ARECOVER command: Recovering a specified aggregate group,” on page 723.](#)

Examples of ARECOVER Commands

For the example site, the ARECOVER command is first issued with the PREPARE parameter:

```
ARECOVER DATASETNAME(PAY1.C.C01V0001) VOLUMES(TAPE23) -
UNIT(3490) PREPARE
```

Using the example control file, PAY1.C.C01V0001, aggregate recovery prepare is performed. The control file volume, TAPE23, is mounted on a 3490 tape unit as specified by the UNIT(3490) parameter. This action prepares the recovery site for subsequent ARECOVER processing using the aggregate group name.

The ARECOVER command is then issued with the VERIFY parameter to verify the recovery environment:

```
ARECOVER AGGREGATE(PAY1) VERIFY
```

The ARECOVER command is then issued with the EXECUTE parameter to invoke aggregate recovery processing:

```
ARECOVER AGGREGATE(PAY1) EXECUTE
```

Aggregate recovery processing is performed using the example site control file, PAY1.C.C01V0001, with a 3490 tape unit used for the control file volume, TAPE23. User tape data sets are recovered to the 3490 tape unit device type because the SETSYS ARECOVERUNITNAME(3490) command was issued previously.

The following INCLUDE data sets are recovered to DASD:

- PAY2.PY002.SOCSEC
- PAY2.PY002.STATUS.G0002V00
- PAY1.PY001.CHECK
- PAY1.PY002.IRA
- PAY1.PY001.BONUS
- PAY1.PY002.LADDER

The following INCLUDE data set is recovered to tape:

- PAY1.PY001.NAMES

The following ALLOCATE data set is allocated on a DASD:

- PAY2.PY001.BENEFIT

The following ACCOMPANY tape data set is cataloged:

- PAY2.PY001.TAXES

If the following command had been issued:


```
ARECOVER AGGREGATE(PAY1) TARGETUNIT(3400-6) EXECUTE
```

Data set PAY1.PY001.NAMES would have been recovered to a 3420 tape device. The ARECOVER TARGETUNIT parameter takes precedence over the SETSYS ARECOVERUNITNAME parameter.

If data sets were transmitted by using a transmit program such as NetView FTP, issue the following command:

```
ARECOVER DATASETNAME(PAY1.C.C01V0001) XMIT -  
TARGETUNIT(3400-6) EXECUTE
```

When the aggregate backup output files are transmitted, you must catalog the control and data files before issuing the ARECOVER command with the XMIT parameter. Transmission programs such as NetView FTP provide the option to have transmitted data sets automatically cataloged at the recovery site.

Issuing the ARECOVER command with the PREPARE parameter

To create an ABR record without the overhead of verification processing or data movement associated with the VERIFY or EXECUTE parameter, issue the ARECOVER command with the PREPARE parameter.

If you specify the PREPARE parameter and an ABR record already exists for the specified aggregate group version, and all the associated ARECOVER input files are properly cataloged, a message is issued, and the PREPARE function ends. If an ABR record already exists but one or more associated files are not cataloged, a message is issued for each file that is not cataloged, the ABR record is deleted (with an accompanying message), and the PREPARE function continues.

Once an ABR record is created, the control file, data files, and instruction/activity log files associated with the aggregate group version are cataloged, if previously uncataloged. The volumes containing these files are added to the RACF HSMABR TAPEVOL set, if necessary. Messages are issued indicating the catalog and RACF action taken.

Storage requirement messages are also written to the ARECOVER activity log during the PREPARE function.

Issuing the ARECOVER command with the VERIFY parameter

When you want only to verify the accuracy of your aggregate recovery without actually recovering the data sets, issue the ARECOVER command with the VERIFY parameter. This optional test run allows aggregate recovery to determine if any errors or data set naming conflicts exist. In the event of unresolved data set naming conflicts, aggregate recovery fails and a conflict resolution data set is created to help you resolve the conflicts. Remember, no data sets are recovered.

When performing verification processing, aggregate recovery calls installation exits ARCSKEXT and ARCCREXT, if requested and successfully loaded. ARCSKEXT allows you to specify if a data set should be skipped so that it is not included in the aggregate recovery. ARCCREXT allows you to specify how a data set naming conflict should be resolved.

If the data set naming conflicts are not resolved, the verification fails and a conflict resolution data set is created.

When processing completes, make sure that the ARECOVER command completed successfully with a return code of zero.

Related reading

For more information about the ARCSKEXT and ARCCREXT installation exits, see [“Using the aggregate recovery installation exits” on page 387](#) and [z/OS DFSMS Installation Exits](#).

Issuing the ARECOVER command with the EXECUTE parameter

When you are ready to have your data sets actually recovered, issue the ARECOVER command with the EXECUTE parameter. Aggregate recovery automatically performs preparation processing, if required, and verification processing, as described in [“Issuing the ARECOVER command with the VERIFY parameter” on page 398](#). The difference is that after verification processing completes successfully or has encountered *only* data set name conflicts, then aggregate recovery proceeds to recover your data sets. Aggregate recovery directs all messages to the aggregate recovery activity log.

When processing completes, determine if the ARECOVER command completed successfully with a return code of zero.

If the ARECOVER command has failed, as indicated by a nonzero return code, review the activity log for error messages. Correct any errors, and reissue the ARECOVER command. Repeat this process until the ARECOVER command completes successfully with a return code of zero. The restart data set, created when the ARECOVER command fails, is used when you restart the aggregate recovery. The restart data set ensures that data sets already recovered before the failure will not cause a data set conflict when you reissue the ARECOVER command.

Related reading

For more information about the files created by an aggregate recovery, see [“Aggregate recovery output” on page 411](#).

Restart data set

A restart data set is created when the ARECOVER command specified with the EXECUTE parameter does not complete successfully. The restart data set contains the names of all data sets successfully recovered. When you reissue the ARECOVER command, the restart data set ensures that aggregate recovery does not attempt to recover these previously recovered data sets. In this way, the restart data set avoids the possibility of numerous data set naming conflicts when you reissue an ARECOVER command that previously did not complete successfully. The format of the restart data set name is: *authid.agname.RESTART.Dyyddd.Thhmmss*

The variables in the restart data set name are explained:

authid

DFSMSHsm-authorized user ID.

agname

Aggregate group being processed.

yddd

Date that the specific aggregate backup for this aggregate recovery was performed.

hhmmss

Time that the specific aggregate backup for this aggregate recovery was performed.

Tip: The error that caused the aggregate recovery to fail may have prevented the restart data set from being closed and cataloged. If this happens, you must manually close and catalog the restart data set so that it is available to aggregate recovery when it is restarted. Message ARC6111I in the activity log identifies the restart data set name.

Issuing the ARECOVER command with the ONLYDATASET parameter

You can use the ARECOVER command to recover individual data sets from an aggregate backup version. You can specify a single fully qualified data set name on the ARECOVER command to recover a particular data set. To recover multiple data sets, you can supply the name of a data set that contains a list of data set names to be recovered from a specific aggregate backup version. Examples of each are provided in this topic.

As an alternative to the ARECOVER command, you can perform these tasks through ISMF panels, as described in [“Using ISMF to recover individual data sets from an aggregate backup” on page 426](#).

Recovering an individual data set from an aggregate backup

To recover one data set from an aggregate backup version, enter the ARECOVER command with the ONLYDATASET parameter specified with the NAME subparameter:

```
ARECOVER ONLYDS(NAME(MY.OWN.DATASET))
```

In this example, *MY.OWN.DATASET* is the fully qualified name of the data set to be recovered from the aggregate version.

Recovering a group of data sets from an aggregate backup

To recover a group of data sets from an aggregate backup version, do the following:

1. Create a data set to use as the list of data sets to be recovered during ARECOVER processing. The data set must be a sequential data set, fixed-block, with a record size of 80, and be cataloged.
2. Edit the data set with the names of data sets to be recovered during ARECOVER processing. Specify one fully qualified data set name per record.
3. Enter the ARECOVER command with the ONLYDATASET parameter specified with the LISTOFNAMES subparameter:

```
ARECOVER ONLYDS(LON(MY.DATASET.LIST))
```

In this example, *MY.DATASET.LIST* is the fully qualified name of the data set that contains a list of data sets to be recovered from the backup copy of an aggregate group.

Recovering the ABARS instruction data set

You can use the INSTRUCTION parameter instead of a DFSMSdss batch job to recover the ABARS instruction data set.

ABACKUP considerations

If an instruction data set is specified in the aggregate group, DFSMSdss is used to write the instruction data set and the activity log data set (if DASD logging is specified), onto one file on a tape (or tapes). To save the volume serial numbers of this file in the control file, the instruction and activity log files are written before the control file. For coexistence with prior releases and for usability, the control file is the first file on the tape. Because of this necessity, the instruction and activity log files are written on a separate tape from the control file.

If an instruction data set is not specified in the aggregate group, the instruction data set name is left blank in the ABR record. A “DUMMY” instruction/activity log file is not created.

ARECOVER considerations

Specifying INSTRUCTION with the EXECUTE, VERIFY, or PREPARE parameters causes recovery of the instruction data set, if it exists. If the instruction data set does not exist when INSTRUCTION is specified, an informational message is issued and ARECOVER processing continues.

When the INSTRUCTION parameter is specified and DFSMSShsm ascertains that an instruction data set has been backed up, the following instruction and activity log file information is determined:

- If the DATASETNAME and VOLUMES parameters are used with the ARECOVER command, the control file is read to determine which volume serial numbers the instruction/activity log file resides on.
- If the DATASETNAME and XMIT parameters are used with the ARECOVER command, a check is made to ensure that the instruction/activity log file is cataloged.

Note: If installations use a utility to convert their existing aggregate backup output files from one tape technology to another, ensure that the newly created tapes are catalogued at the recovery site with the newly created unit name and volume serial number, and use the XMIT parameter on the ARECOVER

command. The XMIT parameter ensures that ARECOVER processing retrieves the allocation information for the ABARS files from the catalog rather than using the volser and unit stored on the control file tape.

- If the parameter AGGREGATE is used with the ARECOVER command, a check is made to ensure that the instruction/activity log file is cataloged.

If the instruction data set exists on an output tape, DFSMSdss is invoked to perform the RESTORE of the instruction data set onto a level 0 volume from the ARPOOL. If the instruction data set currently exists, it is replaced.

If the recovery of the instruction data set fails, or if the INSTRUCTION parameter is specified and there is no instruction data set on the output tape, a message is issued and processing continues.

Backing up and recovering the ABARS activity log

Since there is useful information in the ABACKUP activity log that can be used at the aggregate recovery site, ABARS includes the ABACKUP activity log data set with the instruction data set as a new file on a separate output tape.

The SETSYS ABARSACTLOGTYPE parameter

The SETSYS command ABARSACTLOGTYPE ([SYSOUT[(*class*)] | DASD]) is provided to select the ABARS activity log output data set type. If SETSYS ABARSACTLOGTYPE is not issued, it defaults to the type specified in the SETSYS ACTLOGTYPE command.

- SYSOUT(*class*) specifies that messages are written to a spool data set. For *class*, substitute an alphanumeric character for the class ABARS uses for output. If *class* is not specified, the default is the SETSYS ACTLOGTYPE SYSOUT class. If ACTLOGTYPE has been set to DASD, the ABARSACTLOGTYPE class defaults to the SETSYS SYSOUT class, which in turn defaults to "A".
- DASD specifies that an output data set is dynamically created and cataloged.
- SYSOUT is the initial default setting. If SETSYS ABARSACTLOGTYPE has not been issued, each time SETSYS ACTLOGTYPE is issued it makes the corresponding change to the setting of ABARSACTLOGTYPE.

Note: Once SETSYS ABARSACTLOGTYPE is issued, any subsequent SETSYS ACTLOGTYPE command does not affect the ABARSACTLOGTYPE.

The DFSMSHsm QUERY command, with the SETSYS or ABARS parameter specified, can determine and display the value specified in the ABARSACTLOGTYPE(*type*) parameter.

Conflict avoidance and conflict resolution processing options

The RECOVERNEWNAMELEVEL(*olevel*,*nlevel*) and RECOVERNEWNAMEALL(*level*) parameters provide options that allow you to avoid known data set name conflicts when you issue the ARECOVER command. Specifying these parameters, however, does not guarantee that a data set name conflict for a newly generated name will not occur. If the use of either of these parameters results in a data set name conflict, then conflict resolution processing that involves the newly generated name is invoked as shown in the following table.

**If you specify RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL and Then
you have also specified**

DATASETCONFLICT(RENAMETARGET(*level*))

The data set that exists at the recovery site is renamed by using the high-level qualifier specified in the RENAMETARGET(*level*) parameter, and the ABACKUP data set is recovered using the newly generated name as specified by the RECOVERNEWNAME parameter.

If you specify RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL and Then
you have also specified

DATASETCONFLICT(RENAMESOURCE(<i>level</i>))	<p>The data set is recovered with the data set name generated by using the high-level qualifier specified with the RENAMESOURCE(<i>level</i>) parameter.</p> <p>In this case, the high-level qualifier specified with the RENAMESOURCE parameter will override the high-level qualifier of the data set name that was originally backed up, along with the high-level qualifier generated by the RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL parameters.</p>
DATASETCONFLICT(REPLACE)	<p>The existing data set at the recovery site is deleted, and the data set is recovered with a data set name that uses the high-level qualifier specified with the RECOVERNEWNAMELEVEL or RECOVERNEWNAMEALL parameters.</p>

ARECOVER data set name conflict resolution

DATASETCONFLICT parameter

The optional DATASETCONFLICT parameter of the ARECOVER command allows more flexibility for conflict resolution options. This parameter applies to INCLUDE, ALLOCATE, and ACCOMPANY data sets, with some restrictions. Like-named conflicts are evaluated based on the new name for any data sets renamed by RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL.

The following subparameters of DATASETCONFLICT specify the action to be taken when a like-named conflict is found during ARECOVER verification.

- If you specify RENAMESOURCE(*level*), the ABACKUP data set is recovered with the new high-level qualifier specified by *level*.
- If you specify RENAMETARGET(*level*), the existing data set at the recovery site is renamed with the specified high-level qualifier, and the ABACKUP data set is recovered with its original name.
RENAMETARGET(*level*) performs an ALTER NEWNAME for the existing data set name. Special care should be taken when you change the high-level qualifier so that the existing data set does not become uncataloged, which may occur when no catalog alias exists for the new high-level qualifier.
- If you specify REPLACE, the existing data set at the ARECOVER site is deleted, and the ABACKUP data set is recovered.
- If you specify BYPASS, the ABACKUP data set is skipped and not recovered.
- BYPASS
- RENAMETARGET (if target data set restrictions are met)
- RENAMESOURCE (not supported for ACCOMPANY data sets)

If one or more verification errors occur (and are not resolved), the data sets receiving the verification errors are not recovered, but (when EXECUTE has been specified) processing continues. This causes a RESTART data set to be created. Data sets that did not receive verification errors are recovered.

Related reading

- For more information about using the DSCONFLICT parameter on ARECOVER to allow the EATTR information from the ABACKUP site to be restored, see [“EAV considerations for recovery functions” on page 15](#).
- For more information about the ALTER NEWNAME function, see [z/OS DFSMS Access Method Services Commands](#).

Conflict resolution data set

The use of the IDCAMS command data set is no longer valid. A conflict resolution data set is generated by ARECOVER to simplify the resolution of like-named conflicts discovered during ARECOVER verification.

Each time ARECOVER verification processing cannot resolve a like-named conflict for a data set, the data set name is written to the conflict resolution data set with a default action of BYPASS. A message is issued indicating the name of the conflict resolution data set that was created. The name is generated as follows:
id.agname.CONFLICT.Dyyddd.Thhmmss

The variables in the conflict resolution data set name are explained:

id

DFSMSHsm authorized user ID.

agname

Name of the aggregate group being recovered.

yyddd

Julian date the ABACKUP was performed.

hhmmss

Time the ABACKUP was performed.

The aggregate group name, ABACKUP date, and ABACKUP time are used to make this conflict resolution data set unique to the recovery of a specific ABACKUP version.

An example of a conflict resolution data set is shown in [Figure 52 on page 403](#). It is a physical sequential data set with fixed-length 80 byte records.

```

/* CONFLICT RESOLUTION DATA SET                               */
/* AGGREGATE GROUP=agname, VERSION=nnnn                        */
/* SPECIFY datasetname action ON EACH LINE                    */
/* VALID ACTIONS ARE BYPASS, REPLACE,                         */
/* RENAME SOURCE(level) or RENAMETARGET(level)                */
DATASETNAME1 BYPASS
DATASETNAME2 BYPASS
:
```

Figure 52. Conflict Resolution Data Set

To indicate different conflict resolution actions taken by ARECOVER verification processing for each data set entry in the conflict resolution data set, you may edit the conflict resolution data set before issuing the next ARECOVER command. As shown in [Figure 52 on page 403](#), the format of one line of the conflict resolution data set is a data set name followed by the action to be taken. The syntax of the action is the same as that of the ARECOVER command DATASETCONFLICT subparameters. Valid actions are RENAME SOURCE(*level*), RENAMETARGET(*level*), REPLACE, and BYPASS. Not all actions are valid for all data sets, based on restrictions discussed in topic [“Order of processing for conflict resolution” on page 406](#).

When a subsequent ARECOVER command is issued and the conflict resolution data set exists, it is read and used as input to the verification process. If the conflict resolution data set does not meet format restrictions, a message is issued and ARECOVER processing continues, but ends following verification processing, even if EXECUTE has been specified. The conflict resolution data set is ignored during the verification process when this error occurs.

The lines containing conflict resolution actions are parsed for valid syntax, and data set names or qualifiers are also parsed for validity. If invalid syntax is found, or a data set name entry in the conflict resolution data set does not match the name of a data set entry in the ARECOVER control file, or a data set name entry is a duplicate of another entry, a message is issued and ARECOVER processing continues, ignoring the erroneous entries during verification processing.

When a like-named conflict is found for a data set name that has an entry in the conflict resolution data set, the action specified in the conflict resolution data set is used to attempt to resolve the conflict.

When VERIFY is specified, the action is not actually performed, but restrictions are validated. If the action in the conflict resolution data set is invalid for the data set, a message is issued indicating this, and if the conflict is not resolved by other conflict resolution processing, the data set fails verification. The verification process continues with the next data set.

When you use the ARECOVER command with the VERIFY parameter, a conflict resolution data set may be created. This conflict resolution data set contains entries for data sets that had a name conflict during verification processing, and a default action of BYPASS is indicated for these data sets. If you determine that the conflicts can be handled by either the RECOVERNEWNAMEALL, RECOVERNEWNAMELEVEL, or DATASETCONFLICT parameters on a subsequent ARECOVER command, then you delete the conflict resolution data set before issuing the command. If this deletion is not done, ARECOVER processing uses the conflict resolution data set action of BYPASS, and the data sets are skipped.

When EXECUTE is specified, the conflict resolution action is validated. If valid, the action is performed during data movement. If the action in the conflict resolution data set is invalid for the data set, a message is issued indicating this. If the conflict is not resolved by other conflict resolution processing, the data set fails verification, and is not recovered. ARECOVER EXECUTE processing continues for the remaining data sets. If an error occurs while performing the action during EXECUTE processing, a message is issued, and the data set is bypassed and not recovered.

When EXECUTE is specified, if all conflicts are resolved, and if all data sets in the aggregate group have been recovered—not bypassed through use of the conflict resolution data set—the conflict resolution data set and the RESTART data set are deleted. Data sets which are bypassed by a conflict resolution data set action of BYPASS cause the RESTART data set and conflict resolution data set to be retained and a unique return code to be set. The bypass of these data sets may be intentional, but since the default action placed in the conflict resolution data set is BYPASS, and the default is also to use the conflict resolution data set if it exists, the data sets may be bypassed unintentionally. In this case, deletion of the conflict resolution data set and RESTART data set would make the recovery of the remaining data sets difficult. If your analysis at this point shows that all desired data sets have been recovered, you may then manually delete the RESTART data set and conflict resolution data set.

If the conflict cannot be resolved, an entry is added to the conflict resolution data set indicating a default action of BYPASS (if there is not an entry already, no duplicate entries are created and the action for a current entry is NEVER modified). The data set is not recovered and processing continues. An entry is *only* added to the conflict resolution data set if the conflict cannot be resolved by *any* of the preceding actions.

If the conflict has been found for a component of a VSAM data set being verified, the source VSAM cluster name is added to the conflict resolution data set (not the individual component names).

If the data set has been renamed before verification by a RECOVERNEWNAMEALL or RECOVERNEWNAMELEVEL subparameter, the original data set name is used for the entry in the conflict resolution data set.

Each data set successfully verified is recovered if the EXECUTE parameter is specified.

The following conflict resolution actions are supported.

- RENAMESOURCE(*level*)

Verification checks to see if a data set is already cataloged with the new name at the recovery site. If the source data set is a GDS, neither a GDS generation nor a related GDG base may exist for the new name (a unique message is issued indicating GDS/GDG base conflicts). Before recovery of a renamed GDS generation, a GDG base is defined with the proper new name by using the attributes of the source GDG base. If no like-named conflict is found for the new data set name, the data set is renamed when it is recovered with the new high-level qualifier.

If the new name would cause a like-named conflict, an appropriate message is issued and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

The RENAMESOURCE subparameter is not supported for ACCOMPANY data sets, or for migrated VSAM source data sets. In each case, a message is issued and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

- RENAMETARGET(*level*)

Verification checks the attributes of the existing data set cataloged at the recovery site. If the target data set is one of the following, a message is issued, the target data set is not renamed, and an entry in the conflict resolution data set for the original data set name is created, indicating the default action of BYPASS.

- tape data set
- catalog
- GDS
- GDG base

In addition, the target data set **MUST** be one of the following to be eligible for rename:

- non-VSAM DASD
- VSAM base cluster
- VSAM alternate index (AIX)
- Data component, index component, or path related to a VSAM cluster or AIX (but not related to a catalog)

If the target object in conflict is a component or path related to a VSAM cluster or AIX, the entire target cluster, including component associations, is renamed.

Verification checks to see if a data set is already cataloged at the recovery site with the new name. If so, an appropriate message is issued, the target data set is not renamed, and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

If the data set meets restrictions for renaming and EXECUTE is specified, the existing data set at the recovery site is renamed. A message is issued indicating the action taken. When the conflict occurs for a component of a target VSAM data set, the entire target cluster, including associated components, is renamed. The indicated new high-level qualifier is used to rename the target objects. The ABACKUP data set is recovered with the original name. When the target data set is migrated, the RENAMETARGET action causes it to be recalled. If the rename action fails, a message is issued, and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

RENAMETARGET(*level*) performs an ALTER NEWNAME for the existing data set name. Special care should be taken when you change the high-level qualifier so that the existing data set does not become uncataloged, which may occur when no catalog alias exists for the new high-level qualifier.

- BYPASS

The data set is not recovered during data movement. When a conflict occurs for a component of a source VSAM data set, the entire source VSAM cluster is bypassed and not recovered. A message is issued indicating the action taken.

- REPLACE

Verification checks to see if the existing data set at the recovery site meets restrictions for data sets that may be deleted. If restrictions are not met, a message is issued, and the data set is not recovered. If the data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

Note: When using the DSCONFLICT(REPLACE) parameter, the data set is deleted before attempting the recovery of the data set from the aggregate backup output files. If the recovery fails, the data set remains deleted. Causes of recovery failure include:

- I/O errors on the input tape
- Failure to allocate the new data set
- Recovering at a site that does not support the type of data set that is being recovered

If a GDG base exists and DSCONFLICT(REPLACE) has not been specified, recovery of the associated GDSs fails and an entry is placed in the conflict resolution data set with a default action of BYPASS.

If restrictions are met and EXECUTE is specified, the data set at the recovery site is deleted. When a conflict occurs for a component of a source VSAM data set, the entire target VSAM cluster related to the object in conflict is deleted. A message is issued indicating the action taken. If the delete for the REPLACE subparameter fails, a message is issued and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

ARCCREXT installation exit

Programming Interface Information

If the like-named conflict has not yet been resolved, and the installation exit ARCCREXT is active, the exit is called to attempt to resolve the conflict. If a conflict is found for a component of a source VSAM cluster being verified, only the source VSAM cluster name is passed to the installation exit. The action specified for the VSAM cluster name is used to resolve the conflict for the entire related VSAM cluster.

The installation exit passes back a return code to indicate what action to take to resolve the conflict. If the installation exit returns an invalid action for the data set, a message is issued and the data set is not recovered. If the original data set name is not already in the conflict resolution data set, an entry is created indicating the default action of BYPASS.

Return codes indicate the following actions:

- 0 = no action
- 4 = bypass the data set
- 8 = replace the existing data set
- 12 = rename the source data set (the exit specifies the new name)
- 16 = rename the existing data set at the recovery site with the returned new high-level qualifier before recovery. The source data set is recovered with the original name. This action is comparable to the DATASETCONFLICT RENAMETARGET(level) parameter.

End Programming Interface Information

Migrated non-VSAM data sets can be renamed if only the high-level qualifier of the data set name is changed. Using this exit to rename more than just the data set's high-level qualifier causes an error (message ARC6325E) and the data set will not be processed during ARECOVER.

After conflict resolution processing for all data sets in the control file during verification and before actual data movement, a check is done to verify that no data sets to be recovered are renamed to the same name as another data set entry in the control file. When a data set name is renamed to the same name as an original source data set also being recovered, it is not renamed or recovered. When two or more data sets are renamed to the same name, only the first one is renamed and recovered. Error messages are issued for the other data sets that do not get recovered.

If all verification errors are resolved during ARECOVER EXECUTE processing, the conflict resolution data set is deleted.

DFSMSHsm allows a single ARECOVER to invoke DFSMSdss multiple times to RESTORE all required data sets. Two hundred fifty-five data sets are renamed per invocation of DFSMSdss, until all data sets have been recovered.

Order of processing for conflict resolution

A like-named conflict is identified when a catalog entry exists at the recovery site with the same name as the data set being recovered.

During like-named conflict checking, if the data set being recovered is a VSAM data set, each component of the source VSAM data set is used to locate a like-named conflict. The action taken to resolve the conflict is discussed below under each conflict resolution subparameter.

If a like-named conflict is detected, the conflict is resolved by the following order of conflict resolution actions:

1. If a conflict resolution data set exists and contains an entry for the data set in conflict, the requested action is validated.

If valid, a message is issued indicating an action to take to resolve the conflict during EXECUTE processing. If the action is invalid, a message is issued indicating this, and conflict resolution processing is continued for the data set.

If the conflict was found for a component of a VSAM data set, an entry must be found in the conflict resolution data set for the related source VSAM cluster name. If found, the action specified for the related source VSAM cluster name is used to resolve the conflict for the cluster and all the components.
2. If the conflict is not yet resolved, and the DATASETCONFLICT parameter is specified on the ARECOVER command, the requested action is validated.

If valid, a message is issued indicating an action to take to resolve the conflict during EXECUTE processing. If the action is invalid, a message is issued indicating this, and conflict resolution processing is continued for the data set.
3. If the conflict is not yet resolved and the installation exit ARCCREXT is active, the exit is called to resolve the conflict. The requested action is validated. If valid, a message is issued indicating an action to take to resolve the conflict during EXECUTE processing. If the action is invalid, a message is issued indicating this.

RACF facility class authorization checking

RACF facility class checking supports the DATASETCONFLICT subparameters of REPLACE and RENAMETARGET. When either of these actions is specified through the installation exit ARCCREXT, or through the conflict resolution data set, the user must have the same facility class authorization as when these subparameters are specified on the ARECOVER command.

When facility class authorization is denied for either REPLACE or RENAMETARGET specified as a subparameter of the DATASETCONFLICT parameter on the ARECOVER command, ARECOVER processing fails and a message is issued.

When facility class authorization is denied for REPLACE or RENAMETARGET specified as an action in the conflict resolution data set or in the installation exit ARCCREXT, ARECOVER verification processing fails for the individual data set, and a message is issued. Other data sets not receiving verification errors are processed as ARECOVER continues.

RACF data set authorization is checked before processing the REPLACE or RENAMETARGET action for individual data sets during ARECOVER verification processing, to obtain further security if restricted facility class authorization is indicated. Restricted facility class authorization is indicated when the user has READ or greater access to the following facility class profiles (where applicable):

- STGADMIN.ARC.ARECOVER.agname.REPLACE
- STGADMIN.ARC.ARECOVER.agname.RENTGT

When restricted facility class authorization is indicated, RACF ALTER access authority is required for individual data sets when conflict resolution indicates an action of REPLACE or RENAMETARGET for the existing data set at the ARECOVER site. If RACF access authority is denied, a message is issued and the data set fails verification processing, unless another conflict resolution action (BYPASS or RENAMESOURCE) is selected through the normal order of conflict resolution processing. Remaining data sets are verified and recovered if verification is successful.

The hierarchy of facility class command authorization is determined as follows:

1. Facility class must be active. If not active, the ARECOVER command fails if the user is not DFSMSshm-authorized.
2. If facility class is active but the appropriate profiles are not defined, processing continues as if facility class were inactive. The ARECOVER command fails if the user is not DFSMSshm-authorized.

3. ARECOVER processing checks for restricted authority. Restricted command authority is granted when the user is authorized to the following facility class profiles with an access authority of READ or greater.
 - STGADMIN.ARC.ARECOVER.agname
 - STGADMIN.ARC.ARECOVER.agname.REPLACE (for REPLACE)
 - STGADMIN.ARC.ARECOVER.agname.RENTGT (for RENAMETARGET)

When restricted command authority is indicated, RACF data set authorization checking is performed each time a conflict is detected during ARECOVER verification processing, and a conflict resolution action of REPLACE or RENAMETARGET is selected. The user must have ALTER access to the data set to allow the conflict resolution action to be taken.

4. If restricted authority is not granted, ARECOVER processing checks for comprehensive command authority. Comprehensive authority is granted when a user is authorized to the STGADMIN.ARC.ARECOVER Facility Class profile, with an access authority of READ or greater.

When comprehensive authority is granted, RACF data set authorization checking is not performed during ARECOVER processing.

Holding and releasing ABARS functions

The DFSMSHsm HOLD and RELEASE commands can be used to stop, hold, and release certain aggregate backup and aggregate recovery functions.

Related reading

- For more information about the HOLD command, see [Chapter 50, “HOLD command: Preventing DFSMSHsm functions from running,” on page 895.](#)
- For more information about the RELEASE command, see [Chapter 62, “RELEASE command: Releasing all or some DFSMSHsm functions for processing,” on page 1029.](#)

Holding ABARS functions

You can use the DFSMSHsm HOLD command to hold ABARS functions and to designate a specific aggregate group or control file data set name to be held. The commands HOLD ABACKUP or HOLD ARECOVER may be used to hold ABARS functions, and the commands HOLD ABACKUP(*agname*) or HOLD ARECOVER [DATASETNAME(*controlfiledsn*) | AGGREGATE(*agname*)] may be used to hold a specific aggregate group or control file within the ABARS functions.

If the HOLD EOD option is specified for ABACKUP or ARECOVER, the following processing takes place:

Note: If an aggregate group or control file data set name is specified, this processing takes place only when the specified name matches a currently running or pending ABACKUP or ARECOVER command.

- Currently executing ABACKUP or ARECOVER commands are stopped, at the end of processing a data set, and are not requeued for processing. Data sets and tapes are closed and deallocated, all memory is freed, and processing is immediately terminated.
- If the HOLD EOD option is specified for a currently running ABACKUP command, the ABACKUP output files are uncataloged and the tape volume serial numbers are removed from the HSMABR tape connected set if they are indicated to be RACF protected.
- Pending ABACKUP or ARECOVER commands that were submitted without using the WAIT option are held. They remain in the queue and are not executed until a RELEASE command is issued.
- Pending ABACKUP or ARECOVER commands submitted using the WAIT option are failed with an accompanying error message.

If the HOLD command is specified for the ABACKUP or ARECOVER function (without the EOD option), the following processing takes place:

Note: If an aggregate group or control file data set name is specified, this processing takes place only when the specified name matches a currently running or pending ABACKUP or ARECOVER command.

- Currently executing ABACKUP or ARECOVER commands run to completion. If an aggregate group or control file data set name has been specified, subsequent commands for that name are held.
- Pending ABACKUP or ARECOVER commands submitted without using the WAIT option are held. They remain in the queue and are not executed until a RELEASE command is issued.
- Pending ABACKUP or ARECOVER commands submitted using the WAIT option are failed with an accompanying error message.

When the HOLD command is specified for an ABACKUP or ARECOVER command using an aggregate group or control file data set name and the command is neither running nor pending, subsequent commands issued using that name are held.

Note: During ABACKUP processing, if the control file has already been written and a HOLD EOD is issued, ABACKUP completes normally. However, if ABACKUP has been issued with the MOVE option, ABACKUP will be held.

Releasing held ABARS functions

The RELEASE command, used with the ALL, ABACKUP, or ARECOVER parameters, releases a held ABARS function. The RELEASE command also can be used to specify the aggregate group name for ABACKUP and the aggregate group or control file name with the (ARECOVER AGGREGATE | ARECOVER DATASETNAME) parameter. The previously held ABARS function is released.

Querying held ABARS functions

You can use the QUERY ACTIVE command to perform the following tasks:

- List active ABACKUP or ARECOVER commands with their associated aggregate group or control file name
- Indicate if ABARS is active or inactive
- Indicate whether ABACKUP and ARECOVER is held, not held, or held EOD
- Indicate the specific ABACKUP(*agname*) and ARECOVER [AGGREGATE(*agname*) | DATASETNAME(*controlfiledsn*)] commands that are held

Displaying SMS class definitions during ABACKUP and ARECOVER

During DFSMSHsm ABACKUP processing, a list of SMS construct names—along with individual class definitions for the SMS data, management, and storage classes associated with the data sets in the aggregate group being processed—are written to the control file.

During subsequent ARECOVER processing, a message is issued to the ARECOVER activity log listing the attributes of each SMS data class, management class, and storage class definition present at the ABACKUP site. This allows the recovery site to make any necessary changes to either existing SMS constructs or the SMS environment.

If, however, it is determined that the ABACKUP activity log has been recovered by the ARECOVER command specified with the ACTIVITY parameter (see [“Backing up and recovering the ABARS activity log” on page 401](#)), the SMS construct definitions are not listed in the ARECOVER activity log.

Note:

1. Use the VERIFY or PREPARE option of the ARECOVER command before the EXECUTE option. By doing this, you can determine SMS class definitions, and any needed changes can be made to the SMS environment before actual data movement.
2. Use the ACTIVITY option for ARECOVER as an alternative method of retrieving the SMS construct information (see [“Backing up and recovering the ABARS activity log” on page 401](#)). By using this method, you can obtain SMS construct information from the ABACKUP activity log, if DASD logging has been specified.

ABARS user catalog and catalog alias ARECOVER support

If user catalogs have been specified in the allocate list during aggregate backup, the catalog data and alias definitions are stored in the control file and used by DFSMSHsm during aggregate recovery to define the catalog and its aliases at the recovery location.

The RECOVERNEWNAMEALL and RECOVERNEWNAMELEVEL parameters do not apply to user catalogs in the ALLOCATE list. This restriction prevents conflicts in defining an alias that is the same as the new high-level qualifier.

If the catalogs specified in the ALLOCATE list have conflicts with existing data set names, aggregate recovery verification processing must ensure that these conflicts are resolved with either the conflict resolution data set, the DATASETCONFLICT parameter of the ARECOVER command, or through the invocation of the conflict resolution installation exit ARCCREXT.

All conflict resolution actions are supported to resolve a like-named conflict for a catalog to be recovered, provided the restrictions for each action are met, as discussed in [“ARECOVER data set name conflict resolution” on page 402](#). (The most notable restriction is that RENAMETARGET and REPLACE are not allowed when the existing data set is a catalog.)

If all catalog name conflicts are not resolved, verification fails. An error message is issued for each unresolved catalog conflict. A default entry (BYPASS) is *not* added to the conflict resolution data set, since a catalog recovery should not be bypassed as the default. (The user may, however, manually add an entry in the conflict resolution data set to resolve the catalog name conflict.) The remaining verification is completed, and the ARECOVER processing ends. No further processing takes place, even if EXECUTE was specified.

The alias definitions associated with the user catalogs in the ALLOCATE list must be checked against the current catalog structure at the aggregate recovery site, to ensure the alias is not already defined to an existing catalog. If the alias is already defined, an error message is issued during aggregate recovery verification processing. This error message indicates to which catalog this alias is already defined. Aggregate recovery verification is not failed, but the particular alias definition is not issued.

If any user catalogs have been renamed during conflict resolution RENAMESOURCE processing, the new high-level qualifier must be checked against any aliases not yet defined to ensure that no like-named conflicts exist. If conflicts exist, a message is issued and the renamed user catalog is not recovered unless the conflict can be resolved through means other than RENAMESOURCE.

The DEFINE of the user catalogs in the ALLOCATE list and the DEFINE for the associated ALIASES are performed before the data sets for the particular aggregate are recovered. This allows the data sets in the aggregate being recovered to be cataloged in these catalogs if the data sets have an alias defined to one of these catalogs.

Note: ABARS does not allow multiple defines of aliases at the recovery location. If an alias is defined to a catalog being recovered and a catalog already in existence, the catalog in existence takes precedence.

The aggregate recovery verification step lists the user catalogs contained in the control file and all of the aliases defined to those catalogs. This action is performed when the user specifies the ARECOVER command with the VERIFY parameter.

Data sets that are dynamically allocated during aggregate recovery processing

In addition to the RESTART and conflict resolution data sets, DFSMSHsm dynamically allocates the following data sets during aggregate recovery processing. These data sets are used while aggregate recovery is executing and are dynamically deallocated before completion of aggregate recovery processing. These data sets are allocated with a unit type of SYSALLDA. (If they are non-SMS-managed, you must have at least one STORAGE- or PUBLIC-type device online to use the system, and if they are SMS-managed, your ACS routine must assign them to a storage group that contains storage-type volumes.) You need to be aware of these data set names so that the high-level qualifiers used by ABARS can be defined to the system. You may want to modify your ACS filters to direct allocation of these data sets to specific volumes.

SYSIN data set

The SYSIN data set is used to pass RESTORE parameters to DFSMSdss. The SYSIN data set name is: *authid.SYSIN.agname.Thhmmss*

The variables in the data set name are explained:

authid

DFSMSHsm-authorized user ID.

agname

Aggregate group name being processed.

hhmmss

Hours, minutes, and seconds.

FILTER data set

The FILTER data set is used to pass a list of data set names to DFSMSdss. The FILTER data set name is: *authid.FILTER.agname.Thhmmss*

The variables in the data set name are explained:

authid

DFSMSHsm-authorized user ID.

agname

Aggregate group name being processed.

hhmmss

Hours, minutes, and seconds.

Aggregate recovery output

The activity log, recovered data sets, FSR records, updated catalog entries, GDG base definitions, and updated DFSMSHsm CDS records are the output of aggregate recovery.

The conflict resolution data set and the restart data set are the output of the ARECOVER command when data set conflicts or errors occur. These have been described in previous topics. See [“Restart data set” on page 399](#).

Activity log

An activity log is created each time you issue the ARECOVER command. You may find the activity log useful because it contains messages output during aggregate recovery processing.

The format of the activity log data set name is:

mcvtactn.Hmcvthost.ARECOVER.agname.Dyyddd.Thhmmss

The parts in the data set name are explained:

mcvtactn

Field in the MCVT that contains the value stored for the activity log high-level qualifier. The default value is HSMACT.

Hmcvthost

DFSMSHsm host identification.

ARECOVER

Constant for aggregate recovery.

agname

Aggregate group being processed.

Dyyddd

The date.

Thhmmss

The time.

The activity log is written to either a SYSOUT or DASD data set as determined by the ABARSACTLOGTYPE parameter of the SETSYS command. If you specify DASD and the allocation attempt fails, an attempt is made to allocate the activity log to SYSOUT. See [“Defining SETSYS parameters for aggregate recovery” on page 384](#) for more information on setting the activity log output type.

Here is a partial example of the aggregate recovery activity log:

```
PAGE 0001 DFSMSHSM 1.9.0 DATA FACILITY HIERARCHICAL STORAGE MANAGER 05.350 18:16
ARC6102I AGGREGATE RECOVERY STARTING USING CONTROL FILE DATA SET PAY1.C.C01V0002, AT 18:16:00,
  STARTED TASK = DFHSMABR.ABAR0154
ARC6030I ACTIVITY LOG FOR CONTROL FILE DATA SET PAY1.C.C01V0002 WILL BE ROUTED TO SYSOUT=A
ARC6115I AGGREGATE RECOVERY USING CONTROL FILE DATA SET PAY1.C.C01V0002 WILL USE VOLUMES
A00020
A00016
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY001.NAMES DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.RETIRE DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.SOCSEC DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.STATUS.G0002V00 DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.INSTRUCT DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY001.CHECK DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY001.PGMLIB DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY002.LADDER DURING
AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY002.IRA DURING
AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY001.NAMES HAS BEEN UNCATALOGED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.RETIRE HAS BEEN UNCATALOGED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.SOCSEC HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.INSTRUCT HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY001.CHECK HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY001.PGMLIB HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY002.LADDER HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY002.IRA HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6004I 000I ARECOVER PAGE 0001 5695-DF175 DFSMSdss Data Set Services 97.238 18:16
ARC6004I 000I ARECOVER ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC6004I 000I ARECOVER RESTORE DATASET(FILTERDD(SYS00016)) -
ARC6004I 000I ARECOVER INDDNAME(SYS00015) -
ARC6004I 000I ARECOVER OUTOYNAM( -
ARC6004I 000I ARECOVER (DUMPDS) -
ARC6004I 000I ARECOVER ) -
ARC6004I 000I ARECOVER PERCENTUTILIZED( -
ARC6004I 000I ARECOVER 080 -
ARC6004I 000I ARECOVER ) -
ARC6004I 000I ARECOVER SPHERE -
ARC6004I 000I ARECOVER TGTGDS(SOURCE) -

PAGE 0002 DFSMSHsm DATA FACILITY HIERARCHICAL STORAGE MANAGER 05.350 18:16
ARC6004I 000I ARECOVER CATALOG FORCE
ARC6004I 000I ARECOVER ADR101I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
ARC6004I 000I ARECOVER ADR109I RI01 (01), 97238 18:16:41 INITIAL SCAN OF USER CONTROL STATEMENTS
COMPLETED.
ARC6004I 000I ARECOVER ADR006I (001)-SETUP(01), 97238 18:16:42 EXECUTION BEGINS
ARC6004I 000I ARECOVER ADR780I (001)-TDOS (01), THE INPUT DUMP DATA SET BEING PROCESSED IS IN LOGICAL
DATA SET FORMAT
ARC6004I 000I ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY2.PY002.SOCSEC ALLOCATED, ON VOLUME(S): DUMPDS
ARC6004I 000I ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY2.PY002.SOCSEC HAS BEEN CATALOGED I
IN CATALOG TPCMAST1
ARC6004I 000I ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY2.PY002.SOCSEC WAS RESTORED
ARC6004I 000I ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY2.PY002.STATUS.G0002V00 ALLOCATED, ON
VOLUME(S): DUMPDS
ARC6004I 000I ARECOVER ADR465I (001)-DALOC(01),
ARC6004I 000I ARECOVER DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN CATALOGED IN CATALOG TPCMAST1
ARC6004I 000I ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY2.PY002.STATUS.G0002V00 WAS RESTORED
ARC6004I 000I ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY1.INSTRUCT ALLOCATED, ON VOLUME(S): DUMPDS
ARC6004I 000I ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY1.INSTRUCT HAS BEEN CATALOGED IN CATALOG TPCMAST1
ARC6004I 000I ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY1.INSTRUCT WAS RESTORED
ARC6004I 000I ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY1.PY001.CHECK ALLOCATED, ON VOLUME(S): DUMPDS
ARC6004I 000I ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY1.PY001.CHECK HAS BEEN CATALOGED IN CATALOG
TPCMAST1
ARC6004I 000I ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY1.PY001.CHECK WAS RESTORED
ARC6004I 000I ARECOVER ADR396I (001)-NEWDS(01), DATA SET PAY1.PY001.PGMLIB ALLOCATED, ON VOLUME(S): DUMPDS
ARC6004I 000I ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY1.PY001.PGMLIB HAS BEEN CATALOGED IN CATALOG
TPCMAST1
ARC6004I 000I ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY1.PY001.PGMLIB WAS RESTORED
ARC6004I 000I ARECOVER ADR454I (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
PAY2.PY002.SOCSEC
PAY2.PY002.STATUS.G0002V00
PAY1.INSTRUCT
ARC6004I 000I ARECOVER PAGE 0002 5695-DF175 DFSMSdss Data Set Services 97.238 18:16
ARC6004I 000I ARECOVER PAY1.PY001.CHECK
ARC6004I 000I ARECOVER PAY1.PY001.PGMLIB
ARC6004I 000I ARECOVER ADR013I (001)-CLTSK(01), 97238 18:17:00 TASK COMPLETED WITH RETURN CODE 0000
ARC6004I 000I ARECOVER ADR012I (SCH)-DSSU (01), 97238 18:17:00 DFSMSdss PROCESSING COMPLETE. HIGHEST
RETURN CODE IS 0000
ARC6109I REPLACE OPTION SPECIFIED,
GDG DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN RESTORED USING EXISTING GDG ATTRIBUTES
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY RECOVERED USING AGGREGATE GROUP PAY1:
PAY1.PY001.NAMES
PAY2.PY002.RETIRE
PAY2.PY002.SOCSEC
PAY2.PY002.STATUS.G0002V00
PAY1.INSTRUCT
PAY1.PY001.CHECK
PAY1.PY001.PGMLIB
PAY1.PY002.LADDER
PAY1.PY002.IRA
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY ALLOCATED USING AGGREGATE GROUP PAY1:
PAY2.PY001.BENEFIT
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY CATALOGED USING AGGREGATE GROUP PAY1:
PAY2.PY001.TAXES
ARC6103I AGGREGATE RECOVERY HAS COMPLETED FOR AGGREGATE GROUP PAY1, USING CONTROL FILE
DATA SET PAY1.C.C01V0002,
AT 18:17:31, RETCODE = 000
```


Recovered data sets at the example aggregate recovery site

At the example site, the following data sets have been recovered to the DASD volumes specified:

- Recovered to an SMS-managed volume:
PAY2.PY002.SOCSEC
- Recovered to the level 0 volume PRIM02:
PAY2.PY002.STATUS.G0002V00
PAY1.PY001.CHECK
- Recovered to the migration volume MIG102:
PAY1.PY002.IRA
PAY1.PY001.BONUS
PAY1.PY002.LADDER

The following data sets have been recovered to tape:

- PAY1.PY001.NAMES

Data set PAY2.PY001.BENEFIT has been allocated to the level 0 volume PRIM02. Data set PAY2.PY001.TAXES has been cataloged.

ARECOVER CPU time and accounting codes

ABARS records the CPU time for processing ABACKUP and ARECOVER requests. This information is written to the FSR control block. The FSR can be written as an SMF record if the installation specifies SETSYS SMF(*smfid*). ABARS then writes the FSR as *smfid* + 1 in the SYS1.MANx or SYS1.MANy system data sets.

The ABACKUP CPU time and ARECOVER CPU time are also recorded in the ABR record. There is a difference, however, between the ARECOVER CPU time recorded in the FSR control block and the ABR record. If an ARECOVER request fails and is reissued with a valid RESTART data set, the CPU time in the FSR only reflects the time to process the remaining data sets. The ABR record accumulates the CPU times of each restart until the recovery of the aggregate is successful.

The ISMF aggregate group definition panel allows you to specify an accounting code for an aggregate group. The accounting code gets written to the FSR control block (and associated SMF record where one exists) along with the ABACKUP and ARECOVER CPU processing time for the aggregate. Installations can then use the information to assist in calculating charges to users for ABARS requests.

The accounting code is also written to the ABR record, and the ABACKUP control file. It is written to the ABACKUP control file so the code can be used at the recovery site without requiring an aggregate definition at the recovery site.

The accounting code is optional, and can be up to 32 characters in length. Valid characters are alphanumeric characters, blanks, commas, periods, parentheses, hyphens, slashes, ampersands, and apostrophes (single quotes).

Using ISMF panels to perform aggregate backup

You can use ISMF panels to process aggregate backup instead of issuing the ABACKUP command. You can specify the DFSMSdfp; Aggregate Group Application Selection panel from TSO and use either the LIST or the ABACKUP option for aggregate group backup.

Use the ABACKUP option when you know the aggregate group name you want to back up.

Use the LIST option when you are unsure of the aggregate group name you want to back up or when you want to back up more than one aggregate group.

In the example system, ACTIVE is used for the CDS NAME and the aggregate group name PAY1 is backed up to a 3490 tape unit. Two methods of performing aggregate backup, using either the ISMF LIST or the ISMF ABACKUP option, are shown.

Using the ISMF LIST option to perform aggregate backup

To use the LIST option for aggregate backup, use TSO ISPF panels to access the DFSMSdfp; Aggregate Group Application Selection panel, shown in [Figure 53 on page 414](#).

```

                                AGGREGATE GROUP APPLICATION SELECTION
COMMAND ===> _

TO PERFORM AGGREGATE GROUP OPERATIONS,SPECIFY:

CDS NAME                ===> 'ACTIVE'                (1 to 44 character dataset name)

AGGREGATE GROUP NAME ===> *                (for Aggregate Group List, fully or
                                           partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 1

1 LIST      - Generate a list of Aggregate Groups
2 DISPLAY   - Display an Aggregate Group
3 DEFINE    - Define an Aggregate Group
4 ALTER     - Alter an Aggregate Group
5 ABACKUP   - Backup an Aggregate Group
6 ARECOVER  - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
RESPECIFY SORT CRITERIA                ===> N      (Y or N)

USE ENTER TO PERFORM SELECTION:
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 53. Aggregate Group Application Selection Panel

On the Aggregate Group Application Selection Panel:

1. Enter 'ACTIVE' (quotes required) for **CDS NAME**. You must specify ACTIVE when backing up data sets, because aggregate backup can be initiated only from an active CDS.
2. Enter an asterisk (*) for **AGGREGATE GROUP NAME** to display a list of aggregate group names.
3. Enter 1 for **SELECT ONE OF THE FOLLOWING OPTIONS** to use the LIST option.
4. Press the Enter key to display the Aggregate Group List panel, as shown in [Figure 54 on page 414](#).

```

                                AGGREGATE GROUP LIST
COMMAND ===> _                                SCROLL ===> PAGE

                                           Entries 1-1 of 1
                                           Data Columns 16-19 of 19

CDS NAME:  ACTIVE

ENTER LINE OPERATORS BELOW:

  LINE   AGGGROUP  LAST DATE  LAST TIME  NUMBER  MGMTCLAS
  OPERATOR NAME    MODIFIED  MODIFIED  OF COPIES NAME
---(1)--- --(2)--- --- (16)--- --- (17)--- -- (18)--- -- (19)---
abackup  PAY1      1995/01/08  12:45      1  ABARMC1
          REMOTE    1995/02/04  08:25      1  ABARMC2
          TUCSON    1995/05/11  13:50      1  ABARMC2
-----
                        BOTTOM OF DATA
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 54. Aggregate Group List Panel

On the Aggregate Group List Panel, enter ABACKUP next to the aggregate group name PAY1 and press the Enter key to display the Aggregate Group Backup panel, as shown in [Figure 55 on page 415](#).


```

Panel Utilities Help
-----
DFQDCAG4          AGGREGATE GROUP BACKUP          Page 1 of 2
COMMAND ==>> -----
CDS NAME: ACTIVE

AGGREGATE GROUP NAME . . PAY1

UNIT NAME . . . . . -----
PROCESSING OPTION . . . . 2          (1=VERIFY, 2=EXECUTE)
WAIT FOR COMPLETION . . . N          (Y or N)
STACK / NOSTACK . . . . . -          (S=STACK, N=NOSTACK or blank)
OPTIMIZE . . . . . -          (1 to 4 or blank)
DELETE DATA SETS AFTER ABACKUP . . - (Y or N)

FILTER OUTPUT DATA SET NAME (1 to 44 characters)
==>> -----

Process only          _ L0          _ ML1          _ ML2          _ USERTAPE

USE ENTER CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 55. Aggregate Group Backup Panel

On the Aggregate Group Backup Panel, you do not need to enter the CDS NAME or the AGGREGATE GROUP NAME; they are carried over from the previous panel.

Note: You can change the aggregate group name and initiate backup for another aggregate group from this panel.

On the Aggregate Group Backup Panel:

1. Enter 3490 for **UNIT NAME**.
2. Enter a 1 for **PROCESSING OPTION**. to perform aggregate backup verification processing, If you specify VERIFY, no data movement will take place.

Enter a 2 for **PROCESSING OPTION** to EXECUTE aggregate group backup. Data movement will take place.
3. Enter a Y for **WAIT FOR COMPLETION** to specify that you want to wait for aggregate backup to complete. If you specify Y, you will not be able to perform any other functions from your terminal while aggregate backup is processing.

Enter an N for **WAIT FOR COMPLETION** to specify that you do not want to wait.
4. Enter an S for **STACK / NOSTACK** if you want to stack your ABARS output files on a minimum number of tape volumes.

Enter an N for **STACK / NOSTACK** if you do not want to stack your ABARS output files.

Enter a blank for **STACK / NOSTACK** to cause aggregate backup processing to default to the value specified by the SETSYS ABARSTAPES(STACK | NOSTACK) command. If SETSYS ABARSTAPES isn't specified, the default is STACK.
5. Enter one of the following values for OPTIMIZE, which ABACKUP processing passes to DFSMSDss when DFSMSDss is backing up the level 0 DASD data sets specified in the include list.

Value

Description

1

Causes DFSMSDss to read one track at a time.

2

Causes DFSMSdss to read two tracks at a time.

3

Causes DFSMSdss to read five tracks at a time.

4

Causes DFSMSdss to read one cylinder at a time.

blank

Causes this option to default to the setting specified in the SETSYS ABARSOPTIMIZE(n) command. If SETSYS ABARSOPTIMIZE is not specified, the default is 3 (DFSMSdss reads five tracks at a time).

6. Enter Y for **DELETE DATA SETS AFTER ABACKUP** if you want the data sets deleted following successful completion of ABACKUP processing.

Enter N for **DELETE DATA SETS AFTER ABACKUP** if you do not want the data sets deleted.

7. Enter a slash (/) to choose the **PROCESS ONLY** selection or selections you want to include in your processing. If you do not choose any PROCESS ONLY selections, then the PROCESS ONLY option is not in effect. When the PROCESS ONLY option is not in effect, it is the same as choosing all of the selections; LO, ML1, ML2, and USERTAPE data sets are all eligible for processing.

8. Press the Enter key to initiate processing.

Using the ISMF ABACKUP option to perform aggregate backup

To use the ABACKUP option for aggregate backup, use TSO ISPF panels to access the DFSMSdfp; Aggregate Group Application Selection panel, shown in [Figure 56 on page 416](#).

```

                                AGGREGATE GROUP APPLICATION SELECTION
COMMAND ===> _

TO PERFORM AGGREGATE GROUP OPERATIONS,SPECIFY:

CDS NAME                ===> 'ACTIVE'
                                (1 to 44 character dataset name)

AGGREGATE GROUP NAME ===> PAY1
                                (for Aggregate Group List, fully or
                                partially specified or * for all)

SELECT ONE OF THE FOLLOWING OPTIONS ===> 5

1 LIST      - Generate a list of Aggregate Groups
2 DISPLAY   - Display an Aggregate Group
3 DEFINE    - Define an Aggregate Group
4 ALTER     - Alter an Aggregate Group
5 ABACKUP   - Backup an Aggregate Group
6 ARECOVER  - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
RESPECIFY SORT CRITERIA                ===> N      (Y or N)

USE ENTER TO PERFORM SELECTION:
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 56. Aggregate Group Application Selection Panel

On the Aggregate Group Application Selection Panel:

1. Enter 'ACTIVE' (quotes required) for **CDS NAME**. You must specify ACTIVE when backing up data sets, because aggregate backup can be initiated only from an active CDS.
2. Enter PAY1 for **AGGREGATE GROUP NAME**.
3. Enter 5 for **SELECT ONE OF THE FOLLOWING OPTIONS** to use the ABACKUP option.
4. Press the Enter key to display the Aggregate Group Backup panel, as shown in [Figure 57 on page 417](#).


```

Panel Utilities Help
-----
DFQDCAG4          AGGREGATE GROUP BACKUP          Page 1 of 2
COMMAND ==>> -----
CDS NAME: ACTIVE

AGGREGATE GROUP NAME . . PAY1

UNIT NAME . . . . . -----
PROCESSING OPTION . . . . 2          (1=VERIFY, 2=EXECUTE)
WAIT FOR COMPLETION . . . N          (Y or N)
STACK / NOSTACK . . . . . _          (S=STACK, N=NOSTACK or blank)
OPTIMIZE . . . . . _          (1 to 4 or blank)
DELETE DATA SETS AFTER ABACKUP . . _ (Y or N)

FILTER OUTPUT DATA SET NAME (1 to 44 characters)
==>> -----

Process only          _ L0          _ ML1          _ ML2          _ USERTAPE

USE ENTER CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.

```

Figure 57. Aggregate Group Backup Panel

You do not need to enter the CDS NAME or the AGGREGATE GROUP NAME; they are carried over from the previous panel.

Note: You can change the aggregate group name and initiate backup for another aggregate group from this panel.

On the Aggregate Group Backup Panel:

1. Enter 3490 for **UNIT NAME**.
2. Enter a 1 for **PROCESSING OPTION**. to perform aggregate backup verification processing, If you specify VERIFY, no data movement will take place.

Enter a 2 for **PROCESSING OPTION** to EXECUTE aggregate group backup. Data movement will take place.
3. Enter a Y for **WAIT FOR COMPLETION** to specify that you want to wait for aggregate backup to complete. If you specify Y, you will not be able to perform any other functions from your terminal while aggregate backup is processing.

Enter an N for **WAIT FOR COMPLETION** to specify that you do not want to wait.
4. Enter an S for **STACK / NOSTACK** if you want to stack your ABARS output files on a minimum number of tape volumes.

Enter an N for **STACK / NOSTACK** if you do not want to stack your ABARS output files.

Enter a blank for **STACK / NOSTACK** to cause aggregate backup processing to default to the value specified by the SETSYS ABARSTAPES(STACK | NOSTACK) command. If SETSYS ABARSTAPES isn't specified, the default is STACK.
5. Enter one of the following values for OPTIMIZE, which ABACKUP processing passes to DFSMSdss when DFSMSdss is backing up the level 0 DASD data sets specified in the include list:

Value	Description
-------	-------------

- | | |
|---|--|
| 1 | Causes DFSMSdss to read one track at a time. |
|---|--|

2

Causes DFSMSdss to read two tracks at a time.

3

Causes DFSMSdss to read five tracks at a time.

4

Causes DFSMSdss to read one cylinder at a time.

blank

Causes this option to default to the setting specified in the SETSYS ABARSOPTIMIZE(n) command. If SETSYS ABARSOPTIMIZE is not specified, the default is 3 (DFSMSdss reads five tracks at a time).

6. Enter Y for **DELETE DATA SETS AFTER ABACKUP** if you want the data sets deleted following successful completion of ABACKUP processing.

Enter N for **DELETE DATA SETS AFTER ABACKUP** if you do not want the data sets deleted.

7. Enter a slash (/) to choose the **PROCESS ONLY** selection or selections you want to include in your processing. If you do not choose any PROCESS ONLY selections, then the PROCESS ONLY option is not in effect. When the PROCESS ONLY option is not in effect, it is the same as choosing all of the selections; LO, ML1, ML2, and USERTAPE data sets are all eligible for processing.

8. Press the Enter key to initiate processing.

Using ISMF panels to perform aggregate recovery

This topic demonstrates how, instead of issuing the ARECOVER command, you can use ISMF panels to process aggregate recovery. You can specify the DFSMSdfp; Aggregate Group Application Selection panel from TSO and use option 6 for aggregate group recovery.

For this example ISMF session, PAY1.C.C01V0001 is specified as the control file data set name, a 3490 unit type is specified, and volume 111111 is allocated for the recovery files.

To use the ARECOVER option for aggregate recovery, use TSO ISPF panels to access the DFSMSdfp; aggregate group application selection panel, shown in [Figure 58 on page 418](#).

```

                                AGGREGATE GROUP APPLICATION SELECTION
COMMAND ==> _

TO PERFORM AGGREGATE GROUP OPERATIONS, SPECIFY:

CDS NAME                ==> 'ACTIVE'                (1 to 44 character dataset name)

AGGREGATE GROUP NAME ==> PAY1                (for Aggregate Group List, fully or
                                           partially specified or &asterisk for all)

SELECT ONE OF THE FOLLOWING OPTIONS ==> 6

1 LIST      - Generate a list of Aggregate Groups
2 DISPLAY   - Display an Aggregate Group
3 DEFINE    - Define an Aggregate Group
4 ALTER     - Alter an Aggregate Group
5 ABACKUP   - Backup an Aggregate Group
6 ARECOVER  - Recover an Aggregate Group

IF OPTION 1 CHOSEN ABOVE,
RESPECIFY SORT CRITERIA                ==> N                (Y or N)

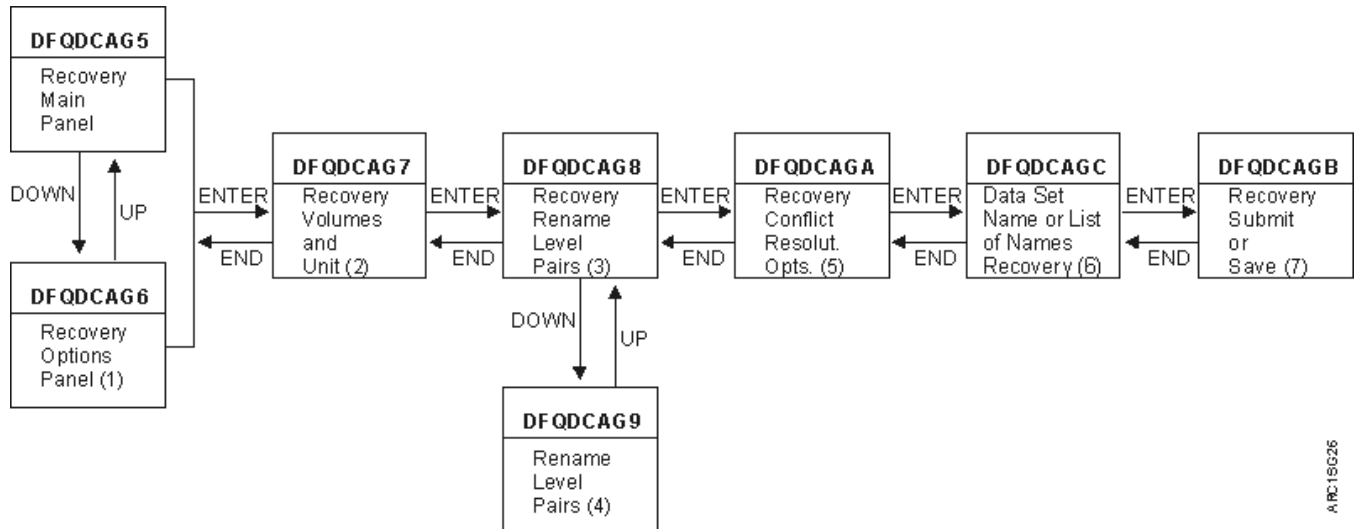
USE ENTER TO PERFORM SELECTION:
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 58. Aggregate Group Application Selection Panel

On the Aggregate Group Application Selection Panel:

1. Enter 6 in the SELECT ONE OF THE FOLLOWING OPTIONS field to select the ARECOVER option.
2. Press the ENTER key to display the first aggregate group recovery panel.

Figure 59 on page 419, with associated notes, maps the progression through the aggregate group recovery panels, and tells how each panel is accessed.



ARC15G26

Figure 59. Aggregate Group Recovery Panels

Note:

1. Panel DFQDCAG6 is displayed only if DOWN is entered on panel DFQDCAG5.
2. Panel DFQDCAG7 is displayed only if a value is entered in the ABACKUP CONTROL DATASET field of panel DFQDCAG5, N is specified in the XMIT field of panel DFQDCAG5, and ENTER is entered on either panel DFQDCAG5 or panel DFQDCAG6.
3. Panel DFQDCAG8 is displayed only if Y is entered in the RENAME SELECTED DATA SETS field of panel DFQDCAG6, and ENTER is entered on panel DFQDCAG7.
4. Panel DFQDCAG9 is displayed only if DOWN is entered on panel DFQDCAG8.
5. Panel DFQDCAGA is displayed only if Y is entered in the SPECIFY CONFLICT RESOLUTION OPTION field of panel DFQDCAG6, and ENTER is entered on panel DFQDCAG8.
6. Panel DFQDCAGC is displayed only if Y is entered in the RECOVER INDIVIDUAL DATASET field of panel DFQDCAG5.
7. Panel DFQDCAGB is displayed only if all required fields are filled in on the other panels, and ENTER is entered on panel DFQDCAGA.

Panel DFQDCAG5

Figure 60 on page 420 shows a representation of the ISMF DFQDCAG5 aggregate recovery main panel.


```

Panel  Utilities  Scroll  Help
-----
DFQDCAG5                      AGGREGATE GROUP RECOVER                      Page 1 of 8
Command ==> -----
Abackup Control Dataset . . 'PAY1.C.C01V0001'-----
                                   (1 to 44 Characters)
Xmit . . . . . N                      (Y or N)
Stack / Nostack . . . . . _          (S=STACK, N=NOSTACK or blank)
Aggregate Group Name . . . . . -----
Date . . . . .                      (yyyy/mm/dd)
Version . . . . .                      (1 to 9999)

Processing Option . . . . . 3          (1=Prepare, 2=Verify, 3=Execute)
Wait for Completion . . . . . Y        (Y or N)
Target GDG Data Set Status . . _      (A=ACTIVE, D=DEFERRED, R=ROLLEDOFF,
                                   S=SOURCE or blank)
Volume Count . . . . . _              (A=ANY, N=NONE or blank)
Recover Instruction Data Set . . N      (Y or N)
Recover Activity Log . . . . . Y        (Y or N)
Recover Individual Data Set . . N        (Y or N)

Use ENTER to Continue; Use DOWN to View Additional Options;
Use HELP Command for Help; Use END Command to Exit.

```

Figure 60. Aggregate Recovery Panel DFQDCAG5

The following is a description of the fields of panel DFQDCAG5, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field generates a DATASETNAME(*dsname*) parameter.
Enter 'PAY1.C.C01V0001' in this field
- The XMIT field indicates whether the ABACKUP files were independently transmitted to the ARECOVER site.
Enter N in this field.
- The STACK / NOSTACK field indicates whether the ABACKUP output files were stacked on a minimum number of tapes, or not.
Enter N in this field.
- The AGGREGATE GROUP NAME field generates an AGGREGATE(*agname*) parameter.
Leave this field blank.
- The DATE field generates a DATE(*yyyy/mm/dd*) parameter.
Leave this field blank.
- The VERSION field generates a VERSION(*nnnn*) parameter.
Leave this field blank.
- The PROCESSING OPTION field allows you to place either the PREPARE, VERIFY, or EXECUTE parameters in the ARECOVER command. A 1 generates the PREPARE parameter, a 2 generates the VERIFY parameter, and a 3 generates the EXECUTE parameter.
Enter a 3 in this field.
- The WAIT FOR COMPLETION field, if Y is entered, generates an HSEND WAIT ARECOVER command, and the terminal is locked until the ARECOVER command completes. If N is entered, this field generates an HSEND ARECOVER command, and the terminal remains unlocked.
Enter Y in this field.
- The TARGET GDG DATA SET STATUS field allows you to specify the status to which you want an SMS-managed generation data set returned. You can specify A for ACTIVE, D for DEFERRED, R for ROLLEDOFF, S for SOURCE, or leave the field blank. SOURCE indicates that you want the data set returned to the status it had when it was backed up. A blank will cause value specified by the SETSYS ARECOVERGTGDS to be used, or its default value of SOURCE.
Leave this field blank.

- The VOLUME COUNT field allows you to specify A for ANY, N for NONE, or a blank. If ANY is specified, the VOLCOUNT(ANY) parameter is passed to DFSMSdss during the restore, which results in the allocation of the target data set on as many volumes as required, up to a maximum of 59 volumes. If NONE is specified, the DFSMSdss VOLCOUNT parameter is not passed to DFSMSdss during the restore. If the field is left blank, the value specified by the SETSYS ABARSVOLCOUNT parameter, or its default value of NONE, is used.

Leave this field blank.

- The RECOVER INSTRUCTION DATA SET field, when Y is entered, generates the INSTRUCTION parameter. If AGGREGATE GROUP NAME is specified with the PREPARE option, either the RECOVER INSTRUCTION DATA SET field, the RECOVER ACTIVITY LOG field, or both must be used.

Enter N in this field.

- The RECOVER ACTIVITY LOG field, when Y is entered, generates the ACTIVITY parameter.

Enter Y in this field.

- The RECOVER INDIVIDUAL DATASET field specifies whether you want to recover an individual data set or a group of data sets from the backup copy of an aggregate group.

Enter N in this field.

Panel DFQDCAG6

Figure 61 on page 421 shows a representation of the ISMF DFQDCAG6 aggregate recovery options panel.

```

Panel  Utilities  Scroll  Help
-----
DFQDCAG6                      AGGREGATE GROUP RECOVER                      Page 2 of 8
Command ==>> -----
AGGREGATE GROUP NAME: 'PAY1.C.COV0001'

Model Entity . . . . ----- (1 to 44 Characters)
Target Unit Name . . -----
Target Utilization Percentage . . ___ (1 to 100)
Backup Migrated Data Sets . . . . Y (Y or N)
New Migration Level . . . . . 3 (1=ML1, 2=ML2, 3=SOURCELEVEL)

Rename All to High Level Qualifier . . ----- (1 to 8 characters)
Rename Selected Data Sets . . . . . Y (Y or N)

Specify Conflict Resolution Option . . Y (Y or N)

Use ENTER to Continue; Use UP to View Previous Options;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 61. Aggregate Recovery Panel DFQDCAG6

The following is a description of the fields of panel DFQDCAG6, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field indicates the name of the ABACKUP CONTROL DATASET currently being processed.
- The MODEL ENTITY field allows the entry of a quoted, fully qualified data set name without a TSO *userid* prefix. This field generates, when filled in, a MENTITY(*dsname*) parameter.

Leave this field blank.

- The TARGET UNIT NAME field generates, when filled in, a TARGETUNIT(*unitname*) parameter.

Leave this field blank.

- The TARGET UTILIZATION PERCENTAGE field, if a percentage number is entered, generates a PERCENTUTILIZED(*nnn*) parameter.
Leave this field blank.
- The BACKUP MIGRATED DATA SETS field generates, if Y is entered, a BACKUPMIGRATED parameter. If N is entered, a NOBACKUPMIGRATED parameter is generated.
Enter Y in this field.
- The NEW MIGRATION LEVEL field indicates whether the MIGRATEDDATA parameter and associated subparameters are specified on the ARECOVER command. A 1 in this field generates a MIGRATEDDATA(ML1) parameter (the default), a 2 in this field generates a MIGRATEDDATA(ML2) parameter, and a 3 in this field generates a MIGRATEDDATA(SOURCELEVEL) parameter.
Enter 3 in this field.
- The RENAME ALL TO HIGH LEVEL QUALIFIER field accepts a valid, 1 to 8 character data set name qualifier that generates a RECOVERNEWNAMEALL(*level*) parameter, with the specified *level* as the subparameter. A value in this field is mutually exclusive with a Y in RENAME SELECTED DATA SETS.
Leave this field blank.
- The RENAME SELECTED DATA SETS field, if Y is entered, displays panel DFQDCAG8, shown in [Figure 63 on page 423](#). A Y in this field is mutually exclusive with a value in the RENAME ALL TO HIGH LEVEL QUALIFIER field.
Enter Y in this field.
- The SPECIFY CONFLICT RESOLUTION OPTION field, if Y is entered, displays panel DFQDCAGA, shown in [Figure 65 on page 424](#).
Enter Y in this field.

Panel DFQDCAG7

[Figure 62 on page 422](#) shows a representation of the ISMF DFQDCAG7 aggregate recovery volumes and unit panel.

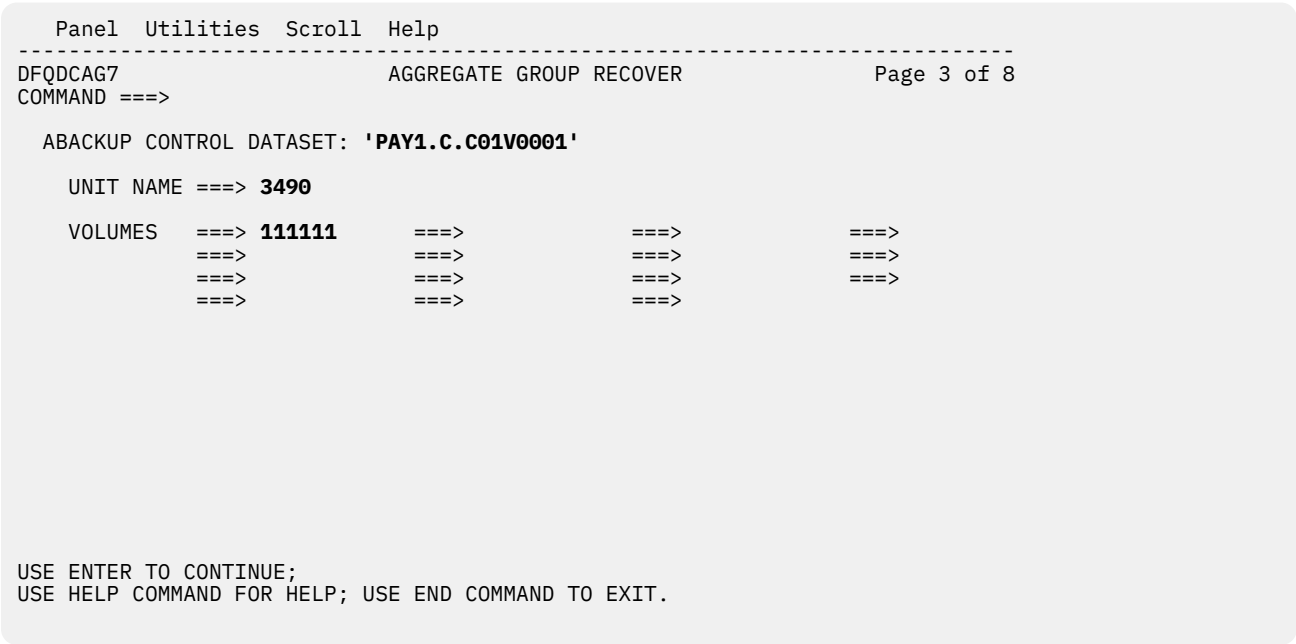


Figure 62. Aggregate Recovery Panel DFQDCAG7

The following is a description of the fields of panel DFQDCAG7, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field indicates the name of the ABACKUP CONTROL DATASET currently being processed.
- The UNIT NAME field generates the UNIT(*unitname*) parameter, where *unitname* is the unit name that you enter in the field.

Enter 3490 in this field.

- The VOLUMES fields can accept up to 15 entries. These fields, when filled in, generate a VOLUMES(*volser1...*,*volsern*) parameter.

Enter 111111 in the first of these fields.

Panel DFQDCAG8

Figure 63 on page 423 shows a representation of the ISMF DFQDCAG8 aggregate recovery rename level pairs panel.

```

Panel  Utilities  Scroll  Help
-----
DFQDCAG8                      AGGREGATE GROUP RECOVER                      Page 4 of 8
COMMAND ===>

ABACKUP CONTROL DATASET: 'PAY1.C.C01V0001'
  TO RENAME SELECTED DATA SETS, SPECIFY LEVEL PAIRS (1 to 8 characters):

    OLD LEVEL ===> LEVEL0      NEW LEVEL ===> LEVEL0N
    OLD LEVEL ===> LEVEL1      NEW LEVEL ===> LEVEL1N
    OLD LEVEL ===> LEVEL2      NEW LEVEL ===> LEVEL2N
    OLD LEVEL ===> LEVEL3      NEW LEVEL ===> LEVEL3N
    OLD LEVEL ===> LEVEL4      NEW LEVEL ===> LEVEL4N
    OLD LEVEL ===> LEVEL5      NEW LEVEL ===> LEVEL5N
    OLD LEVEL ===> LEVEL6      NEW LEVEL ===> LEVEL6N
    OLD LEVEL ===> LEVEL7      NEW LEVEL ===> LEVEL7N
    OLD LEVEL ===> LEVEL8      NEW LEVEL ===> LEVEL8N
    OLD LEVEL ===> LEVEL9      NEW LEVEL ===> LEVEL9N
    OLD LEVEL ===> LEVELA      NEW LEVEL ===> LEVELAN
    OLD LEVEL ===> LEVELB      NEW LEVEL ===> LEVELBN
    OLD LEVEL ===> LEVELC      NEW LEVEL ===> LEVELCN
    OLD LEVEL ===> LEVELD      NEW LEVEL ===> LEVELDN
    OLD LEVEL ===> LEVELE      NEW LEVEL ===> LEVELEN

USE ENTER TO CONTINUE; USE DOWN TO SPECIFY ADDITIONAL LEVEL PAIRS;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 63. Aggregate Recovery Panel DFQDCAG8

The following is a description of the fields of panel DFQDCAG8, and the entries that are made for the example system:

- The ABACKUP CONTROL DATASET field indicates the name of the ABACKUP CONTROL DATASET currently being processed.
- The OLD LEVEL and NEW LEVEL fields can each contain a valid 1 to 8 character data set name. Values other than blanks in these fields generate the RECOVERNEWNAMELEVEL(*olevel1*,*nlevel1* [,*oleveln*,*nleveln...*]) parameter. At least one LEVEL pair must be specified. Each OLD LEVEL name specified must have a corresponding NEW LEVEL name. The OLD LEVEL name must differ from the corresponding NEW LEVEL name. Up to 15 pairs of names can be entered on this panel.

Enter the LEVEL names as shown in Figure 63 on page 423.

A second panel, DFQDCAG9, with fields for 15 more name pairs, can be displayed by using the DOWN command.

Panel DFQDCAG9

Figure 64 on page 424 shows a representation of the ISMF DFQDCAG9 aggregate recovery rename level pairs panel, which is a continuation of the ISMF DFQDCAG8 panel.


```
Panel Utilities Scroll Help
-----
DFQDCAG9                                AGGREGATE GROUP RECOVER                Page 5 of 8
COMMAND ==>

ABACKUP CONTROL DATASET: 'PAY1.C.C01V0001'
TO RENAME SELECTED DATA SETS, SPECIFY LEVEL PAIRS (1 to 8 characters):

OLD LEVEL ==> LEVELF                     NEW LEVEL ==> LEVELFN
OLD LEVEL ==> LEVELG                     NEW LEVEL ==> LEVELGN
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>
OLD LEVEL ==>                          NEW LEVEL ==>

USE ENTER TO CONTINUE; USE UP TO VIEW PREVIOUS LEVEL PAIRS;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 64. Aggregate Recovery Panel DFQDCAG9

This panel is a continuation of panel DFQDCAG8, and the explanation of the fields is the same.

Enter the LEVEL names as shown in Figure 64 on page 424.

Panel DFQDCAGA

Figure 65 on page 424 shows a representation of the ISMF DFQDCAGA aggregate recovery conflict resolution options panel.

```

Panel  Utilities  Scroll  Help
-----
DFQDCAGA                                AGGREGATE GROUP RECOVER                Page 6 of 8
COMMAND ==>

ABACKUP CONTROL DATASET: 'PAY1.C.C01V0001'

SPECIFY CONFLICT RESOLUTION OPTION:

RENAME SOURCE NEW LEVEL ==>                (1 to 8 characters)
RENAME TARGET NEW LEVEL ==>                (1 to 8 characters)
REPLACE                                ==>  N    (Y or N)
BYPASS                                ==>  Y    (Y or N)

USE ENTER TO CONTINUE;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 65. Aggregate Recovery Panel DFQDCAGA

The following is a description of the fields of panel DFQDCAGA, and the entries that are made for the example system:

- The **ABACKUP CONTROL DATASET** field indicates the name of the **ABACKUP CONTROL DATASET** currently being processed.

- The RENAME SOURCE NEW LEVEL field can contain a valid 1-to-8 character data set name that generates the RENAMESOURCE(*level*) subparameter of the DATASETCONFLICT parameter, indicating the specified level.

Leave this field blank.

- The RENAME TARGET NEW LEVEL can contain a valid 1-to-8 character data set name that generates the RENAMETARGET(*level*) subparameter of the DATASETCONFLICT parameter, indicating the specified level.

Leave this field blank.

- The REPLACE field, when Y is entered, generates the REPLACE subparameter of the DATASETCONFLICT parameter.

Enter N in this field.

- The BYPASS field, when Y is entered, generates the BYPASS subparameter of the DATASETCONFLICT parameter.

Enter Y in this field.

Note: The RENAME SOURCE NEW LEVEL, RENAME TARGET NEW LEVEL, REPLACE, and BYPASS fields are mutually exclusive: only one field can have an entry.

Panel DFQDCAGC

If you entered Y in the RECOVER INDIVIDUAL DATASET field of panel DFQDCAG5 (Figure 60 on page 420), your ISMF session includes panel DFQDCAGC. Figure 66 on page 425 shows a representation of the ISMF DFQDCAGC individual data set recovery panel.

```

Panel  Utilities  Scroll  Help
-----
DFQDCAGC                      AGGREGATE GROUP RECOVER          Page 7 of 8
COMMAND ===>

AGGREGATE GROUP NAME:
AG1

Specify Data Set Name for Individual Recovery:

Single Data Set Name . . . ----- (1 to 44 Characters)

List of Names Data Set . . . ----- (1 to 44 Characters)

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

Figure 66. Aggregate Recovery Panel DFQDCAGC

The following is a description of the fields of panel DFQDCAGC, which contains two fields. Enter either a single data set name or the name of a data set that contains a list of data sets to be recovered:

- The SINGLE DATA SET NAME field specifies the recovery of a single data set.
- The LIST OF NAMES DATA SET specifies the data set name that contains a list of data sets.

Only one of the two fields can be specified; leave the other field blank.

Panel DFQDCAGB

Figure 67 on page 426 shows a representation of the ISMF DFQDCAGB aggregate recovery submit or save panel.

```

Panel  Utilities  Scroll  Help
-----
DFQDCAGB                      AGGREGATE GROUP RECOVER          Page 8 of 8
COMMAND ===>

DFSMSHSM COMMAND AND PROCESSING OPTION:
ARECOVER DATASETNAME(PAY1.C.C01V0001) EXECUTE

ENTER 1 TO SUBMIT DFSMSHSM ARECOVER COMMAND
ENTER 2 TO SAVE GENERATED ARECOVER PARAMETERS


SELECT OPTION    ===> 1                                (1-SUBMIT, 2-SAVE)


USE ENTER TO PERFORM SELECTION;
USE HELP COMMAND FOR HELP; USE END COMMAND TO EXIT.
```

Figure 67. Aggregate Recovery Panel DFQDCAGB

The following is a description of the fields of panel DFQDCAGB, and the entries that are made for the example system:

- The DFSMSHSM COMMAND AND PROCESSING OPTION field indicates the DFSMSHsm command and processing option currently being processed. In the case of the example, it appears as ARECOVER DATASETNAME(PAY1.C.C01V0001) EXECUTE.
- The SELECT OPTION field allows one of two choices: entering 1 saves the generated ARECOVER parameters and submits the ARECOVER command to DFSMSHsm, and entering 2 saves the generated ARECOVER parameters but does not submit the ARECOVER command.

Enter 1 in this field.

Using ISMF to recover individual data sets from an aggregate backup

To recover an individual data set from an aggregate backup version, you can use ISMF panels, as described in this chapter.

Perform the following steps to recover an individual data set from the backup copy of an aggregate group:

1. To access the ARECOVER option for aggregate recovery, use TSO ISPF panels to access the DFSMSdftp; aggregate group application selection panel, shown in [Figure 58 on page 418](#).
2. Select option 9, Aggregate Group, and press Enter. The Aggregate Group Application Selection panel is displayed.
3. Select option 6, ARECOVER, and press Enter to view the first Aggregate Group Recover panel DFQDCAG5."
4. Specify the values for the aggregate group. Specify Y for the field "Recover Individual Data Set". Press Enter to view the Aggregate Recovery Group Panel DFQDCAGC ([Figure 66 on page 425](#)).
5. On the Aggregate Recovery Group Panel DFQDCAGC, specify either a single data set name or the name of a data set that contains a list of data sets to be recovered from the backup copy of an aggregate group.

To create a list of data sets, use a fixed-block, sequential data set with a record size of 80 and logical record length (LRECL) of 80, and then catalog it. Edit the data set and specify one fully qualified data set name per record.

6. Enter the END command to submit the recover request.

Using the NetView file transfer program to transmit aggregate Backup files

You can use the NetView FTP to create file transfer requests for aggregate backup files and submit them to NetView FTP for processing. Requests can be created either by entering values in the input fields of the NetView FTP panels or by writing and running a batch job. After submitting one request, you can immediately submit another; you do not have to wait until NetView FTP has processed the first request.

Requests are submitted to a component of NetView FTP called the *queue handler*. The queue handler assigns a request number to your request and adds it to the request queue, where it waits to be processed. The request queue is located on DASD.

After your request has been added to the request queue, you can DELETE it from the queue or QUERY the queue for information about a request.

Before you can use NetView FTP to transmit your backup files, you will need to process the aggregate backup command. The ABACKUP command can create up to four data sets that should be transmitted to a remote location. The naming convention for these four data sets is as follows:

- outputdatasetprefix.D.CnnVnnnn

This data set contains data from the data file that originally resided on level 0 DASD before it was backed.

- outputdatasetprefix.O.CnnVnnnn

This data set contains data from the data file that was migrated or resided on user tape volumes before it was backed up.

- outputdatasetprefix.C.CnnVnnnn

This data set contains the control information from the control file.

- outputdatasetprefix.I.CnnVnnnn

This data set may contain valid instructions to the recovery site, or the ABACKUP activity log, if it was directed to DASD.

Note: Any ACCOMPANY data sets must also be transmitted.

Related reading

For more information about NetView FTP, see *Tivoli® NetView File Transfer Program* publications.

Using NetView file transfer program panels

You can use NetView FTP panels to create, save, change, add, query, and delete your transfer requests. Which panels NetView FTP displays and the order in which it displays them, depend on the task you select and on the data you enter in the input fields of the panels. The following topics contain examples of NetView FTP panels and examples of how to use the panels to transfer copies of aggregate backup files to a receiving location.

Invoking NetView FTP panels

Invoke NetView FTP from your TSO selection menu. The Main Task Selection panel, which is shown in [Figure 68 on page 428](#), is displayed.


```
DVGSPEPE----- NetView FTP Main Task Selection -----
OPTION ==> 1

Select an option by number

1 Create          - Create a new file-transfer request
2 Control         - Query, delete, modify, hold, release submitted requests
3 List Saved      - List saved requests
4 Change Saved    - Change a saved request
5 Submit Saved    - Submit a saved request
6 Administer      - Administer the request queue

Specify the name of the request for option 4 or 5
Request Name ==>

Press the END PF key to return, HELP PF key for help
```

Figure 68. NetView FTP Main Task Selection Panel

Option 1 from the NetView FTP Main Task Selection panel is selected to show you how to create a file transfer request for the data file data set that contains the data backed up by using DFSMSdss.

Creating a request

Each request is made up of *parameters* that tell NetView FTP everything it needs to know to transfer this file.

In the example system, the aggregate backup file PAY1.D.C01V0001 is transferred to the *Remote Node ID* TKSMVS1. The file is stored on the labeled tape volume serial number ABRT01. You want the data set to be received at the remote location under the data set name PAY1.D.C01V0001 and stored on a 3490 tape volume, serial number ABRT01. You have decided that a data set transferred from tape will be given a *request class* of 3. To protect the request, you enter the *request password* ROSE. The password does not appear on the screen when typed. You do not want NetView FTP to process the request before 6:00 p.m. today. You want NetView FTP to notify the sending user ID, X999999 at node XYZMVS4, and the receiving user ID, Z000000 at node TKSMVS1, that the data set has been transferred.

This is how you use the NetView Main Transfer Parameters panel, which is shown in [Figure 69 on page 428](#), to tell NetView FTP how to conduct this file transfer.

```
DVGIT20E----- Main Transfer Parameters -----
COMMAND ==>

Specify whether you want to transfer a file to or from a remote node
Transfer Mode      ==> TO      (TO, FROM)

Specify either a remote node ID or a remote LU name
Remote Node ID     ==> TKSMVS1
Remote LU name     ==>

Specify a request class, priority, and password
Request Class      ==> 3      (0 to 9; A to Z)
Request Priority    ==> 0      (0 to 9)
Request Password    ==>      optional

Specify a running mode for each server
Sending Server     ==> CONT    (CONT, SINGLE)
Receiving Server   ==> CONT    (CONT, SINGLE)

Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 69. Main Transfer Parameters Panel

After the Main Transfer Parameters panel is filled in, press the Enter key. The NetView FTP Main Data Set Parameters panel shown in [Figure 70 on page 429](#) is displayed.

```
DVGIT1SE----- Main Data Set Parameters -----
COMMAND ==>

Sending Data Set (Local):

Data Set Name ==> 'PAY1.D.C01V0001'
DD Name       ==>
Member Name   ==> (blank=list, *=all)
Data Set Type ==> LAB (blank,VSAM,P0,PS,LAB,UNLAB,USER)
Volume       ==>

Receiving Data Set (Remote):

Data Set Name ==> 'PAY1.D.C01V0001'
DD Name       ==>
Member Name   ==>
Data Set Type ==> LAB (blank,VSAM,P0,PS,LAB,UNLAB,USER)
Catalog Name  ==>

Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 70. Main Data Set Parameters Panel

After the Main Data Set Parameters panel is filled in, press Enter. NetView FTP displays the Additional Transfer Parameters panel, where you enter the not-before time and the names of the report recipients as shown in [Figure 71 on page 429](#).

```
DVGID30E----- Additional Transfer Parameters -----
COMMAND ==>

Not-before   Date ==>  /  / (YY / MM / DD)
              Time ==> 18 : 00 (HH : MM)
Not-after    Date ==>  /  / (YY / MM / DD)
              Time ==>  : (HH : MM)

Restart from Checkpoint ==> NO (YES,NO)
Automatic Transfer Restart ==> YES (YES,NO)
Compression Method ==> ADAPT (SNA,ADAPT,NONE)
Hold Request ==> NO (YES,NO)

Recipient of Report from Sending Program ==> X999999 at XYZMVS4
Recipient of Report from Receiving Program ==> Z000000 at TKSMVS1

Local LU Name ==> LU name of local server to be used

Installation Exit Routine Input
==> tape check

Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 71. Additional Transfer Parameters Panel

After you complete your entries on the Additional Transfer Parameters panel and press Enter, NetView FTP displays the Additional Sending Data Set Parameters panel, where you specify the device type, volume serial number, user ID (user ID is required), and security parameters as shown in [Figure 72 on page 430](#).

You do not need to fill in these fields when transferring any of the data sets, because they are cataloged during aggregate backup processing.


```
DVGID2RE Additional Parameters for the Sending Data Set (PO,PS,LAB,UNLAB) -----
COMMAND ==>

DCB Parameters:
  Record Format      ==>                               (F,FB,V,VB,U,VS,VBS)
  Logical Record Length ==>
  Physical Block Size ==>

Security parameters:
  User ID ==> X999999
  Password ==>
  Group ID ==>

Specify values for the following if data set not cataloged:
  Device Type      ==> 3490
  Volume Serial Numbers ==> ABRT01 ==> ==>
  ==> ==> ==>

Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 72. Additional Parameters for the Sending Data Set Panel

Press Enter after you fill in the additional parameters for the Sending Data Set panel. NetView FTP displays the additional parameters for the Receiving Data Set panel, where you specify the disposition, device type, volume serial number, user ID (user ID is required), and security parameters for the receiving data set, as shown in [Figure 73 on page 430](#).

Note: If CAT is not entered in the disposition field, the data set must be manually cataloged at the receiving location.

```
DVGID2RE Additional Parameters for the Receiving Data Set (PS,LAB,UNLAB) -----
COMMAND ==>

Disposition      ==> CAT      (NEW,OLD,CAT)

Security Parameters:
  User ID      ==> Z000000
  Password     ==>
  Group ID     ==>

Specify values for the following if data set is not cataloged
  Device Type      ==> 3490
  Volume Serial Numbers ==> ABRT01 ==> ==>
  ==> ==> ==>

Press the ENTER key to continue, END PF key to return, HELP PF key for help
```

Figure 73. Example 2: Additional Parameters for the Receiving Data Set Panel

After you fill in the additional parameters for the Receiving Data Set panel and press Enter, NetView FTP has all the input it needs to conduct the file transfer.

Using a batch job to transmit aggregate backup files

You can transmit your data files, without using NetView FTP panels, by creating a batch job as shown in [Figure 74 on page 431](#):


```
//STEP0012 EXEC PGM=DVGIFBI                                INVOKE NBT
//*
//DVGLOG      DD SYSOUT=A
//SYSABEND    DD DUMMY
//SYSIN       DD *
FUNCTION=ADD
XMODE=TO
REMNODE=TKSMVS1
SNOTIFY=(X999999.XYZMVS4)
RNOTIFY=(Z000000.TKSMVS1)
SSECURP=(X999999.ROSE)
RSECURP=(Z000000.CLIFF)
WAIT=YES
SFTYPE=LAB
SFILEID='PAY1.D.C01V0001'
RFTYPE=LAB
RFILEID='PAY1.D.C01V0001'
RTDISP=CAT
/*
```

Figure 74. Sample Batch Job Used to Transmit Aggregate Backup Files



Attention: Because you cannot specify a file sequence number with the volume serial number when you are using NetView FTP or transmitting a batch job, you must catalog the data set if you need to transmit a second file of the data file.

Related reading

For more information about how to use the NetView FTP panels, see *NetView File Transfer Program for MVS: User's Guide*.

Issuing the ARECOVER command when using NetView FTP

When you use NetView FTP to transmit aggregate backup files, the following steps must be taken before you issue the ARECOVER command:

1. Ensure that all control file and data file data sets and any accompany data sets are cataloged before issuing the ARECOVER command.
2. Use the XMIT parameter on the ARECOVER command to indicate you are recovering data sets that have been transferred using NetView FTP.
3. *Do not* specify the UNIT and VOLUME parameters when XMIT is specified on the ARECOVER command, as shown in this example:

```
ARECOVER PAY1.C.C01V0001 REPLACE XMIT
```

4. Do not use the AGGREGATE parameter with the XMIT parameter.

ABARS considerations for sequential data striping

DFSMSHsm does aggregate back up and aggregate recover processing for single-stripe and multiple-stripe sequential extended format data sets. To process these data sets, the data mover must be DFSMSdss, and the data sets must be SMS-managed.

Aggregate backup processing

DFSMSHsm ABACKUP processing invokes DFSMSdss to perform a logical data set dump when it encounters a sequential extended format data set—both single-stripe and multiple-stripe—in the INCLUDE list.

If a sequential extended format data set resides on a non-SMS-managed volume, the logical data set dump fails.

Aggregate recovery processing

DFSMSHsm invokes dynamic allocation (SVC 99) to allocate the data sets in the ALLOCATE list:

- During ARECOVER processing, dynamic allocation allocates the data set as sequential extended format or not sequential extended format (regardless of the number of stripes at the backup site) according to what was specified at the recovery site for the DATA SET NAME TYPE attribute in the data class.
- Dynamic allocation also allocates the data set as single-stripe or multiple-stripe sequential extended format (regardless of the number of stripes at the backup site) according to what was specified at the recovery site for the SUSTAINED DATA RATE attribute in the storage class. If GUARANTEED SPACE = YES is specified, the number of volsers or VOLCNTs specified is the number of stripes; the value for SUSTAINED DATA RATE is ignored.
- Dynamic allocation allocates the data set as a data set that is not sequential extended format if the attribute DATA SET NAME TYPE in the data class is defined as (EXTENDED,C), and one of the following conditions is met:
 - SMS is not active at the recovery site.
 - None of the SMS-managed volumes is defined.
 - None of the SMS-managed volumes is connected to an ESCON 3990 model 3.
 - None of required volumes are available.
- Dynamic allocation does *not* allocate a data set if the attribute DATA SET NAME TYPE in the data class is specified as (EXTENDED,R), and one of the following conditions is met:
 - SMS is not active at the recovery site.
 - None of the SMS-managed volumes is defined.
 - None of the SMS-managed volumes is connected to an ESCON 3990 model 3.
 - Insufficient volumes are available to meet the allocation.
 - Insufficient space is available on the selected volumes.

DFSMSdss performs a logical data set restore for a sequential extended format data set as follows:

- A sequential extended format data set that was multiple-stripe when it was backed up can be restored as a multiple-stripe or single-stripe sequential extended format data set, depending on the storage class SUSTAINED DATA RATE value, and the availability of the required volumes.
- Conversely, a sequential extended format data set that was single-stripe when it was backed up can be restored as a multiple-stripe or single-stripe sequential extended format data set, depending on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.
- A data set that was sequential extended format when backed up is recovered as a non sequential extended format but still SMS-managed data set if no SMS volumes supporting sequential extended format data sets are available.
- A data set that was not sequential extended format when it was backed up is always recovered as a data set that is not sequential extended format. DFSMSHsm will not convert a data set that is not sequential extended format to a sequential extended format data set through ARECOVER processing.

If the data class attribute DATA SET NAME TYPE is modified to indicate that the data set should be allocated as a data set that is not sequential extended format, the restore does not occur.

Aggregate roll-off processing

When a new version of an aggregate group is created, ABARS processing ensures that the number of versions kept does not exceed the amount specified by the NUMBER OF VERSIONS attribute in the management class. If NUMBER OF VERSIONS is exceeded, ABARS rolls off (deletes) the oldest version of that aggregate.

Roll-off processing can occur each time a new ABR record is written to the BCDS. Since ARECOVER creates an ABR record if one does not exist, roll-off processing can occur there as well as during ABACKUP.

Roll-off processing does not occur when the VERIFY parameter is used, because no ABR record is created.

When an aggregate group is removed during roll-off processing, its control file name is located in the catalog and a list of volume serial numbers that the control file resides on is requested. When necessary, the volumes are deleted from the RACF HSMABR tape volume list. The control file is uncataloged after the volume serial number list has been exhausted. When the file does not exist in the catalog, it is not an error condition, and the ABR record can still be deleted.

If it is active, installation exit ARCTVEXT is called for each volume being processed, indicating to the exit that the volume being expired is an ABARS tape volume.

This process is repeated for the data files and the instruction and activity log files.

After the control file, data files, and instruction and activity log file have been uncataloged and removed from the HSMABR tape volume set, the ABR record is deleted and processing continues with the next ABR record for the group being processed.

Note: It is possible to code SMS ACS routines that direct ABARS files to DASD. If this is done, and the files get migrated to DASD, an HSM DELETE is issued to remove them during roll-off processing. If that delete fails, the file may remain cataloged with no ABR record pointing to it. If this happens, the data set must be manually deleted. See “Directing ABARS files to DASD” on page 434 for more information.

You can use the SETSYS ABARSDELETEACTIVITY(Y | N) command to specify whether you want DFSMSshm to automatically delete the ABARS activity log that corresponds to the ABACKUP version being rolled off. The deletion occurs during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing, and removes the need to manually manage the ABARS activity logs. The default is N, which specifies no automatic deletion.

Defining GDGs at the recovery site

An alternative way to have GDGs defined at the recovery site is to specify a GDG base name in the ALLOCATE statement and have ABARS predefine the GDG base name during ARECOVER processing. This can be done even if the installation has not specified an associated GDS name in the selection data set. During ARECOVER processing, all GDG base names are defined before restoring any GDSs specified in the aggregate. GDG base names specified in the ALLOCATE statement have the following restrictions:

- They are only defined if they do not already exist at the recovery site.
- The DSCONFLICT parameter of the ARECOVER command does **not** apply to these GDG base names.
- The ARCCREXT installation exit does **not** get control when active for conflicts that arise with GDG base names in the ALLOCATE statement.
- RECOVERNEWNAMELEVEL and RECOVERNEWNAMEALL **do** apply to these GDG base names.

Ensuring backup copies of ARECOVERed SMS-managed data sets exist at the recovery site

When ABARS is used to move data sets to a recovery site, DFSMSdss moves the catalog information for all data sets residing on primary volumes from the disaster site to the recovery site. If backup copies of these data sets are not also moved to the recovery site, the LBACKUP dates in the catalog are not correct when the data sets are ARECOVERed. If these data sets are not updated, then no backup is created for SMS-managed data sets because the LBACKUP dates in the catalog show that valid backup copies exist at the recovery site when, in fact, they do not. This creates a potential data integrity problem if the original data sets are ever lost or damaged.

Two methods can be used to ensure that backup copies of these data sets are made:

- Issue a BACKVOL TOTAL for ARECOVERed volumes. This creates backup copies unless the management class specifies that no backups should be made.
- Use CLIST to allow the ARECOVER activity log to process the data sets, opening them for output so that the change bit is set.

Directing ABARS files to DASD

It is possible to code SMS ACS routines that direct ABARS files to DASD. If this is done, and the files get migrated to DASD, an HSM DELETE is issued to remove them during roll-off processing. If that delete fails, the file may remain cataloged with no ABR record pointing to it. If this happens, the data set must be manually deleted.

Because ABARS is designed to write to tape, space parameters are not passed. When directing ABARS files to DASD, the space parameters must be obtained using the SPACE parameter of the ALLOCxx PARMLIB member. You can verify these parameters by issuing:

```
D ALLOC,OPTIONS
```

Note that changing these parameters may affect any other allocations using the defaults.

By defining DATACLAS and using the PRIMARY, SECONDARY, and AVGREC fields, you can specify how many records will be contained in primary and secondary storage. Your specification can be done in kilobytes (K), megabytes (M), or bytes (U).

In addition, you need to define and assign a management class for ABARS files that will be allocated to DASD using the NOLIMIT operand for the following fields:

- EXPIRE NON-USAGE
- EXPIRE DATE/DAYS
- RET LIMIT

This will ensure that the aggregate backups are not prematurely expired or deleted. The retention period is controlled by the management class specified in the aggregate definition. You may wish to omit the backup attribute, depending on your requirement. If not omitted, DFSMSHsm will take incremental backups of the ABARS files that are allocated to DASD. Note that there may be two separate management classes, one that applies only to the ABACKUP output files, and one that defines how to manage the aggregate group as a whole.

When creating an ABARS aggregate to be directed to DASD, use the NOSTACK parameter. Otherwise, it will fail and you will receive message ARC6055I and a return code of 94. You can either specify NOSTACK on the ABACKUP command, or you can use the SETSYS command to specify NOSTACK for all aggregate backups, as follows:

```
SETSYS ABARSTAPES(NOSTACK)
```


Chapter 9. Making disaster backup copies of DFSMSHsm-owned tape volumes

The ABACKUP and ARECOVER functions described in [Chapter 8, “Aggregate backup and recovery support \(ABARS\),” on page 319](#) provide comprehensive recovery of your data in case of a disaster.

However, you also need protection against damage or loss of your backup and ML2 tape volumes. DFSMSHsm allows you to make copies of your backup and ML2, single-file-format tape volumes for storage at a site different from the computer site or for use at the computer site to recover lost or damaged tape volumes.

In addition to considering tape-capacity usage, you should consider the tape-related implications of disaster backup described below, in deciding whether you want to handle partially full tapes by SETSYS PARTIALTAPE(MARKFULL) or SETSYS PARTIALTAPE(REUSE).

Related reading

For more information about the SETSYS command, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#).

Planning disaster recovery

The premise for disaster recovery is that you will have copies of all your vital data sets stored at a location separate from your computer site and that a computer system of approximately the same capabilities will be available to be used to resume operations. Consider what this implies:

- You must have a copy of all your system control data sets stored at the off-site location. System control data sets change every time maintenance is applied, which for some installations is weekly or biweekly, so having automated scheduled backups are recommended. You can also dump the data sets by command whenever necessary.
- You must have copies of all your vital user data sets. You can use SMS to ensure that all of the vital user data sets are stored on the appropriate DASD volumes. You can provide these copies by using the DFSMSHsm ABACKUP function.
- You must have copies of your ML1 volumes in order to recall any vital data sets that have migrated to ML1. You can use the DFSMSHsm ABACKUP or automatic dump functions to provide these copies of DASD ML1 volumes.
- You must have copies of your ML2 volumes in order to recall any vital data sets that have migrated to ML2. If you use DASD for ML2, you can provide these copies by using the DFSMSHsm ABACKUP or command dump functions. If you use cartridge-type, single-file-format tape for ML2, you can use the duplex tape option when creating the ML2 tapes, or you can use other DFSMSHsm disaster backup facilities as discussed in this topic. ABACKUP can be used for ML2 tape data sets as well.
- You should have copies of your backup volumes in order to reconstruct your user volumes. If you make copies of your backup volumes, you can make fewer automatic dump copies of entire user volumes. You reconstruct the user volumes from the combination of the latest dump copy and the incremental backup versions of data sets made after the dump copy was made. If you use DASD for backup, you can provide these copies by using the DFSMSHsm ABACKUP or command dump functions. If you use cartridge-type, single-file-format tape for backup, you can use the duplex tape option when creating backups, or you can use other DFSMSHsm disaster backup facilities as discussed in this topic.

Note: ABACKUP cannot be used for incremental versions stored on backup DASD. Only full volume dumps can provide offsite copies of data on backup DASD.

- You must have VTOC copy data sets for some cases. The VTOC copy data sets are required when you are restoring data from dump copies combined with incremental backup versions. By dumping ML1 volumes you have VTOC copy data sets available at the recovery site.

- You need a set of instructions for reconstructing the system and resuming operations.

Disaster backup facilities

To provide disaster backup, two things are necessary: (1) a way to make copies of DFSMSHsm-owned tape volumes and to identify the copied volumes to DFSMSHsm; and (2) a way to cause DFSMSHsm to use the copied volumes instead of the original volumes. To fulfill these needs, DFSMSHsm provides the duplex tape option or the TAPECOPY command, and the TAPEREREPL command.

Making copies of DFSMSHsm-owned tape volumes

There are two methods of making copies of DFSMSHsm-owned tape volumes. One method is by using the duplex tape option. The other method is by using the TAPECOPY command.

Duplex tape option

The duplex tape option provides an alternative to TAPECOPY processing for backup and migration cartridge-type tapes. Two tapes are created concurrently with one designated the original, the other the alternate. The intent is that the original tape be kept onsite, while the alternate can either be created in a remote tape library, or taken offsite.

The alternate tape must have the same tape geometry as the original. For example, if the original tape is a 3590-1 tape, the alternate must be also.

The duplex tape option also requires one additional tape drive per writing task. Adequate numbers of drives must be available for the tasking level used by the functions using duplexing during the time periods those functions run. This includes recycle.

With the SETSYS DUPLEX command you can specify that the duplex tape option be used for backup tapes only, migration tapes only, or both backup and migration tapes. If you use duplex creation for both migration and backup and you have been auto scheduling tape copying, you probably want to remove the auto scheduling. If you are duplexing either migration or backup tapes but not both, you can auto schedule tape copies based on the type indicated in the ARC0421I message.

In an SMS environment, the ACS routines can direct the alternate tape to the same or to a different tape library. Different tape libraries can exist at the same location, or a remote location. ESCON technology defines how distant the remote location can be. In a non-SMS environment, the output restrictor is used for both the original and the alternate. ACS filtering can be set up to cause the desired tape allocations to occur.

Tapes created by the duplex tape option are compatible with tapes created by TAPECOPY.

DFSMSHsm initial tape selection selects tapes for the duplex option using the following selection process:

- A partial tape and its alternate are selected.
- If no partial tapes with alternates are found, an empty ADDVOLed volume with a scratch volume as an alternate are selected.
- If there are no empty ADDVOLed volumes, two scratch volumes are selected.

When problems occur that allow an original tape to be created, but prevent the alternate from being created by the duplex tape option, DFSMSHsm automatically schedules internal TAPECOPY processing to ensure that valid alternate tapes exist. If the original tape is demounted before being filled it is marked as duplexed but unavailable for output until the alternate is successfully created. At either the EOV for the original volume or the end of the task, DFSMSHsm schedules TAPECOPY processing to create an alternate tape. If that is unsuccessful, automatic scheduling of internal TAPECOPY processing occurs during secondary space management for migration volumes, and during automatic backup processing on the primary host for backup volumes.

As part of the automatic internal scheduling of TAPECOPY processing, a tape copy needed (TCN) record is written in the OCDS. When the automatic internal TAPECOPY completes successfully, the TCN record

for the source volume (which is the original volume of the duplexed pair) is deleted, and the original tape volume is made available for reuse.

In a SETSYS PARTIALTAPE(REUSE) environment, both the duplexed original and its alternate can be reused (extended). If you have a non-duplexed original with an alternate created by TAPECOPY, only the original can be reused. The alternate is discarded.

The SETSYS DUPLEX command also allows you to specify that alternate and original migration tapes are to be marked full if an error occurs when writing to the alternate tape volume. To do so, enter the SETSYS DUPLEX(MIGRATION(Y ERRALT(MARKFULL))) command with the ERRORALTERNATE(MARKFULL) parameter specified:

```
SETSYS DUPLEX(MIGRATION(Y ERRALT(MARKFULL)))
```

The default for ERRORALTERNATE is CONTINUE. Here, if the alternate tape volume encounters an I/O error, the alternate is demounted and returned to scratch, but DFSMSHsm continues writing to the original. When the writing is completed, DFSMSHsm schedules a TAPECOPY request to create a new alternate tape.

A new flag has been added to the ARCTVEXT installation exit parameter list. When an alternate tape is being replaced and returned to scratch, and if the ARCTVEXT installation exit is active, the flag indicates if the action is the result of a user-issued TAPECOPY command. This information is provided so users are aware that an alternate tape is being returned to scratch, and that the original tape is not.

You can use the LIST and HLIST commands to show duplex tape status and alternate tape volume serial numbers.

Duplex tape does not support SETSYS TAPEUTILIZATION NOLIMIT set for the output unit being duplexed. This is because the alternate tape must match the original tape exactly, and with SETSYS TAPEUTILIZATION set to NOLIMIT this is impossible to guarantee, since either tape could be longer than the other. When TAPEUTILIZATION is set to NOLIMIT, the duplex tape option causes the tapes to reach a logical EOVS when they are 97% full.

The duplex tape option does not support creating duplex copies of ML2 tapes during ARECOVER processing. Instead, one copy of the ML2 tape is recovered by ARECOVER processing, and a TCN record is written in the OCDS. The TCN record causes a TAPECOPY to be scheduled during the next occurrence of secondary space management processing. The TAPECOPY is not scheduled immediately because a TAPECOPY running during the recovery process might not be wanted.

The duplex tape option handles errors in the following way:

- When an attempt is made to allocate and mount a partially full duplexed tape, but the associated alternate volume cannot be found, both the original and alternate are rejected and marked as unavailable for the duration of DFSMSHsm processing.
- When an attempt is made to allocate and mount an empty original tape and a new alternate scratch tape, but the scratch tape cannot be allocated or mounted, the duplex tape option continues to write the original volume. When the original volume (ADDVOLed or scratch) is demounted, DFSMSHsm marks the volume as duplexed and unavailable for output, and schedules an internal TAPECOPY for that volume. DFSMSHsm makes the original volume available after internal TAPECOPY processing successfully creates an alternate tape volume.
- If the duplexed original encounters an I/O error, both the original and alternate tapes are marked full, demounted, and two different tapes are selected. The user data set related to the I/O error is restarted from the beginning on the different tapes.
- If the alternate tape volume encounters an error, duplex tape processing depends on whether you specified the ERRORALTERNATE(MARKFULL) option on the SETSYS DUPLEX(MIGRATION) command. If so, both the original and alternate tapes are marked full if the alternate tape volume encounters an error. This option applies to migration tapes, for both initial migrations and subsequent RECYCLE processing. The option is not available for backup tapes. The ERRORALTERNATE(MARKFULL) option is recommended for installations that use higher capacity tapes and perform tape duplexing for disaster recovery purposes.

If you choose not to use this option, the default error handling is in effect. If the alternate tape volume encounters an error, the alternate is demounted and returned to scratch, but the original continues to be written. The original is marked as duplexed and unavailable for output, and an internal TAPECOPY is scheduled to automatically create an alternate tape. DFSMSHsm makes the original volume available after internal TAPECOPY processing successfully creates an alternate tape volume.

- When the original tape reaches its percent-full capacity, DFSMSHsm must flush all the data to both tapes before performing a FEOV. If natural EOV is reached on either tape, DFSMSHsm marks the tapes as *FULL* and restarts processing of the current data set from its beginning on two new tapes.

TAPECOPY command

The TAPECOPY command allows you to make copies of single-file-format, DFSMSHsm-owned tapes. The copies are known as alternate tapes and are identified in the offline control data set (OCDS) as alternates to the tapes that are copied.

DFSMSHsm provides commands to copy categories of DFSMSHsm-owned tape volumes:

```
TAPECOPY BACKUP
```

```
TAPECOPY MIGRATIONLEVEL2
```

```
TAPECOPY ALL
```

These commands cause DFSMSHsm to make alternate copies of full, single-file-format tapes that do not already have an alternate copy. The TAPECOPY MIGRATIONLEVEL2 and TAPECOPY ALL commands also cause DFSMSHsm to make alternate copies of partially full ML2 tapes if the environment is PARTIALTAPE(MIGRATION(MARKFULL)).

You can make copies of all volumes categorized as either ML2, backup (or both), or of individual tape volumes. You can issue one TAPECOPY command per DFSMSHsm host as long as the requests do not overlap. For example, you can issue a TAPECOPY ML2 command on one host and a TAPECOPY BACKUP command on another.

For individual tape volumes, you can make copies of either partial or full volumes, and of volumes with or without an alternate copy already existing.

The TAPECOPY command provides for copying only cartridge-type, single-file-format tapes. If you have data on multi-file format tapes (such as reel tapes), you can transfer the data onto new cartridge tapes by using the recycle function. These cartridge tapes can then be copied, and use of the reel tapes can be discontinued.

The output or alternate tape volumes used by the tape copy process begin as scratch tape volumes. These scratch tape volumes cannot be volumes that have been added to DFSMSHsm with the ADDVOL command. Using scratch tape volumes not added to DFSMSHsm eliminates backup, dump, or ML2 tape volumes from being used as alternate tape volumes.

DFSMSHsm copies the security information (RACF and expiration date) of the original volume, or you can specify a different expiration date when an alternate volume is created. Because password protection is not copied to an alternate volume, you should use RACF or an expiration-date protection for the alternate volume. If the alternate volume is in the RACF DFSMSHsm tape volume set, the RACF protection is removed from the alternate volume when it is returned to scratch status. If the tape resource class (TAPEVOL) is inactive during a recycle request for a RACF-protected original volume with an alternate, the RACF protection for the alternate will be deleted based on the TTOC information available for the original volume.

Information about the alternate tape volumes is stored in the tape table of contents (TTOC) records of the OCDS. While, over time, a given tape volume might be copied to several alternate volumes, only the most current alternate tape volume reference is stored in the TTOC record.



Attention: Be aware that some tape management products do not protect DFSMSHsm data tapes from being mounted for output in response to a nonspecific mount PRIVAT request, and therefore they can allow DFSMSHsm to write over some of its own tapes. This exposes alternate tapes because DFSMSHsm does not maintain a way to quickly identify an alternate tape as a DFSMSHsm

tape, so there is no check to prevent an alternate from being overwritten. DFSMSrmm does not have this exposure.

Assigning expiration dates to tapes provides only partial protection from inadvertent rewriting over valid data. The general solution is to use policies defined with a function like "vital record specifications" in a tape management product like DFSMSrmm. DFSMSHsm uses tape-volume exit ARCTVEXT to communicate to tape management products, except DFSMSrmm, when a tape can safely be returned to scratch status.

Related reading

For more information about using DFSMSHsm with DFSMSrmm, see [Chapter 16, "Using DFSMSHsm with DFSMSrmm,"](#) on page 591.

Copying individual volumes

You may want to make copies of individual volumes under the following conditions:

- Message ARC0421I, written to the console when a tape volume is marked full, allows making an alternate copy of that volume automatically, if you are not using the duplex tape option to create alternate tapes.
- You have replaced an original volume with an alternate volume. You would then immediately want to make a new alternate volume for protection against loss of what is now the only usable volume.
- An alternate volume is lost so that you must make a new alternate volume, even though it appears (in a TTOC record) that an alternate volume already exists.

When DFSMSHsm makes an alternate volume, it records the alternate volume in the OCDS, thus associating the alternate volume with the original volume. For copying individual volumes, if an alternate volume already exists:

- DFSMSHsm replaces the old alternate volume with the new alternate volume.
- DFSMSHsm issues message ARC0436I.
- No record is kept of the old alternate volume.
- For duplex tapes only, the tape volume exit ARCTVEXT is invoked, if active, to release the previous alternate tape volume through the tape management system. A flag in the ARCTVEXT parameter list indicates that the tape being released is an alternate tape being replaced by a user TAPECOPY, and that the original tape is not being released.
- If SETSYS PARTIALTAPE(MARKFULL) is specified, the original volume is marked full.

The command to make alternate volumes for individual volumes is:

```
TAPECOPY ORIGINALVOLUMES(ovol1,ovol2,...,ovoln)
TAPECOPY ORIGINALVOLUMES(ovol1,ovol2,...,ovoln) -
    ALTERNATEVOLUMES(avol1,avol2,...,avoln)
```

This command copies the volumes specified, creating an alternate tape volume for each. If you use the first form of the command, a default volume serial number of PRIVAT is used for the alternate tape volumes. PRIVAT indicates to DFSMSHsm that a scratch volume be mounted for each alternate tape volume. However, if you use the second form of the command, DFSMSHsm uses the volumes you specify with the ALTERNATEVOLUMES parameter to become the alternate volumes. If you use the ALTERNATEVOLUMES parameter, you must specify exactly the same number of volumes with the ALTERNATEVOLUMES parameter as you specified with the ORIGINALVOLUMES parameter. There is a one-to-one correspondence between the original volumes and alternate volumes that you specify. That is, *ovol1* is copied to *avol1*, *ovol2* is copied to *avol2*, and so forth.

An alternative exists for copying your existing DFSMSHsm-owned volumes. You can issue a LIST command to obtain a list of your ML2 and backup volumes. You can then issue a LIST TTOC command to determine which of the volumes have alternate copies already. Send the LIST TTOC command output to a data set.

Edit the data set to remove those volumes for which you do not want to make an alternate copy, convert the data set to the format required for the input data set, and issue the following command:

```
TAPECOPY INDATASET(volcopy.list.dsname)
```

DFSMSHsm makes an alternate copy of each of the volumes contained in the list of volumes provided by the data set.

Related reading

- For more information about using the LIST command, see [Chapter 14, “Obtaining information from DFSMSHsm,”](#) on page 559.
- For more information about the TAPECOPY command, see [Chapter 68, “TAPECOPY command: Making copies of selected cartridge-type tape volumes,”](#) on page 1227.

Converting tape volumes marked partially full to marked full

The DFSMSHsm default behavior is to mark a tape volume full only when the tape volume reaches the maximum block count specified by the SETSYS command TAPEUTILIZATION parameter. This is known as a "reuse environment." In a reuse environment, issuing a generic TAPECOPY command processes only the tape volumes that are marked full and that do not already have an alternate tape copy. In addition, partially full tape volumes without an alternate tape volume are not selected for tape volume duplexing. Therefore, it is advantageous to convert tape volumes marked partially full in the DFSMSHsm inventory to marked full.

There are two methods for converting tape volumes marked partially full to marked full:

- Use the DELVOL BACKUP(MARKFULL) or DELVOL MIGRATION(MARKFULL) command to mark each partially full tape volume full. This method is immediate and applies to tape volumes used for both volume and data set migration and backup.
- Use the SETSYS command PARTIALTAPE(MARKFULL), PARTIALTAPE(MIGRATION(MARKFULL)), or PARTIALTAPE(BACKUP(MARKFULL)) parameters to operate DFSMSHsm in what is known as a "mark-full environment." In a mark-full environment, DFSMSHsm will mark partially full tape volumes without an associated duplex copy as full when processing a generic TAPECOPY command or at the end of volume migration, backup, or both (depending on the SETSYS setting). However, tape volumes will not be marked full during data set migration or backup. This method is gradual as a partially full tape volume must be selected for volume migration or backup before being marked full.

Related reading

- For more information about using the DELVOL command, see [Chapter 42, “DELVOL command: Removing a volume from DFSMSHsm management,”](#) on page 841.
- For more information about using the SETSYS command, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,”](#) on page 1057.
- For more information about using the TAPECOPY command, see [Chapter 68, “TAPECOPY command: Making copies of selected cartridge-type tape volumes,”](#) on page 1227.

Replacing an original with an alternate tape volume

If an original ML2 or backup volume is lost or damaged, the volume can be replaced with its alternate volume:

1. To determine if an alternate tape volume exists for the original volume, use the LIST TTOC(*volser*) command.
2. Issue the TAPERPL ORIGINALVOLUMES(*ovol*) command to replace the original volume with the alternate. After the command completes, the alternate tape volume reference in the TTOC record replaces the original volume reference.

3. Issue the TAPECOPY ORIGINALVOLUMES(*newovol*) command to make a new copy of the volume (using the volume serial number of the old alternate volume) to provide a new alternate volume.

When the TAPERREPL command successfully replaces a tape volume, DFSMSHsm issues the ARC0427I message and (with two exceptions) invokes the tape-volume exit ARCTVEXT to indicate to a tape management product that the original volume can be returned to scratch. The two exceptions are when you are using TAPERREPL with the DISASTERALTERNATEVOLUME optional subparameter, or when your tape management product is DFSMSrmm.

For duplex tapes, after the original volume is replaced, it is marked full and is no longer a duplex tape.

Creating duplex tapes off site

With the duplex tape option, you can write the alternate volumes directly to a remote tape library. ESCON technology defines how distant the remote location can be.

When you process the RECYCLE command, DFSMSHsm invalidates the alternate tapes at the same time that it invalidates the tapes from which they were copied. Invalidated alternate tapes are returned to the scratch pool at their respective sites.

Sending alternate tapes off site

To use alternate tapes for disaster backup by sending them to a safe site, you must determine what tapes to send away and determine when to bring the tapes back and reuse them. Use the following procedure to send alternate tapes offsite:

1. Issue the TAPECOPY command for your non-duplexed tapes (backup, migration, or all, as you desire).
2. Issue LIST TTOC to determine which volumes have alternate copies.
3. Send the alternate tapes off site.
4. When you process the RECYCLE command, DFSMSHsm invalidates the alternate tapes at the same time that it invalidates the tapes from which they were copied. Both the original and the alternate tape are passed to the ARCTVEXT installation-wide exit.
5. Request the return from the offsite location of the alternate tapes that have been invalidated, and return the tapes to the scratch pool.

Managing alternate volumes

DFSMSHsm provides full support for producing and managing one alternate tape for each volume in the categories of ML2 and backup tapes. As shown above, this includes using the duplex tape option, or using a generic TAPECOPY command to copy full tapes not already having a copy, and optionally interfacing to the DFSMSHsm installation exit ARCTVEXT to return emptied tapes to scratch status.

Examples

Although it is possible for you to make multiple alternate tapes, that comes with the price of doing additional management of them, as detailed in the following two examples:

1. If you issue the TAPECOPY command to create an alternate tape for an original volume not marked full, then the original will likely be extended later.

When entries are being added to a base TTOC or when a tape is marked full for reasons other than a DELVOL MARKFULL command, any existing alternate volume is no longer valid. If, in either case, the tape already has an alternate volume, then message ARC0441I is issued to the console and appropriate activity log, and the existing alternate is removed. DFSMSHsm does not take exit ARCTVEXT, unless it was a duplexed alternate, since it is not known whether the alternate tape should be returned to scratch status. If it was a duplexed alternate, DFSMSHsm takes the ARCTVEXT installation exit, doesn't get rid of the alternate, and doesn't issue the ARC0441I message.

2. If you issue a specific TAPERREPL command specifying the alternate volume to replace the original tape, and the specified alternate is different from the alternate volume DFSMSHsm knows about, then

message ARC0439I is issued to the submitter and the command activity log. The alternate previously known to DFSMSHsm is not retained. It is also not passed to the ARCTVEXT exit (although the original is) because DFSMSHsm does not know whether you want it to be returned to scratch status.

For duplex tapes, after the original volume is replaced, it is marked full and is no longer a duplex tape.

The first example above is an instance where DFSMSHsm directs a specific message to the console so that if automatic operations are desired, they can be implemented based on intercepting a console message.

If an ML2 or backup tape is deleted using the DELVOL command, DELVOL processing issues message ARC0420I for the alternate volume recorded by DFSMSHsm.

Disaster alternate volumes

DFSMSHsm provides a way—once you have created alternate tapes or have duplexed alternate tapes—to use those alternates in place of the original tapes for doing recalls or recovers.

If an actual disaster happens at your home site or if you are running a disaster recovery test, you should proceed as follows to establish operations at your recovery site:

1. Bring up the system by restoring the system packs, libraries, and DFSMSHsm CDSs.
2. If your strategy includes dump tapes, set the DEFINE command at the recovery site to limit DFSMSHsm recovery to those dump classes that are present there. It is likely that the CDS indicates the dump classes at the recovery site are those available at the home site.
3. Start DFSMSHsm, and restore the required data sets from DASD.
4. If you have created alternate copies of DFSMSHsm-owned tape volumes and the alternate volumes are present at the recovery site, run the TAPEREPL command to convert all of DFSMSHsm's information about the original volume serial number to the alternate volume serial number. Since this process is very time-consuming, a significantly reduced form of the TAPEREPL command can be run during the earliest stages of bringing up a recovery site. This command consists of TAPEREPL, followed by required parameters, followed by DISASTERALTERNATEVOLUMES, abbreviated to DAVOLUMES. For example:

```
TAPEREPL ALL DAVOLUMES
```

When the DAVOLUMES parameter is used, only base OCDS TTOC records of ML2 and backup tapes that have alternate volumes are updated as having *disaster alternate volumes*.

The designation of "disaster alternate volumes" can help distinguish between alternate volumes initially present at the recovery site, and alternate volumes created later at the recovery site.

Note: When the DAVOLUMES option is used, DFSMSHsm does *not* invoke the tape-volume exit ARCTVEXT, since the original tape is not being totally replaced. For example, if this is done for a disaster recovery test, the resulting control data sets may be discarded after the test, and the original tapes again used at the home site.

DFSMSHsm invokes ARCTVEXT when the complete replacement is done, using TAPEREPL without the DAVOLUMES option.

5. Before running customer jobs, issue the SETSYS command with the DISASTERMODE parameter. For example:

```
SETSYS DISASTERMODE
```

One approach would be to put the SETSYS DISASTERMODE command in a startup ARCCMDxx PARMLIB member that is used exclusively for starting DFSMSHsm in disaster mode.

While DFSMSHsm is running in disaster mode, recalls and recovers from tape volumes that are marked as having disaster alternate volumes will substitute the volume serial number of the disaster alternate volume from the TTOC record for the original volume serial number that appears in the data set records.

6. If you use TAPEDELETION(HSMTAPE) or PARTIALTAPE(REUSE) options then the inventory will likely contain empty and partially filled backup and migration tapes that were destroyed at the home site. These tapes will be selected for output at the recovery site. To prevent DFSMSHsm from requesting tapes that are not available, identify these tapes using the LIST TTOC SELECT(NOTFULL) command, then mark the tapes full with the MARKFULL option of the DELVOL command:

```
DELVOL volser BACKUP(MARKFULL)
DELVOL volser MIGRATION(MARKFULL)
```

7. As jobs run at the recovery site and new data sets are created, backup versions and migration copies are created. As tapes are filled, they may be copied. These “new” alternate copies are not considered “disaster alternate volumes” and will not be substituted for the original volumes during recall or recover.
8. If you have finished running a disaster recovery test or you are finished using the alternates, reset the Disaster Alternate Volume flag in the TTOCs by issuing the following command:

```
TAPEREPL ONLYDISASTERALTERNATES(RESET)
```

If you were performing a disaster recovery test, you may want to make scratch tapes out of any DFSMSHsm output tapes created at the recovery site. You may leave the alternate volumes for future disaster recovery.

9. If you are recovering from a true disaster, you now need to plan for the return to your home site. You need to convert the disaster alternate volumes to original volumes. You can issue:

```
TAPEREPL ONLYDISASTERALTERNATES
```

to replace all, and only, those original volumes identified as having disaster alternate volumes. The command HOLD TAPEREPL may be used if the TAPEREPL command needs to be stopped.

10. You can also replace specific original volumes with their disaster alternates. First, issue

```
LIST TTOC SELECT(DISASTERALTERNATEVOLUMES)
```

to identify those tape volumes with disaster alternate volumes, and then issue the TAPEREPL command without the DAVOLUMES parameter. For example, the command:

```
TAPEREPL OVOLS(TKS001,TKS002,TKS003)
```

where TKS001, TKS002, and TKS003 are volume serial numbers that were among those returned from the LIST TTOC command—will replace these specific original volumes with their disaster alternates.

These newly converted original tapes need to be copied now for future disaster recovery.

Note: The AUDIT MEDIACONTROLS processing will fail the audit of a tape volume that is marked as having a disaster alternate volume.

Related reading

- For more information, see Chapter 69, “TAPEREPL command: Replacing cartridge-type tape volumes with their alternate volumes,” on page 1235.
- For more information, see Chapter 42, “DELVOL command: Removing a volume from DFSMSHsm management,” on page 841.

Considerations when using disaster alternate volumes

When doing a generic RECYCLE or TAPECOPY, a tape volume marked as having a disaster alternate volume that is otherwise selected is skipped. A specific volume RECYCLE or TAPECOPY for such a

Making disaster backup copies of DFSMSHsm-owned tape volumes

volume fails; message ARC0424I is issued for a TAPECOPY failure and message ARC0445I is issued for a RECYCLE failure.

Chapter 10. Other availability management considerations

The following additional topics are related to each of the aspects of availability management: backup, dump, recover, and restore.

Additional considerations for backup

Additional topics related to backup are:

- Possible tape volume contention in single and multiple DFSMSHsm-host environments
- Recover tape takeaway contention
- Implications of redefining the backup cycle
- Inline backup of data sets in the middle of a job
- Preventing backup processing
- GUARANTEED SPACE attribute and VSAM data sets

Backup volume contention resolution

DFSMSHsm resolves volume contention among the different tasks that can request the same tape volume. The data set backup by command function may hold a tape for an extended period of time. Consequently, DFSMSHsm allows a backup tape to be taken away from a backup or recycle task to satisfy a recover request. Table 45 on page 445 summarizes the contention possibilities and scope of what DFSMSHsm does to resolve volume contention for both single and multiple DFSMSHsm-host environments.

Table 45. Volume Contention in Single and Multiple DFSMSHsm Host Environments

Function That Has Volume	Function That Needs Volume	Result
Backup	Backup	Backup selects a new target volume.
Recover	Backup	Backup selects a new target volume.
Recycle (source)	Backup	Backup selects a new target volume.
Recycle (target)	Backup	Backup selects a new target volume.
Tapecopy (source)	Backup	Backup selects a new target volume
Backup (data set)	Recover	Recover initiates a tape takeaway and waits for the tape to become available or to reach the maximum delay time. Data set backup completes any data set that is currently being processed, releases the tape, and selects a different target tape.
Backup (volume)	Recover	Recover fails.
Recover	Recover	In a single DFSMSHsm-host environment, there is no contention. In a multiple DFSMSHsm-host environment, the Recover fails.
Recycle (source)	Recover	Recover initiates a tape takeaway and waits for the tape to become available or to reach the maximum delay time. When recycle processing is complete for the data set that is being processed, the allocated volume is released and a new volume is selected for recycle processing.
Recycle (target)	Recover	Recover initiates a tape takeaway and waits for the tape to become available or to reach the maximum delay time. When recycle processing is complete for the data set that is being processed, the allocated volume is released and a new volume is selected for recycle processing.

Table 45. Volume Contention in Single and Multiple DFSMSHsm Host Environments (continued)

Function That Has Volume	Function That Needs Volume	Result
Tapecopy (source)	Recover	Recover initiates a tape takeaway and waits for the tape to become available or to reach the maximum delay time. If tapecopy does not complete before the maximum time, tapecopy fails. Either way, recover takes the tape.
Backup	Recycle source	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recover	Recycle source	Recycle of this volume fails. Recycle of other eligible volumes continues.
Recycle (source)	Recycle source	In a single DFSMSHsm-host environment, there is no contention. In a multiple DFSMSHsm-host environment, Recycle skips the volume because another host is recycling it.
Recycle (target)	Recycle source	In a single DFSMSHsm-host environment, Recycle selects a new target volume. In a multiple DFSMSHsm-host environment, Recycle skips the volume, because another host is recycling it.
Tapecopy (source)	Recycle source	In a single DFSMSHsm-host environment, Recycle skips the volumes, and recycle of other volumes continues. In a multiple DFSMSHsm-host environment, Recycle skips the volumes, and recycle of other volumes continues.
Backup	Recycle target	Recycle selects a new target volume.
Recover	Recycle target	Recycle selects a new target volume.
Recycle (source)	Recycle target	Recycle selects a new target volume.
Recycle (target)	Recycle target	In a single DFSMSHsm-host environment, there is no contention. In a multiple DFSMSHsm-host environment, Recycle selects a new target volume.
Tapecopy (source)	Recycle target	Recycle selects a new target volume.
Backup	Tapecopy source	Tapecopy of this volume fails. Tapecopy of other eligible volumes continues.
Recover	Tapecopy source	Tapecopy of this volume fails. Tapecopy of other eligible volumes continues.
Recycle (source)	Tapecopy source	Tapecopy of this volume fails. Tapecopy of other eligible volumes continues.
Recycle (target)	Tapecopy source	Tapecopy of this volume fails. Tapecopy of other eligible volumes continues.
Tapecopy (source)	Tapecopy source	No contention.

Implications of redefining the backup cycle

If you redefine the backup cycle, you might leave daily backup volumes unused.

Example: Assume the following conditions:

- You have specified the DEFINE BACKUP(YNNYNNY) command. Automatic backup occurs on days 1, 4, and 7.
- You have assigned volume A to day 1, volume B to day 4, and volume C to day 7.
- You later specify DEFINE BACKUP(YNYN). Automatic backup occurs on days 1 and 3.

These are the results:

- Volume A remains assigned to day 1.
- Volume B is still assigned to day 4, but automatic backup no longer runs on day 4. Therefore, DFSMSHsm does not use volume B unless someone issues a BACKVOL command on day 4 in the backup cycle.

- DFSMSHsm does not use volume C any more because it was assigned to day 7 and the new cycle has only four days.

To remove volume C (and volume B, if you want) from its day in the backup cycle and to make it an unassigned backup volume, use the DELVOL command. However, do not specify the PURGE parameter, because any backup version on the tape backup volume would be lost to DFSMSHsm and the volume information for the tape or DASD volume would be deleted. The UNASSIGN parameter of the DELVOL command does not destroy any of the backup versions already contained on the volume or prevent DFSMSHsm from accessing these backup versions during recovery, but it allows the volume to be used again if necessary.

In redefining the backup cycle, you can also use the CYCLESTARTDATE subparameter to redefine the starting date for the cycle.

Inline backup

Inline backup allows you to request a back up of a data set or UNIX file in a batch environment. Inline backup is a way of overcoming the problems of executing the terminal monitor program (TMP) in the background in order to issue HBACKDS commands.

When using inline backup, you can use the JCL data set reference capabilities. This allows you to specify a relative generation number for a GDG. Also, in executing a batch job with TMP, if a data set is created (DISP=NEW), an exclusive enqueue is done on the data set name, and the enqueue is not released until the last job step referring to that data set has ended. If a job step after the TMP job step backward-references a newly created data set, the HBACKDS command fails.

Inline backup allows you to back up data sets or UNIX files in the middle of a job. You add a new step to a job by specifying the inline backup module as the program to execute. You specify the data sets or UNIX files being backed up on specially named DD statements. Inline backup extracts the data set or UNIX file names associated with these DDnames and then performs the back up.

You can use the ARCINBAK program to make a quick backup that can be used in a later step (of the same job). This quick backup enables a restart in the middle of a job stream. Because it is mandatory that the target backup volume is available to perform a recovery even if a continuous backup load exists, DFSMSHsm has a backup tape takeaway capability from data set or UNIX file backup.

The results of each attempted data set or UNIX file backup can be written to a specially named output data set. SNAP dumps of storage relating to inline backup can be written to a specially named output data set for certain error conditions.

The following data sets are not supported:

- Uncataloged data sets
- Backup-while-open data sets
- Non-SMS-managed VSAM data sets that are open for processing by another task at the time of inline backup processing

Related reading

- For more information about how to start the ARCINBAK program, see [“Invoking inline backup” on page 447](#).
- For more information about how to start the ARCINBAK program, see [z/OS DFSMSHsm Managing Your Own Data](#).

Invoking inline backup

To use inline backup:

1. Include a step at the correct place in the job to execute a program called ARCINBAK.

2. Specify the name of the data set or UNIX file to be processed by including a valid DD statement with a DDNAME of BACKxxxx (where xxxx is 1 to 4 characters). Do this for each data set or UNIX file you want backed up.

Note:

1. Only cataloged data sets are supported for inline backup. If volume and unit information is specified on the DD statement, an uncataloged data set is assumed, and the data set is not processed.
2. ARCINBAK does not support data sets allocated with any of the following three dynamic allocation options: XTIO, UCB NOCAPTURE, and DSAB above the line, except when the calling program supplies an open DCB.
3. A UNIX path must be specified with the PATH= parameter. Since the JCL PATH statement only supports absolute pathnames up to 255 characters in length, this is the maximum length filename that ARCINBAK can process.

Any form of the data set name is acceptable to ARCINBAK. Specification of a relative generation number for a GDG or a reference to a DDname in a previous step is acceptable. A DD statement with a DDNAME of ARCPRT can be optionally included in the ARCINBAK program job step. This DD statement defines a SYSOUT data set to contain messages about data sets for which a backup was attempted.

The possible messages are as follows:

```
BACKUP FOR (DSNAME) SUCCESSFUL
DDNAME (DDNAME) NOT BACKED UP, UNABLE TO GET ASSOCIATED JFCB
BACKUP FOR (DSNAME) FAILED, DATA SET NOT CATALOGED
BACKUP FOR (DSNAME) FAILED, RC = (RETURN CODE), REAS = (REASON CODE)
```

At the completion of the ARCINBAK program job step, one of the following return codes is displayed:

- 0 — All backups are successful.
- 4 — No DDnames prefixed with BACK are found or the JFCB associated with a DDname cannot be found.
- 8 — Backup of an uncataloged data set has been attempted. Uncataloged data sets are not supported by inline backup.
- 12 — Backup of a data set failed. One of the following reason codes is also displayed:

RC100 : DFSMSHsm is not active; request rejected.

RC1–RC99 : For message return codes and reason codes, see message ARC13XXI, where XX is the return code.

A DD statement with a DDNAME of ARCSNAP can be included in the ARCINBAK program job step. This DD statement defines a SYSOUT data set for the output from a SNAP macro. The following conditions cause a SNAP dump of storage relating to inline backup:

- MWE storage can be dumped when an error occurs in issuing the service call.
- A complete image of the task storage can be dumped when no DDnames or data sets are processed.

A job stream example of data sets that can and cannot be backed up by ARCINBAK in a batch environment is shown in [Figure 75 on page 449](#).


```
//JOBNAME JOB . . . ,USER=USERID,PASSWORD=USERPSWD
//STEP1 EXEC PGM=USERPGM
//SYSPRINT DD SYSOUT=A
//DSET1 DD DSN=USERID.N03.GDG(-1),DISP=OLD
//DSET2 DD DSN=USERID.N03.PSFB,DISP=OLD
//DSET3 DD DSN=USERID.N04.PSFB,DISP=OLD
//DSET4 DD DSN=USERID.N03.KSDS,DISP=OLD
/*
//STEP2 EXEC PGM=ARCINBAK, PARM=('TARGET=(TAPE)',
// 'CC=(PREFERRED,PHYSICALEND)')
//ARCPRINT DD SYSOUT=A
//ARCSNAP DD SYSOUT=A
/*
//* -----
//* BACKUP OF GDG DATA SET SHOULD BE SUCCESSFUL.
//* -----
//BACK01 DD DSN=*.STEP1.DSET1,DISP=SHR
//* -----
//* BACKUP OF NON-VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----
//BACK02 DD DSN=*.STEP1.DSET2,DISP=SHR
//* -----
//* BACKUP OF VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----
//BACK03 DD DSN=*.STEP1.DSET4,DISP=SHR
//* -----
//* BACKUP OF GDG DATA SET SHOULD BE SUCCESSFUL.
//* -----
//BACK04 DD DSN=USERID.N01.GDG.G0001V00,DISP=SHR
//* -----
//* BACKUP OF NON-VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----
//BACK05 DD DSN=USERID.N01.PSFB,DISP=SHR
//* -----
//* BACKUP OF UNCATALOGED DATA SET SHOULD FAIL.
//* -----
//BACK06 DD DSN=USERID.N02.UNCAT,VOL=SER=VOL003,UNIT=3390,DISP=SHR
//* -----
//* BACKUP OF VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----
//BACK07 DD DSN=USERID.N01.KSDS,DISP=SHR
//* -----
//* BACKUP OF OPEN IN-USE VSAM DATA SET SHOULD BE SUCCESSFUL.
//* -----
//BACK08 DD DSN=USERID.N02.KSDS,DISP=SHR
//* -----
//* BACKUP OF RACF PROTECTED NON-VSAM DATA SET
//* BY AN UNAUTHORIZED USER SHOULD FAIL.
//* -----
//BACK09 DD DSN=USERXX.N02.PSFB,DISP=SHR
//* -----
//* DDNAME PREFIX OTHER THAN BACK IS NOT PROCESSED.
//* -----
//BAK01 DD DSN=*.STEP1.DSET3,DISP=SHR
//* -----
//* DDNAME PREFIX OTHER THAN BACK IS NOT PROCESSED.
//* -----
//BAK02 DD DSN=USERID.N02.PSFB,DISP=SHR
/*
```

Figure 75. Example of Inline Backup Job Stream

The return code from executing the inline backup job stream example of the ARCINBAK program is 12, and the resulting messages for each DDname prefixed with BACK are shown in [Figure 76 on page 449](#):

BACKUP FOR USERID.N03.GDG.G0001V00	SUCCESSFUL
BACKUP FOR USERID.N03.PSFB	SUCCESSFUL
BACKUP FOR USERID.N03.KSDS	SUCCESSFUL
BACKUP FOR USERID.N01.GDG.G0001V00	SUCCESSFUL
BACKUP FOR USERID.N01.PSFB	SUCCESSFUL
BACKUP FOR USERID.N02.UNCAT	FAILED, DATA SET NOT CATALOGED
BACKUP FOR USERID.N01.KSDS	SUCCESSFUL
BACKUP FOR USERID.N02.KSDS	SUCCESSFUL
BACKUP FOR USERXX.N02.PSFB	FAILED, RC = 039, REAS = 008

Figure 76. Example of Messages Resulting from Inline Backup Job Stream

Preventing backup processing when using the HOLD BACKUP command

You can use the HOLD BACKUP command to prevent all backup processing, prevent only automatic backup processing, or prevent fast replication backup processing. If you do not want any backup processing at all (for volume, data sets, and auto backups), issue the HOLD BACKUP command by itself. You can specify the AUTO subparameter to prevent only automatic backups from running. You can specify the DSCOMMAND subparameter to prevent only data set backups from running.

Note:

1. The DSCOMMAND option applies to the BACKDS, HBACKDS, ARCHBACK, and ARCINBAK data set backup requests and to backups that are performed as INUSE RETRY from automatic backup.
2. The HOLD ALL command prevents all DFSMSShm functions (except logging) from running.
3. If you are using SETSYS BACKUP parameters or a data set backup installation exit (ARCBDEXT) to direct DFSMSShm to retry backup requests that fail because data sets are in use (see [“Specifying how DFSMSShm should respond to a data set in use”](#) on page 191), and if DFSMSShm has scheduled any such retries during automatic backup, those backup retries will not be affected by the HOLD BACKUP(AUTO) command. You will need to use HOLD BACKUP(DSCOMMAND).

The following command prevents processing of all data set backups (BACKDS, HBACKDS, ARCHBACK, and ARCINBAK), and it prevents the processing of all backups that would be the result of INUSE RETRY from automatic backup. All WAIT requests fail, and control returns to the requester.

```
HOLD BACKUP(DSCOMMAND)
```

The following command prevents processing of data set backups that are directed to ML1 DASD. All WAIT requests that target DASD fail, and control returns to the requester.

```
HOLD BACKUP(DSCOMMAND(DASD))
```

The following command prevents processing of data set backups that are directed to tape. All WAIT requests that target tape and all WAIT requests for large data sets that do not specify a target fail. Control returns to the requester.

```
HOLD BACKUP(DSCOMMAND(TAPE))
```

The following command prevents processing of automatic backups, data set backups that are directed to tape, and all WAIT requests for large data sets that do not specify a target. However, data set backups to ML1 DASD are not affected.

```
HOLD BACKUP(AUTO DSCOMMAND(TAPE))
```

The following command prevents processing of automatic backups and data set backups that are directed to ML1 DASD. Data set backups to tape are not affected.

```
HOLD BACKUP(AUTO DSCOMMAND(DASD))
```

During automatic or volume backup, you can specify whether the holding action occurs immediately at the end of processing each data set or at the end of processing each volume currently being backed up. If you hold backup at end of volume, it can take some time for all of the volumes that are being processed by the different backup tasks to complete and the hold to take full effect. If you hold backup at the end of data sets, DFSMSShm resumes data set backup with the next queued entry when the hold is released. However, for automatic backup, the volume interrupted by the HOLD at end of data set remains eligible for processing if automatic backup is restarted.

The holding action begins when an MWE is selected for processing, not when the HOLD command is entered. If you enter a wait-type backup request and the backup function is held, DFSMSShm fails the wait-type backup request because the hold backup function gives no indication when the backup function will be available.

When a non-wait-type backup request is issued and the appropriate backup function is held, the request remains on the queue and is processed when the appropriate backup request function is released. The held backup processing function can be released with the RELEASE BACKUP command.

Related reading

For more information about other ways to prevent DFSMSShm from processing, see [“Preventing the processing of data sets during space management”](#) on page 149.

GUARANTEED SPACE attribute

The storage class attribute of GUARANTEED SPACE allows the volumes on which an SMS data set is to reside to be chosen explicitly when the data set is created. If GUARANTEED SPACE is used and if any of the volumes specified are not part of one of the storage groups to which the storage class maps, the data set creation fails. The use of GUARANTEED SPACE requires some special considerations if DFSMSHsm is to manage data sets assigned to such a storage class.

DFSMSHsm is used as the data mover on partitioned data sets, catalogs, and other data sets that are not supported by DFSMSdss. When DFSMSHsm is the data mover, the Access Method Services functions of EXPORT and IMPORT back up and recover VSAM data sets. The IMPORT function redefines a VSAM data set by using a copy of the catalog information saved in the EXPORT copy of the data set. This defines the request explicitly specifies the volumes from which the data set was exported. The explicit specification of the volume affects the way data sets are imported with the GUARANTEED SPACE attribute. Likewise, for non-VSAM data sets, VTOC Data Set Services uses the volume serial number passed by DFSMSHsm if the storage class is GUARANTEED SPACE.

Results of recovering GUARANTEED SPACE data sets

With DFSMSHsm as the data mover, data sets recovered with the GUARANTEED SPACE attribute are normally placed on the volume from which they were backed up. Consider the effects on a backed up, multiple-volume VSAM data set. A subsequent recover preserves the volume-to-component relationship that originally existed as long as the data set remains SMS-managed. For a data set in a GUARANTEED SPACE storage class, the relationship includes returning the data set to the same volumes from which it was backed up. Thus, if a data set has been originally defined with the data component on one volume and the index component on another, a backup and recover sequence preserves the relationship between the components and the volumes. Similarly, if a key range data set has three key ranges on three separate volumes, the relationship is preserved.

Problems that can occur

In an environment in which the storage groups and storage classes do not change, the preceding results are adequate to allow DFSMSHsm to manage the data sets. However, you can expect problems in an environment that is subject to storage group reconfiguration or subject to changes in the mapping of data sets to storage classes. These problems can discourage the use of DFSMSHsm space management for these data sets and might require special contingency procedures for availability management. The problems that occur relate to:

- Changes in the storage group
- Changes in the storage-class-to-data-set mapping
- Conversions of data sets to the GUARANTEED SPACE storage class

Changes in the storage group

Consider the consequence of making a specific volume request in a storage group configuration that has changed. Assume that a data set with the GUARANTEED SPACE storage class has been backed up. Sometime later, one or more of the volumes on which the data set has resided are removed from the storage groups that contain GUARANTEED SPACE data sets. A subsequent recover of the data set fails because at least one of the requested volumes is no longer part of a storage group for a GUARANTEED SPACE request. The result is the same for both single-volume data sets and multiple-volume VSAM data sets if the specific volume required is unavailable in the necessary storage group. Use of EAV volumes in mixed environments where the volume cannot be brought online to all systems can also result in problems recalling/recovering data sets that belong to a Storage Class having the Guaranteed Space attribute.

If a storage group is changed to remove volumes that data sets migrated from, you must do one of the following:

- Change the ACS storage class routine so that a storage class with GUARANTEED SPACE is not assigned to the data set.

- Change the volume configuration again so the storage group contains the volumes from which the data set was backed up.
- Change the set of storage groups to which the GUARANTEED SPACE storage class maps to include one that already has the necessary volumes defined. This step applies only if the volumes are already SMS-managed.
- Use the RECOVER command with the FORCENONSMS and the VOLUME parameters. This RECOVER command causes the data set to be recovered to a non-SMS-managed volume as a non-SMS-managed data set. You can then move, copy, or convert the data set as necessary.

Note: If the data set was multiple-volume when it was backed up, it is recovered to a single volume.

Changes in the storage-class-to-data-set mapping

Consider the consequence of making a specific volume request in a storage class mapping that has changed. Assume that a data set is defined and one of the following conditions occur:

- The ACS routines do not assign the data set to any storage class.
- The ACS routines assigns the data set to a storage class that does not have the GUARANTEED SPACE attribute.

Later, assume one of the following changes occur:

- The ACS routine is changed so that it assigns the data set to a storage class with the GUARANTEED SPACE attribute.
- The original storage class is changed to include the GUARANTEED SPACE attribute.

Now, when you recover this data set, the recovery might fail because the requested volumes are not in the storage groups that receive GUARANTEED SPACE requests.

If an SMS-managed data set's original storage class is changed to specify the GUARANTEED SPACE attribute before the data set is recovered, you must do one of the following:

- Change the ACS storage class routine so that a storage class without GUARANTEED SPACE is assigned to the data set.
- Change the storage class of the data set to a storage class without GUARANTEED SPACE, without recalling the data set, by using the IDCAMS ALTER command.
- Change the volume configuration in the appropriate storage groups to contain the volumes from which the data set was backed up. The volumes might need to be converted to SMS management first. It can be difficult to determine all of the volumes involved. DFSMSHsm does not save all of the original volume serial numbers, only that of the volume on which the base data component resided.
- Change the set of storage groups to which the GUARANTEED SPACE storage class maps to include one that already has the necessary volumes defined. This step applies only if the volumes are already SMS-managed.
- Use the RECOVER command with the FORCENONSMS and the VOLUME parameters. This RECOVER command causes the data set to be recovered to a non-SMS-managed volume as a non-SMS-managed data set. You can then move, copy, or convert the data set as necessary.

Note: If the data set was multiple-volume when it was backed up, it is recovered to a single volume.

Conversions of data sets to GUARANTEED SPACE

If the data set was non-SMS-managed when it was backed up and is assigned a storage class with GUARANTEED SPACE during recover, it is assigned to a volume in a storage group that contains GUARANTEED SPACE data sets.

If the data set was SMS-managed when it was migrated but did not have GUARANTEED SPACE, the volumes from which it was backed up might not be in a storage group that receives GUARANTEED SPACE requests. To avoid errors, you must do one of the following before the recover:

- Change the target storage class with GUARANTEED SPACE to map to a storage group that contains the original volumes.

- Add the original volumes to a storage group mapped to by the target storage class with GUARANTEED SPACE.

Data set or UNIX file backup with concurrent copy

The data set or UNIX file backup function allows the use of either concurrent copy (or UNIX file Snapshot) or a standard backup method for copying both SMS-managed and non-SMS-managed data sets or UNIX files. Use the concurrent copy (CC) keyword to specify that you want concurrent copy (or UNIX file Snapshot for UNIX files), rather than standard backup methods, to back up your data sets by command.

Note: For SMS data sets, you do not have to use the CC keyword. Use the CC keyword only if you want to override SMS concurrent copy specifications.

To control the use of concurrent copy (CC) for a data set (or the use of UNIX file Snapshot processing for a UNIX file), several options are available for the data set backup function. If the options are omitted, DFSMSHsm backs up the data set or UNIX file using the management class attributes for SMS-managed data. The issuer of the backup will not receive notice until physical completion. For non-SMS managed data sets or UNIX files, DFSMSHsm uses the standard backup method if the concurrent copy option is omitted.

The following keywords can be specified in the CC parameter of HBACKDS and BACKDS commands, ARCHBACK macro or the ARCINBAK program and control the concurrent copy usage:

REQUIRED

DFSMSHsm uses concurrent copy to backup the data set. If virtual concurrent copy is unavailable DFSMSHsm uses cachebased concurrent copy . If both types of concurrent copy are unavailable, the backup fails. When processing a UNIX file, UNIX File SnapShot processing will be used. If UNIX File SnapShot processing is unable to be used, the backup fails.

PREFERRED

DFSMSHsm uses concurrent copy to backup the data set. If virtual concurrent copy is unavailable, DFSMSHsm uses cache-based concurrent copy. If both types of concurrent copy are unavailable, the data set backup continues as if the CC keyword were not specified. When processing a UNIX file, UNIX File SnapShot processing will be used. If UNIX File SnapShot processing is unable to be used, the backup continues as if the CC keyword were not specified.

STANDARD

DFSMSHsm backs up the data set without using concurrent copy (the default). Control returns to the requester at the physical end of the copy. When processing a UNIX file, file is processed without using UNIX File SnapShot. Control returns to the requester at the physical end of the copy.

VIRTUALREQUIRED

DFSMSHsm uses virtual concurrent copy to backup the data set. If virtual concurrent copy is unavailable, the backup fails. When processing a UNIX file, UNIX File SnapShot processing will be used. If UNIX File SnapShot processing is unable to be used, the backup fails.

VIRTUALPREFERRED

DFSMSHsm uses virtual concurrent copy to backup the data set. If virtual concurrent copy is unavailable, the data set backup continues as if the CC keyword was not specified. When processing a UNIX file, UNIX File SnapShot processing will be used. If UNIX File SnapShot processing is unable to be used, the backup fails.

CACHEREQUIRED

DFSMSHsm uses cache-based concurrent copy to backup the data set. If cache-based concurrent copy is unavailable, the backup fails. When processing a UNIX file, UNIX File SnapShot processing will be used. If UNIX File SnapShot processing is unable to be used, the backup fails.

CACHEPREFERRED

DFSMSHsm uses cache-based concurrent copy to backup the data set. If cache-based concurrent copy is unavailable, the data set backup continues as if the CC keyword was not specified. When processing a UNIX file, UNIX File SnapShot processing will be used. If UNIX File SnapShot processing is unable to be used, the backup continues as if the CC keyword were not specified.

PHYSICALEND

Users receive data set backup results when the backup operation is complete (the default). For a WAIT-type request, the job receives control only after the physical end of the copy.

LOGICALEND

DFSMSHsm releases serialization for the data set or UNIX file and returns control to the data set backup WAIT requester after concurrent copy initialization successfully completes. If concurrent copy initialization fails and PREFERRED was specified, notification is delayed until the end of the backup operation. If concurrent copy initialization fails and REQUIRED was specified, the request fails. Because users are not notified at the physical end of the backup operation, use caution if you use this option.

Note: Use these additional considerations when using cache-based concurrent copy to process large data sets or volumes, or when using multiple tasks to process a large number of data sets or UNIX files:

- Use concurrent copy or UNIX file Snapshot only during periods of light update activity for the data sets UNIX files, or volumes involved.
- Spread multiple concurrent copy jobs across as many LPARs as possible.
- Ensure that jobs and address spaces that will use concurrent copy are assigned a WLM service class with a high execution velocity. Never assign a discretionary goal to concurrent copy work.
- Ensure that volumes that will be processed will not be reserved by another system during the concurrent copy.

Authorizing concurrent copy with RACF

By default, all users can use the CC options. However, if RACF indicates a lack of authority, DFSMSHsm fails the data set backup request if the specified concurrent copy request was specified with the REQUIRED,VIRTUALREQUIRED or CACHEREQUIRED subparameters. If the CC option was used without the REQUIRED,VIRTUALREQUIRED or CACHEREQUIRED subparameters and RACF indicates a lack of authority, DFSMSHsm continues to back up the data set as if the CC keyword were not specified on the backup command.

Data set backup using NEWNAME

The NEWNAME parameter of the (H)BACKDS command allows you to create a data set backup copy and store it under a different, specified data set name. This functionality allows you to keep a permanent copy of data (for example, a year-end copy of a critical data set) as long as a unique new name is specified for the permanent version. The command can be used for many other purposes, such as to create special backup copies to be used for disaster recovery, or to populate a test environment with backup versions of data sets copied to that system under a different name.

You can also use the (H)BACKDS dsname NEWNAME(newdsname) command to convert backup copies created with a competitive product to be DFSMSHsm-managed backup copies by following the steps below:

- Restore all backup versions from the competitive product with temporary names
- (H)BACKDS the oldest temporary version of the data set using the original name in the NEWNAME parameter (No allocation of the original data set required.)
- (H)BACKDS the next oldest temporary version of the data set with original name in the NEWNAME parameter. Repeat until the most recent version has been backed up
- Delete the temporary data set copies

To preserve the original creation date and time of the backup copy, specify the DATE and TIME parameters on the (H)BACKDS NEWNAME command. If DATE is not specified, the current date will be assigned to the backup copy. If DATE is specified, DFSMSHsm will use the date to chronologically position the backup copy in relation to other existing backup versions. If DATE is specified without TIME, the current time will be assigned to the backup version. If the TIME keyword is not specified, and a backup copy with the data specified on the command and the current time already exists, the command will fail.

The BACKDS NEWNAME command will fail when:

- The 'dsname' data set is migrated, uncataloged, or is a catalog
- The 'dsname' and 'newname' data sets do not have the same dataset organization
- The 'dsname' and 'newname' data sets do not have the same SMS-managed or non-SMS-managed status
- More than one 'dsname' or more than one 'newname' is specified
- The 'newname' data set is pre-allocated on tape

UNIX file backup using NEWNAME

The NEWNAME parameter of the (H)BACKDS command allows you to create a UNIX file backup copy and store it in the DFSMSHsm inventory under a directory name that is a subset of the existing directory path. This functionality can be used to populate a test environment with backup versions of UNIX files copied to that system under a temporary directory.

You can also use the (H)BACKDS '/pathname/' NEWNAME('/subpath/') command to convert backup copies created with a competitive product to be DFSMSHsm-managed backup copies by following these steps:

- Restore directory tree backup versions from the competitive product into a temporary directory.
- (H)BACKDS the oldest temporary version of the directory structure, specifying RECURSE. Specify the absolute directory path of the original in the NEWNAME parameter.
- (H)BACKDS the next oldest temporary version of the directory structure, specifying RECURSE. Specify the absolute directory path of the original in the NEWNAME parameter. Repeat until the most recent version has been backed up.
- Delete the temporary UNIX file copies.

To preserve the original creation date and time of the backup copy, specify the DATE and TIME parameters on the (H)BACKDS NEWNAME command. If DATE is not specified, the current date will be assigned to the backup copy. If DATE is specified, DFSMSHsm will use the date to chronologically position the backup copy in relation to other existing backup versions. If DATE is specified without TIME, the current time will be assigned to the backup version. If the TIME keyword is not specified, and a backup copy with the data specified on the command and the current time already exists, the command will fail.

Tip: It is best if the NEWNAME directory exists at the time of the backup. DFSMSHsm will store the name of the containing zFS data set in the backup record for future SMS-management checks. (This attribute is used for the number of versions to keep for roll-off processing, ALTERDS checking, and whether the Recover processing should end due to a No-Crossmounts condition on a recursive file request. If the NEWNAME directory doesn't exist, DFSMSHsm cannot store the containing zFS name, and some functions, such as ALTERDS may be unable to be performed.

Additional considerations for dump

Several additional topics are related to dump:

- Dumping DFSMSHsm-owned volumes
- Dumping control data sets
- Dumping the journal data set
- Preventing dump processing
- Invoking full-volume dump compression
- Bypassing enqueue processing between full-volume dump and DFSMSdfp functions

Dumping DFSMSHsm-owned volumes

DFSMSHsm-owned DASD backup and migration level 2 volumes are not automatically dumped. You must issue a DFSMSHsm BACKVOL command for each migration level 2 volume you want to dump. The activity

against the volumes should be at an absolute minimum while they are being dumped. The volumes do not have any internal DFSMSHsm serialization performed for them to prevent migration copies or backup versions from being created on them while they are being dumped, so it is imperative to dump these volumes when no migration or backups are being done at the same time. Also, no VTOC copy data sets are created for DFSMSHsm-owned volumes regardless of the VTOCCOPIES subparameter of the target dump classes.

Dumping control data sets

If you dump the volumes that contain the control data sets, *you will destroy your control data sets if you do a restore of the volume*. If you restore the control data sets, you must immediately recover the control data sets. If you dump the volume containing the BCDS, no concurrent dump activity should be processing on another host. A lockout can occur.

Related reading

For more information about how to recover control data sets, see [“Reasons for recovering the control data sets” on page 507](#).

Dumping the journal data set

If you dump the volume containing the journal data set, if your environment consists of multiple hosts with a global resource serialization product, and if the journal resource QNAME=ARCGPA and RNAME=ARCRJRN and the resource QNAME=SYSVTOC and RNAME=volser-containing-journal, both resources must be treated as either local resources or as global resources.

If the resource QNAME=SYSVTOC and if RNAME=volser-containing-journal is named in the systems exclusion resource name list, the resource QNAME=ARCGPA and RNAME=ARCRJRN must also be named in the systems exclusion resource name list. In this case, global resource serialization treats both as local resources. The reserve conversion resource name list is not searched, and the system is allowed to issue a hardware reserve for these resources.

If the journal data set shares a volume with a data set that has a higher access priority requirement than the journal data set and if the resource QNAME=ARCGPA and RNAME=ARCRJRN is named on the reserve conversion resource name list, then the resource QNAME=SYSVTOC and RNAME=volser-containing-journal must also be named on the reserve conversion resource name list. In this case, global resource serialization treats both as global resources to be processed around the ring and suppresses any hardware reserves for these resources.

Preventing dump processing

You can use the DFSMSHsm HOLD command to prevent all dump processing or to prevent only automatic dump processing.

The following command prevents both command and automatic dump processing:

```
HOLD DUMP
```

The following command prevents automatic dump processing without preventing command dump processing:

```
HOLD DUMP(AUTO)
```

The following command prevents fast replication target volumes from being dumped by command:

```
HOLD DUMP(FRBACKUP)
```

You can specify whether you want the holding action to occur immediately at the end of processing each data set currently being dumped or to occur at the end of each volume being dumped. If you hold dump

at end of volume, it can take a long time for all of the volumes that are being processed by the different dump tasks to be completed and the hold to take full effect. If you hold dump at the end of data sets, DFSMSHsm resumes command dump with the next queued entry when the hold is released. However, for automatic dump, the volume that was being processed when the HOLD at end of data set occurred remains eligible for processing if automatic dump is restarted.

The holding action begins when an MWE is selected for processing, not when the HOLD command is entered. If you use an HSEND CMD command to enter a wait-type dump request and the dump function is held, DFSMSHsm fails the wait-type dump request because the hold dump function gives no indication of when the dump function is available.

When a non-wait-type dump request is issued and the dump function is held, the request remains on the queue and is processed when the dump function is released. The held dump function is released with the RELEASE DUMP command.

Related reading

For more information about other ways to prevent DFSMSHsm from processing a dump function, see [“Preventing the processing of data sets during space management” on page 149](#).

Dumping all allocated space

Programming Interface Information

DFSMSHsm’s full-volume dump invocation of DFSMSdss specifies the ALLEXCP and ALLDATA(*) parameters. If the field in the DSCB that shows the last-used track and the block on that track contain zeros, the ALLEXCP option dumps all allocated space. When ALLDATA(*) is specified, all allocated space of SAM and PAM data sets, and data sets with a null DSORG field and whose last-used track and block on that track is greater than zero, are dumped. Use the following PATCH commands to suppress these parameters:

To suppress ALLEXCP:

```
PATCH.MCVT.+3C3 BITS(1.....)
```

To suppress ALLDATA(*):

```
PATCH.MCVT.+3C3 BITS(01.....)
```

To suppress both ALLEXCP and ALLDATA(*):

```
PATCH.MCVT.+3C3 BITS(11.....)
```

End Programming Interface Information

Invoking full-volume dump compression

You can compress all volume dumps by specifying the compress parameter at the DFSMSHsm full-volume dump invocation of DFSMSdss. Use the following PATCH command to specify the compress parameter:

Programming Interface Information

```
PATCH .MCVT.+3C3 BITS(001.....)
```

End Programming Interface Information

Note: This is ignored if HWCOMPRESS(YES) is active.

Bypassing enqueue processing between full-volume dump and DFP functions

To avoid contention problems between DFSMSHsm full-volume dump and DFP scratch, rename, and allocate services, major resource name ADRLOCK and minor resource name NONSPEC are allocated EXCLUSIVE when the volume is not known.

Since this process can have an impact on dump performance, you can bypass it by using the following patch:

Programming Interface Information
PATCH .MCVT.+38F X'10' VERIFY(.MCVT.+38F X'00')
End Programming Interface Information

Volume recovery and restore considerations

The following topics discuss items to consider when doing volume recovery and restore:

- Restoring data sets or volumes directly with DFSMSdss
- Requesting a list of dump volumes for a dump copy
- Determining the dump copy data set name
- Invoking DFSMSdss's RESTORE function
- Preventing recovery and restore processing
- Use of the date-last-referenced field for recovery

Restoring data sets or volumes directly with DFSMSdss

It is possible to invoke DFSMSdss directly to restore a data set or volume from a DFSMSHsm-owned dump copy. This requires you to perform the following steps:

1. Request from DFSMSHsm a list of the tape volumes that comprise the dump copy.
2. Determine the data set name that is used for the dump copy.
3. Invoke the DFSMSdss RESTORE function directly.

Requesting a list of dump volumes for a dump copy

Use the DFSMSHsm LIST command to request the set of dump volumes that compose the required dump copy. You must know the volume serial number of the volume you need to restore or the volume serial number of the volume on which the data set you want to restore resided when it was dumped.

Determining the dump copy data set name

DFSMSHsm generates a unique data set name for each dump copy that it has created through DFSMSdss. This data set name must be known in order to allocate the dump volumes. If you need to perform a restore using DFSMSdss directly without DFSMSHsm, you need the DFSMSHsm-generated data set name. This name can be obtained by issuing the following command:

```
LIST DUMPVOLUME(volser) DUMPCONTENTS
```

Related reading

For more information about the format of these data set names and how to determine the name of a specific dump copy, see [“Names of dump copy data sets” on page 197](#).

Invoking the DFSMSdss RESTORE function

You can invoke DFSMSdss in one of two ways: either write the necessary JCL to submit a batch job, or if you have ISMF, use the menu-driven panels of ISMF to build a job stream that will perform a DFSMSdss RESTORE.

If you received an invalid track format error (ADR367E) during a DFSMSdss full-volume restore, you can restore this volume by issuing the following DFSMSdss restore command in your JCL or ISMF job stream:

```
RESTORE FULL INDDNAME(tapename) OUTDDNAME(dasdname)
PURGE COPYVOLID
```

This restore restriction exists because DFSMSHsm uses the CANCELERROR keyword in its full-volume dump restore. Volumes that receive invalid track format errors (ADR367E) can only be restored outside of DFSMSHsm by using the DFSMSdss restore command, as previously explained.



Attention: DFSMSdss cannot directly restore dump tapes that are password protected by DFSMSHsm. Such tapes require special consideration during restore processing. Because DFSMSdss uses EXCP for reading, OPEN requires that the data be protected from destruction. Therefore dump tapes with password protection must be write-protected (ring removed or cartridge knob set) before they are mounted for restore processing.

Related reading

For more information about using DFSMSdss JCL code or the ISMF panels for DFSMSdss, see [z/OS DFSMSdss Storage Administration](#).

Preventing recovery and restore processing when using the HOLD RECOVER command

You can use the HOLD RECOVER command to prevent all recovery, to restore processing, or to prevent recovery alone from tape.

The following command prevents recovery processing from tape without preventing recovery processing from DASD:

```
HOLD RECOVER(TAPEDATASET)
```

Note: Holding tape recovery has the effect of holding all restores, because all dump copies managed by DFSMSHsm are on tape. Holding tape recovery does not hold the data set recovery from a mounted backup tape already in use.

The following command prevents both tape and DASD recovery processing:

```
HOLD RECOVER
```

The holding action begins when an MWE is selected for processing, not when the HOLD command is entered. If you enter a wait-type recovery or restore request and the recovery function is held, DFSMSHsm fails the wait-type recovery request because the hold recovery function gives no indication of when the recovery function will be available.

When a non-wait-type recovery or restore request is issued and the appropriate recovery or restore function is held, the request remains on the queue and is processed when the appropriate recovery function is released. The held recovery or restore function can be released with the RELEASE command.

Using the date-last-referenced field for recovery

When a data set is recovered or restored by an unauthorized user (HRECOVER), the date-last-referenced field in the data set VTOC entry of the data set is updated with the current date. When a data set is recovered or restored by an authorized user, the date-last-referenced field in the data set VTOC entry is

updated with the date-last-referenced field from the data set's VTOC entry that was saved at the time of backup or dump. When a data set is recovered or restored with the NEWNAME parameter of the RECOVER command, the date-last-referenced field in the data set VTOC entry is updated with the current date regardless of the authorization of the user.

Deferred generation data sets

Deferred generation data sets are generation data sets for which a failure has occurred before the data set is correctly associated with the generation data group. These data sets are temporary in the sense that at some time they may become associated with the GDS. DFSMSHsm treats deferred generation data sets as permanent non-GDS data sets.

Security considerations for availability management

DFSMSHsm does not check data set security during automatic backup and dump. DFSMSHsm also bypasses security checking when it processes operator commands entered at the system console or commands issued by a DFSMSHsm-authorized user.

DFSMSHsm checks security for data sets when a user who is not DFSMSHsm-authorized issues an HBACKDS, HBDELETE, HALTERDS, or HRECOVER command. Security checking is not done when DFSMSHsm-authorized users issue the user commands. If users are not authorized to manipulate data, DFSMSHsm does not permit them to back up data sets, delete backup versions, change the conditions for backup versions, or recover data sets. [Table 46 on page 460](#) shows the RACF authority required to perform each availability management function.

Table 46. Resource Access Authority Required for DFSMSHsm Functions

DFSMSHsm Function	RACF Resource Access Authority Required
Back up a data set	Update
Recover a backup version without specifying the NEWNAME parameter	Alter If profile recovery is required, authority is needed to create a discrete RACF profile for the recovered data set.
Recover a backup version and specify the NEWNAME parameter	Read authority to the data set being recovered. Alter authority to the newly named data set if one exists. If profile recovery is required, authority is needed to create a discrete RACF profile for the recovered data set.
Changing backup parameters	Alter
Deleting backup versions	Alter

Authorization checking during recovery

During recovery, DFSMSHsm makes a first authorization check against the original profile. If it cannot find either a generic profile or a discrete profile for a cataloged data set that was RACF-indicated at the time of backup or recovery, DFSMSHsm determines whether a backup profile exists. If a backup profile exists, DFSMSHsm makes a second authorization check with the backup profile name. If the second authorization check fails, the recovery fails.

When DFSMSHsm is the data mover, a new RACF access control environment element (ACEE) is created that describes the user's RACF status. This ACEE exists for the duration of the cataloging operation and is effective only for the recovery task. After the cataloging operation has completed, DFSMSHsm removes the ACEE.

For a non-VSAM data set, DFSMSdfp catalog management is invoked to catalog the data set. DFSMSdfp does its own security checking of the requester against the catalog using this ACEE. If security checking

against the catalog fails for lack of correct authority, DFSMSHsm sets a return code to reflect the failure of the catalog operation. RACF issues messages describing the reasons for denying access to the catalog.

For a VSAM data set, a new RACF ACEE is created when the target data set of the Access Method Services IMPORT has not been cataloged at the start of the recovery operation. This ACEE is effective for the duration of DFSMSHsm's invocation of Access Method Services to perform the IMPORT operation. DFSMSdftp catalog management will use this ACEE to verify the requester's authority to catalog the data set. If the catalog operation fails, the IMPORT fails and generates an appropriate return code. RACF issues messages describing the reasons for denying access to the catalog.

When DFSMSdss is the data mover, the authorization process is the same, except that DFSMSdss does the cataloging.

Recovering or restoring RACF-indicated data sets

When you use the RECOVER or HRECOVER command to recover or restore a RACF-indicated, cataloged, non-VSAM data set and the target volume is different from the source volume, DFSMSHsm changes the volume serial number and device type information in the RACF discrete data set profile to indicate the volume where DFSMSHsm has recovered the data set.

If the attempt to change a profile fails because the profile cannot be found, DFSMSHsm attempts to recover a backed up profile for the recovered data set.

When a RACF naming conversion table converts a data set name to a different profile name, DFSMSHsm will not be able to recover the discrete profile. The RACF administrator will need to enable access.

Recovering RACF discrete profiles

If a recovered data set was cataloged and RACF-indicated when it was backed up or when it was recovered or restored, DFSMSHsm tries to recover the RACF discrete profile for the data set even if you specified the NOPROFILEBACKUP parameter of the SETSYS command or if the profile already exists.

DFSMSHsm creates a RACF discrete profile during recovery by performing the following actions:

1. Modeling the new profile after the original profile, and if that isn't possible,
2. Modeling the new profile after the backup profile and if that isn't possible,
3. Creating a minimal discrete profile.

DFSMSHsm tries to recover the RACF discrete profile by using the profile of the original data set as a model for creating a profile. If the original data set's profile no longer exists, DFSMSHsm tries to create a profile by using a backup profile for the model.

If a backup profile does not exist, DFSMSHsm tries to create a minimal discrete profile. A minimal discrete profile is one for which DFSMSHsm supplies no access list and no model profile. The access authority with which users can address the data set is taken from the defaults defined for the user (or group if the high-level qualifier is a group name). If a DFSMSHsm-authorized user recovers the data sets, defaults are taken from the user ID associated with the DFSMSHsm-started procedure.

DFSMSHsm creates a minimal discrete profile only if neither the original profile nor a backup profile exists for the recovered data set. If the profile creation fails because a profile already exists for the recovered data set, DFSMSHsm considers the profile recovery to be successful, and normal processing continues. When this occurs, RACF may issue message ICH408I at the security console to indicate that the resource is already defined.

Note: DFSMSHsm recovers a discrete profile if one existed at the time of backup or recovery. If you change from RACF discrete profile protection to generic profile protection, DFSMSHsm still recovers the discrete profiles when it recovers the backup versions of data sets that were once protected with discrete profiles. So, discrete profiles can reappear even if you now use generic profile protection.

Related reading

For more information about the installation of generic profile options, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Authorization checking during profile recovery

When a DFSMSHsm-authorized user issues a RECOVER or HRECOVER command, no security checking is performed. If the recover request originates with a non-DFSMSHsm-authorized user, the recover is done with the user's ACEE, which is a control block for RACF. DFSMSHsm informs RACF that security checking is to be done, and the recover request fails if RACF determines that the user is not authorized to create a discrete profile for the data set.

Characteristics of the recovered profile

When a DFSMSHsm-authorized user issues a RECOVER or HRECOVER command, DFSMSHsm recovers profiles using DFSMSHsm's ACEE. Consequently, the recovered profile shows the group name from the user ID associated with the DFSMSHsm-started procedure. If a minimal profile is created when a DFSMSHsm-authorized user issues a recover request, additional information from this user ID is placed in the recovered profile. This information includes levels of auditing, access authority required by other users (universal access), and profile ownership.

If a non-DFSMSHsm-authorized user issues the recover request, the user's ACEE is used to create a new profile. The recovered profile reflects the group name of the requester. If a minimal profile is created, the profile also reflects the requester's user ID in the owner field and the requester's defaults for audit level and universal access authority.

Note: For steps “2” on page 461 and “3” on page 461, if the DFSMSHsm-started procedure is not associated with a RACF user ID, the user and group names of the recovered profile appear as an asterisk (*) if the data set is recovered by a DFSMSHsm-authorized user.

Effects of profile modeling on profile recovery

Profile modeling is a RACF function that merges information from a *model profile* into a new profile when it is created. Profile modeling is available at both the user ID and group level. If the user ID associated with the DFSMSHsm-started procedure has a model profile, information from the model is put into all backup profiles. Model profiles can provide information for universal access, auditing level, profile owner, level field, installation-defined data, and an access list. If a profile is subsequently recovered, the additional information from the profile model can be carried along and inserted into the recovered profile.

When a non-DFSMSHsm-authorized user issues a recover request that results in profile recovery, the user's ACEE is used to create the profile. Consequently, if profile modeling is active for the user or the user's group, information from the appropriate profile is merged into the newly created, recovered profile.

Related reading

For more information about profile modeling, see [z/OS Security Server RACF Security Administrator's Guide](#).

Accessing a data set when the profile cannot be recovered

Even if DFSMSHsm cannot recover a discrete profile, you can access the recovered data set. [Table 47 on page 463](#) shows how to access data sets using the DFSMSHsm RECOVER command when recovering data sets that are RACF-protected with discrete profiles and DFSMSHsm is unable to recover a discrete profile. These data sets were generally backed up by DFHSM before Version 2 Release 2.0.

Table 47. Recovering RACF-Protected Data Sets Backed Up under DFHSM before Version 2 Release 2.0

Type of Data Set	Parameter	Result	How to Access the Data Set
Cataloged Non-VSAM	RECOVER NEWNAME	If the RACF indicator was on when the backup version was created for the current data set, the RACF indicator is on in the recovered data set.	If a RACF profile does not already exist, create a generic profile or use the RACF ADDSD command with the NOSET parameter to create a RACF discrete profile for the data set with the new name.
Cataloged VSAM	RECOVER NEWNAME	DFSMSHsm scratches the newly named data set. The recovered data set replaces the newly named data set, including the catalog record.	If the original data set was protected with a discrete profile and the newly named data set was not, use the ADDSD NOSET command to create a new discrete profile. Or, you can create a generic profile for the newly named data set.
Uncataloged	RECOVER TOVOLUME (Volume different from the one the data set was backed up from)	If the RACF indicator was on when the backup version was created or the newly named data set is RACF indicated, the RACF indicator is on in the recovered data set.	If a RACF profile does not already exist, use the RACF ADDSD command with the NOSET parameter to create a RACF profile for the data set.
Cataloged Non-VSAM or VSAM	RECOVER (NEWNAME is not specified)	If the RACF indicator was on when the backup version was created for the current data set, the RACF indicator is on in the recovered data set.	If a RACF profile does not already exist, create a generic profile or use the RACF ADDSD command with the NOSET parameter to create a RACF discrete profile for the data set.

Maintaining password protection for recovered data sets

Password protection in MVS is implemented differently for non-VSAM and VSAM data sets. Password protection is not applicable to SMS-managed data sets. The passwords for all non-VSAM data sets are stored in one system password data set. The password for each VSAM data set is stored in a catalog record for the specific VSAM data set. Therefore, VSAM passwords are maintained during DFSMSHsm processing as a part of regular catalog record creation and updating.

If you are recovering an existing VSAM data set, the password is the current password. If you are recovering a VSAM data set that has been deleted, the password is the password that existed when DFSMSHsm backed up the data set.

If you specify the NEWNAME parameter of a RECOVER command for a password-protected, non-VSAM data set, DFSMSHsm marks in the DSCB of the recovered data set the same password protection as the original data set had. Enter a password for the recovered data set in the system password data set. Then, use the password to access the recovered, non-VSAM data set.

For VSAM data sets, whether you specify the NEWNAME parameter, DFSMSHsm marks in the catalog that the recovered data set is protected and enters the old password in the catalog record of the recovered data set. Use the old password when you want to access the recovered VSAM data set.

If DFSMSHsm migrated a password-protected VSAM data set whose backup versions are to be deleted, you must specify the password or DFSMSHsm fails the request. When deleting backup versions of VSAM data sets that no longer exist or that were migrated by DFSMSHsm Version 1 Release 3, DFSMSHsm does not check the password.

Be aware of the following password issues when you attempt to change password protection while data sets are backed up:

- You can change the password of a backed up non-VSAM data set by using the IEHPROGM utility. However, to change the password of a backed up VSAM data set you must use the Access Method Services facility. For a backed up VSAM data set, the password required to recover a backup version is the one that existed when the backup version was created if the data set is not currently cataloged.
- If you try to add password protection to a non-existing non-VSAM data set that has backup versions, the password is added to the system password data set. However, the backup versions do not

have protection, and when DFSMSHsm recovers the data set, the DSCB does not indicate password protection. If an existing non-VSAM data set is password protected at the time it is recovered or restored, even if it was not protected when the data set was backed up or dumped, the data set is password indicated after the recovery or restore.

- If you try to add password protection to a backed up VSAM data set, you must use the Access Method Services facility. If you recover the backup versions, DFSMSHsm does not protect the recovered data set if the data set was not protected when DFSMSHsm backed it up.

Note: If a catalog has update protection, you are prompted for the password when DFSMSHsm tries—during an unauthorized DFSMSHsm function from an unauthorized DFSMSHsm user—to create a catalog entry in the protected catalog.

DFSMSHsm retrieval of management classes

If you are changing the definitions of management classes, you must know when the new definitions will take effect. When DFSMSHsm begins automatic availability management or availability management from a volume command, it retrieves a list of all currently defined management classes before it begins processing unless volume-type processing is operating or has just completed. Volume-type processing consists of the following functions:

- Automatic backup
- BACKVOL command

If volume-type processing is in progress or has just been completed when a new volume-type processing is to begin, DFSMSHsm uses the list of management classes that exists from the prior processing. If you are changing the management classes, DFSMSHsm must complete its volume-type processing before you can expect the changes in the management classes to become effective.

Availability management considerations for extended SAM and extended and compressed VSAM data sets

When DFSMSHsm does backup and recovery processing for extended format SAM and VSAM compressed data sets, be aware of the following results:

- When DFSMSHsm backs up an extended format VSAM compressed data set, it recovers that data set only as an extended format compressed data set.
- When DFSMSHsm backs up an extended format SAM compressed data set, it recovers that data set as either an extended format compressed or nonextended format SAM data set.
- When DFSMSHsm backs up a compressed SAM or VSAM data set, it suspends software compression for that data set.
- When a compressed SAM or VSAM data set is backed up, DFSMSHsm records in the BCDS that it is a compressed data set. It also records both the compressed size of the data set and the size the data set would be if it were not compressed. This information is passed to DCOLLECT.

Availability management considerations for sequential data striping

DFSMSHsm does backup and recovery processing for single-stripe and multiple-stripe sequential data sets. To process these data sets, the data mover must be DFSMSdss, and the data sets must be SMS managed.

Backup processing

Incremental backup processing backs up a single-stripe or multiple-stripe data set when the data set's name is specified in the BACKDS command, and other required conditions are satisfied.

During volume incremental backup processing, DFSMSHsm backs up a sequential extended format data set when the first stripe (volume) is being processed. Backup is skipped when subsequent stripes (volumes) are being processed.

If a sequential extended format data set is larger than the available free space on one DASD backup volume, then the backup fails. Note that it is not advisable to use extended address volumes (EAVs) as backup volumes because the cylinder managed space will not be utilized.

The space used on all volumes is reported (message ARC0734I) for the sequential extended format data set backed up.

During full physical volume dump processing, DFSMSHsm dumps only the stripe on the volume being dumped. You are responsible for performing full physical volume dumps for each volume containing a stripe of a sequential extended format data set. It is recommended that you use incremental backup versions for multiple-stripe sequential extended format data sets.

Incremental backup processing does not occur under the following conditions:

- A sequential extended format data set is not backed up if it is on a non-SMS-managed volume.
- A multiple-stripe sequential extended format data set is not backed up if a SETSYS USERDATASETSERIALIZATION is not presented to DFSMSHsm.
- A multiple-stripe sequential extended format data set is not backed up if it is protected by a RACF discrete profile.

Recover and restore processing

DFSMSHsm recovers or restores a sequential extended format data set (indicated in the MCC record) using DFSMSdss as the data mover.

During full volume restore with APPLYINCREMENTAL, DFSMSHsm does not apply the incremental backup version that is more recent than the full physical dump copy to a currently cataloged multiple-stripe sequential extended format data set. The stripe restored from the dump copy is scratched from the volume if the data set is no longer cataloged on the volume being processed.

During data set recovery of a multiple-stripe sequential extended format data set, without FROMDUMP(DUMPVOLUME(*volser*)) being specified, DFSMSHsm always recovers an incremental backup version made by DFSMSdss, even if a physical full-volume dump copy is more recent than the incremental backup version. A message is issued to indicate that an incremental backup version has been recovered instead of a more recent dump copy.

During a recovery attempt of a multiple-stripe sequential extended format data set, when FROMDUMP(DUMPVOLUME(*volser*)) has not been specified, recovery fails if the data set has only been backed up by physical full-volume dump.

DFSMSdss restores a sequential extended format data set as follows:

- A sequential extended format data set that was multiple-stripe when it was backed up or dumped can be recovered or restored as a multiple-stripe or single-stripe sequential extended format data set, regardless of the existence of the target data set before the RECOVER command is issued.

The number of stripes depends on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.

- A sequential extended format data set that was single-stripe when it was backed up or dumped can be recovered or restored as a multiple-stripe or single-stripe sequential extended format data set, regardless of the existence of the target data set before the RECOVER command is issued.

Again, the number of stripes depends on the storage class SUSTAINED DATA RATE value and the availability of the required volumes.

A sequential extended format data set that was either single-stripe or multiple-stripe when it was backed up or dumped is recovered or restored as a sequential extended format data set or non sequential extended format data set, regardless of the existence of the target data set before the RECOVER command is issued.

A data set that was not sequential extended format when it was backed up is always recovered or restored as a data set that is not sequential extended format. DFSMSHsm does not convert a data set that is not sequential extended format to a sequential extended format data set through recovery or restore processing, even if the data set is preallocated as sequential extended format before the RECOVER command is issued.

If ACS filter routines are modified to direct a data set that was sequential extended format when it was backed up to a non-SMS-managed volume, the recovery or restore fails.

If FORCENONSMS is specified on the RECOVER command to force a sequential extended format data set to be recovered or restored to a non-SMS-managed volume, the data set will be recovered as a non sequential extended format data set. The maximum size of the non-SMS SAM data set is 1 volume.

If FROMVOLUME is specified on the RECOVER command to recover a sequential extended format data set, the recovery or restore fails.

If reblocking is required for a sequential extended format data set, DFSMSHsm passes the requirement to DFSMSdss, which performs the reblocking.

To restore a single-stripe sequential extended format data set from a physical full volume dump tape, you can issue the RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) command for the stripe dumped to the specified dump tape.

To restore an existing multiple-stripe sequential extended format data set from physical full volume dump tapes, you must issue the RECOVER *dsname* FROMDUMP(DUMPVOLUME(*volser*)) command for each stripe of the data set. If the sequential extended format data set no longer exists, you must catalog the data set after the restores, since DFSMSdss and DFSMSHsm do not catalog a multiple-stripe sequential extended format data set during physical data set restore processing.

DFSMSHsm fails the recovery of a sequential extended format data set if the desired backup version indicates that the data set was sequential extended format when it was backed up, and the data mover was DFSMSHsm.

Note: DFSMSHsm recovery or restore processing does not honor the “stripedness” specified in the data class DSNTYPE in the ACS filter routine. Changing the DSNTYPE definition or allocating the target data set before issuing the recover command has no effect on converting to or from striped data during DFSMSHsm recovery or restore processing.

DFSMSHsm does not recover a multiple-stripe sequential extended format data set during incremental full volume recovery.

If the target data set is preallocated and the REPLACE keyword is specified, DFSMSHsm data set recovery, restore, incremental volume recovery, and full volume restore with APPLYINCREMENTAL processing delete the non-VSAM target data set, both SMS-managed and non-SMS-managed, before recovering or restoring the backup or dump copy.

DFSMSHsm command data set recovery always lets DFSMSdfp select the target SMS-managed volume for a non-VSAM data set, even if the target data set is cataloged before the RECOVER command is issued. The volume the data set is cataloged on is still passed to DFSMSdfp, but the final target volume is selected by DFSMSdfp.

Common dump queue

The common dump queue (CDQ) is a queue of DFSMSHsm dump requests from BACKVOL VOLUMES commands, BACKVOL STORAGE GROUP DUMP commands, FRBACKUP COPYPOOL DUMP commands, and auto dump requests. The CDQ is shared by multiple DFSMSHsm hosts. It is implemented through the use of the cross-system coupling facility (XCF), to balance the dump workload across the resources available in each DFHSMhsm host in an HSMplex. A DFSMSHsm host is defined to XCF as a member of a group to be used for dump processing. Distributing dump work to all of the members in the group increases the number of available tasks to perform the work, improving the throughput of the dump function, as the dump workload is not concentrated on a single host's address space.

To maximize the use of available tasks in the CDQ group, you can use a MINSTACK (minimum stack) option, in conjunction with the STACK option, on the DEFINE DUMPCLASS and BACKVOL commands. MINSTACK indicates that fully using the available dump tasks in the CDQ group takes priority over optimizing stacking; in other words, maximizing the use of available dump tasks may result in less stacking of volumes on the tape. However, more stacking is possible as volumes from different copy pools are allowed to be stacked onto the same tape, up to the maximum value specified by STACK, if they have the same dump class. This applies to the dump function running in a non-CDQ environment, as well. For example:

- With STACK(100), if there are 100 volumes and 10 available dump tasks, all 100 volumes are dumped using a single task to the same tape.
- If the MINSTACK(10) option is defined, 10 volumes are stacked on a single tape from each of the 10 dump tasks.
- With more volumes, stacking continues on a single tape up to the maximum of 100 volumes defined with STACK(100).

Figure 77 on page 467 shows a CDQ that is shared by three DFSMSHsm hosts that are connected to an XCF defined group, where submitted dump requests are distributed and processed among the dump resources available in the CDQ group, with results returned back to the host where the request originated.

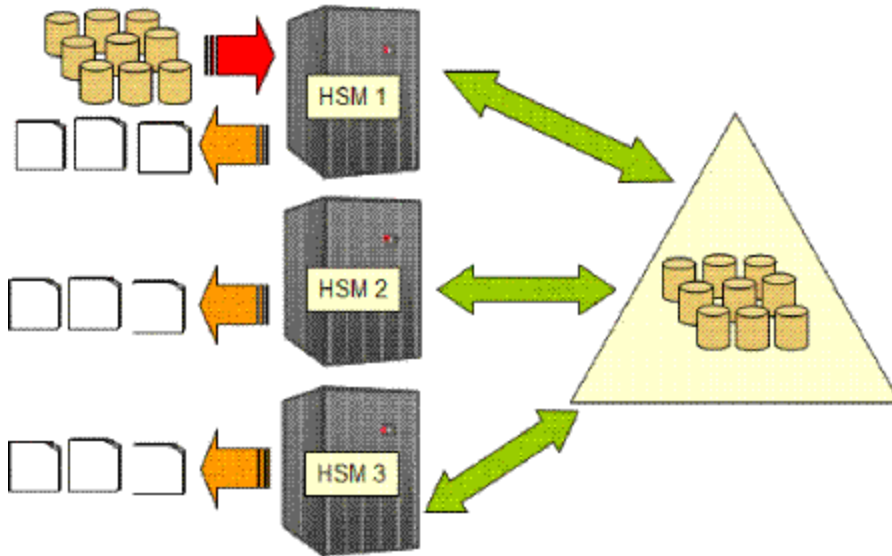


Figure 77. Common Dump Queue

As illustrated in Figure 78 on page 468, the CDQ group allows for flexible configurations. This provides the capability to:

- Define multiple queues in the same HSMplex
- Allow group members to both receive and process requests, only process requests, or only receive requests.

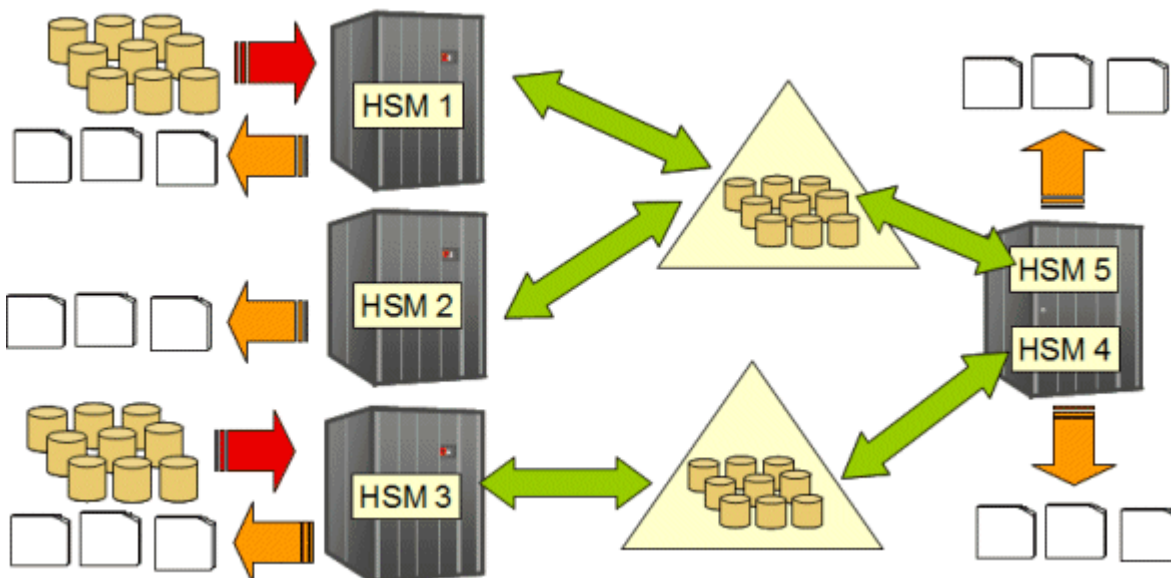


Figure 78. CDQ -- Flexible Configurations

In CDQ environments:

- A CDQ cannot span an HSMplex
- System affinity is not supported
- Priority of requests is supported
- A host can be connected to only one CDQ.

Results are returned to the submitting host, but progress and status messages are recorded on the host processing the request. By using the message simplification support along with CDQ processing, each CDQ group member writes all DFSMSHsm and DFSMSdss messages related to the processing of a particular copy pool to a single shared message data set. This simplifies diagnostics, as all messages are collected from each host into a single data set. The data set is managed by the master scheduler (MS) host for the CDQ. Refer to the description of the FASTREPLICATION(MESSAGE DATASET(YES)) option in Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.

Types of hosts used in a common dump queue

The master scheduler (MS) is the DFSMSHsm host that manages all of the dump requests in the CDQ. All dump requests are sent to it, and it assigns work to all of the other hosts in the group that are eligible to process requests from the CDQ. Each group needs a master scheduler to manage CDQ processing for the group; otherwise, dump requests are processed locally.

The other principal types of DFSMSHsm hosts are the submitting host and the processing host.

Submitting host

Host that receives requests from commands and, using XCF messaging, sends it to the master scheduler host. Performs command complete processing and posts the user that the command has completed.

Master scheduler (MS)

- Accepts requests from submitting host members and locally submitted requests and maintains them as a single common queue of all requests for the CDQ function. The master scheduler can be the submitting host – in this case the command is not sent.
- Assigns or distributes the requests among the available tasks across all of the eligible hosts (processing hosts) connected to the CDQ while balancing the requests among all the hosts.

- Manages "work completed" messages received from a processing host to support functions like stacking.
- Sends "command complete" messages back to the submitting host.

Processing host

- Receives assigned work requests from the master scheduler and places them on its local queue for processing.
- Sends "work complete" messages back to the master scheduler with specific protocols to support functions like stacking.
- The processing host could be the submitting host and/or the master scheduler host.

Figure 79 on page 469 shows a CDQ group where:

- HSM1 is the submitting and processing host
- HSM2 is the master scheduler and processing host. It receives all requests, managing them on a single queue, and distributes the requests among the eligible hosts connected to the common queue, through XCF messaging.
- HSM3 is a processing host.

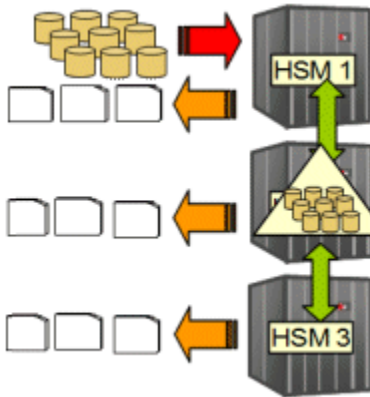


Figure 79. CDQ Host Types

Connecting to a common dump queue

The SETSYS COMMONQUEUE(DUMP(CONNECT(*basename*))) command defines the CDQ group name to XCF, and connects a DFSMSHsm host to it. *basename* is a four-character suffix to be added to the DFSMSHsm-designated group name prefix, ARCD. For example, a SETSYS CQUEUE(DUMP(CONNECT(QUE1))) command defines a DFSMSHsm host as a member of the XCF group named ARCDQUE1.

The first DFSMSHsm host from which the SETSYS command with the CONNECT option is issued defines the XCF group and connects itself to this group. After that, DFSMSHsm hosts connect to the XCF group defined by the first host.

Master scheduler

The first member to join the CDQ group that is a master scheduler candidate becomes the master scheduler host for the CDQ group.

The master scheduler candidate (MSC) setting can be defined:

- In conjunction with the CONNECT option of the SETSYS command:

```
SETSYS COMMONQUEUE(DUMP(CONNECT(basename) MSC(YES|NO))
```

MSC defaults to YES.

- Separately, to set the MSC state before the CONNECT option or to change it afterward:


```
SETSYS COMMONQUEUE(DUMP(MSC(Y)))
```

For example, an MSC(N) option on the SETSYS command issued for each host prior to the CONNECT option being used ensures that no host becomes a master scheduler candidate. In this case the CDQ group status is DISABLED and no CDQ processing occurs. All work is processed locally. After all SETSYS commands with the CONNECT option have been issued, the MSC state for one or more hosts could be changed with the MSC(Y) option, to allow that host to take on the master scheduler responsibilities. A DFSMSHsm host with a MSC(Y) state is eligible to take over when the current master scheduler host fails or disconnects from the group.

It is recommended that all of your DFSMSHsm hosts be defined to be master scheduler candidates. This ensures that one host will take on the master scheduler responsibilities when a CDQ group is defined, and ensures that another host will take on the master scheduler responsibilities once the current one is removed from the group due to a disconnection or system errors. However, if a system might not have enough capacity to perform this work, you may want to define it to not be a master scheduler candidate.

Query commands

To display the current SETSYS setting, use the QUERY SETSYS command and look for message ARC1500I. To display the master scheduler status, use the QUERY ACTIVE command and look for message ARC1560I.

The DISPLAY XCF,GROUP command shows the group name (ARCDQUE1 in the example) and the member name as HOST#*hsmid*, where *hsmid* is the HSM ID of this host.

For more information

For more information, refer to Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.

Placing requests on the common dump queue

A host that is participating in CDQ processing places its dump requests onto the CDQ for processing. The following considerations apply:

- To verify that a host is connected to the CDQ, issue the QUERY ACTIVE command, then look for CONNECTION STATUS of CONNECTED in message ARC1560I.
- HOLD DUMP (any type) issued on the host that issued the dump request places the request on the CDQ if the master scheduler is also not held, (WAIT type requests are failed if the master scheduler is held). Both WAIT and NOWAIT type requests are placed as the master scheduler is available to assign and process requests in the CDQ. Tasks for this host will subsequently not be assigned by the master scheduler for processing.
- HOLD DUMP (any type) issued on the master scheduler host prevents it from assigning and processing requests. Note that NOWAIT request types are placed on the CDQ even though dump was held on the master scheduler. To keep the master scheduler's responsibilities intact, that is, if you do not want tasks on the master scheduler host to be used by the CDQ, issue a SETSYS MAXDUMPTASKS(0) command instead of HOLD DUMP. Alternatively, you can transition the master scheduler responsibilities to another host with SETSYS CQUEUE(D(MSC(N))) prior to issuing the HOLD.
- HOLD DUMP (any type) issued on all the hosts in the CDQ places the master scheduler in the DISABLED state. Use the messages issued in response to the QUERY ACTIVE to verify this state. Note that the MASTER SCHEDULER STATUS is DISABLED(HELD). When DISABLED:
 - New WAIT request types are failed locally
 - New NOWAIT request types are queued locally
 - Queued WAIT request types in the CDQ master scheduler host are failed
- SETSYS MAXDUMPTASKS(0) issued on the host that issued the dump request places the request on the CDQ, but tasks will subsequently not be assigned by the master scheduler for processing.

- Any request that is not submitted to the master scheduler host (for example, due to errors) is placed on the local queue of the host that issued the request.
- Requests submitted on the master scheduler host are placed on its local queue.
- Requests that are sent to the master scheduler host from other members are placed on the local master scheduler queue and then selected to run locally on the master scheduler, or assigned to be processed on another member in the CDQ group.
- The default priority of WAIT dump requests over NOWAIT dump requests does not change with the CDQ. However, the ALTERPRI command does support change requests currently queued to the CDQ.

Selecting requests from the common dump queue

The master scheduler is the DFSMSHsm host that manages all of the dump requests in the CDQ. All dump requests are sent to it, and it assigns work to all of the other hosts eligible to process requests from the CDQ. The following considerations apply to selecting requests to be processed by the CDQ:

- The master scheduler selects hosts to process requests based on the current utilization of available tape resources in each host relative to each other. The first host with the lowest utilization is selected to process the request.
- A host's resources are not available if the HOLD DUMP or EMERGENCY command has been issued, or if the SETSYS MAXDUMPTASKS(0) has been issued. Each command indicates to the master scheduler with XCF messaging that this host has no resources available.
- Selected hosts that are not the master scheduler are assigned the work with XCF messaging. The assigned processing host selects the task to use.
- When the master scheduler selects its own dump task to process a CDQ request, the task is attached and assigned the work.

Processing requests off of the common dump queue

The master scheduler selects a host member of the CDQ to process a request placed on the CDQ. This host could be the master scheduler itself, another member in the CDQ, or even the member that placed the request on the CDQ that the request originated from. This selected member is the processing host and is responsible for selecting an available task to use for the copy-to-tape processing that it will perform. The following considerations apply to a processing host in the CDQ:

- The processing host selects an available task to process the request.
- Status and progress messages, along with diagnostic trace data that is generated, are recorded on the processing host. A single dump request may generate many different volume dumps, with each one processed by individual CDQ group members. Messages and diagnostic data are placed on the host processing the request, making it difficult to put together the message flow of an individual dump command from all the processing hosts.
- By using the message simplification support along with CDQ processing, each CDQ group member writes all DFSMSHsm and DFSMSdss messages related to the processing of a particular copy pool to a single shared message data set that is managed by the master scheduler host. This simplifies the understanding of the message flow for the original fast replication copy pool command, as all messages are collected from each host into this single data set. XCF messages are used to perform this function. Refer to the SETSYS FASTREPLICATION(MESSAGE DATASET(YES)) option in [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#).
- As volumes complete processing, the selected processing host working in conjunction with the master scheduler host manages the process of stacking the minimum (specified with MINSTACK) up to the maximum (specified with STACK) number of volumes to be placed on a single tape. This is accomplished with XCF messaging.
- The processing host could be the submitting host and/or the master scheduler host, or a host that is neither the submitting host nor the master scheduler host.

Disconnecting from the common dump queue

The SETSYS COMMONQUEUE(DUMP(DISCONNECT)) command requests that a member of the CDQ group that this host is currently connected to is to leave the group. The following considerations apply to this disconnect process:

- All CDQ requests being processed on this host, and requests that originated from this host and are selected to be processed in the CDQ, are waited upon for completion before the host is disconnected from the CDQ. This host will no longer be selected to process new dump work from the CDQ.
- A HOLD DUMP issued prior to the disconnect command prevents queued requests from being selected to be processed. After the disconnect has completed, a RELEASE DUMP allows the queued requests to be processed locally.
- When the disconnect is for the host that is currently the master scheduler, all CDQ active and queued work to is waited upon for completion before the host is disconnected. Once the host is disconnected, another member in the CDQ group that has been previously defined as a master scheduler candidate automatically takes over the master scheduler responsibilities. A HOLD DUMP prior to a disconnection prevents queued requests from being selected for processing. A subsequent RELEASE DUMP after the disconnecting allows these requests to be processed locally.
- To verify the disconnection, use the QUERY ACTIVE command and look for message ARC1560I.
- Dump resources available on this host after the disconnection has completed are not known or available to the remaining CDQ members. All dump requests received are placed on this host's local queue and processed locally.
- For shutdown procedures, a DISCONNECT should first be issued on each host being removed from the HSMplex to allow CDQ requests to complete first.
- The DISCONNECT FORCE option should be reserved for emergency situations. With FORCE specified, requests that originated from this host and are selected to be processed in the CDQ are not waited upon for completion before disconnecting. Messages are issued to indicate that a request has been failed but still may complete in the CDQ.

For more information, refer to [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.](#)

Using QUERY, CANCEL and ALTERPRI in the common dump queue

The QUERY, CANCEL and ALTERPRI commands support the CDQ. The following considerations apply to QUERY processing:

- QUERY SETSYS displays the current SETSYS settings for the CDQ on the host that issued the query command. Message ARC1500I displays the SETSYS settings on this host.
- QUERY ACTIVE displays the current state of the CDQ and active requests running it.
 - The state and active requests that are reported are those that originated from the host issuing the query command
 - Active requests reported could be active in any member in the CDQ. XCF messaging is used to retrieve active requests by sending the query command to the master scheduler. These requests could be running in any member of the CDQ group and must match the filter criteria of the query. In this case, the active requests must have originated from the host issuing the query command. The generated messages on the master scheduler are returned to the query host using XCF and in turn the query host reports them. The query host then goes on to report its own active requests.
 - When the query host is also the master scheduler, no XCF messaging is used to obtain the active requests, as the CDQ queues are located on the master scheduler host.
 - For auto functions running in the CDQ, only requests running on the host that issued the QUERY ACTIVE command are displayed.
- QUERY REQUEST or USER display requests that originated from the host issuing the query command that match the REQUEST or USER filter criteria.

- Active requests reported could be active in any member in the CDQ and queued in the master scheduler. XCF messaging is used to retrieve these requests by sending the query command to the master scheduler. These active requests could be running in any member of the CDQ group or queued in the master scheduler and must match the filter criteria of the query. In this case, the requests must have originated from the host issuing the query command and match the REQUEST or USER criteria on the query command. The generated messages on the master scheduler are returned to the query host using XCF messaging and in turn the query host reports them. The query host then goes on to report its own requests.
- When the query host is also the master scheduler, no XCF messaging is used to obtain the queued and active requests, as the CDQ queues are located on the master scheduler host.
- QUERY WAITING displays the number of requests that are queued in the CDQ that originated from the host issuing the query command.
 - Queued requests in the CDQ are queued in the master scheduler. XCF messaging is used to retrieve these queued request numbers by sending the query command to the master scheduler. These queued requests in the master scheduler must match the filter criteria of the query. In this case, the requests must have originated from the host issuing the query command. The generated messages on the master scheduler are returned to the query host using XCF messaging, and the query host reports them. The query host then determines its own queued request numbers, combines them with what was returned from the CDQ, and reports the totals.
 - When the query host is also the master scheduler, no XCF messaging is used to obtain the queued request numbers, as the CDQ queues are located on the master scheduler host.
- QUERY COMMONQUEUE(DUMP) displays all queued and active requests being managed by the CDQ. Use this query to get a snapshot of all CDQ activity from any host connected to the CDQ.
 - Queued and active requests in the CDQ are managed by the master scheduler for the CDQ group. XCF messaging is used to retrieve these requests by sending the query command to the master scheduler. These queued and active requests in the master scheduler have no matching criteria to satisfy. All requests are reported. The generated messages on the master scheduler are returned to the query host using XCF messaging, and the query host reports them.
 - When the query host is also the master scheduler, no XCF messaging is used to obtain the queued and active requests as the CDQ queues are located on the master scheduler host. The filter criteria is all queued requests running in the CDQ regardless of where the dump request command originated.
 - Messages ARC1562I and ARC1563I are reported for the active and queued requests in the CDQ.
- The CANCEL REQUEST or USERID commands cancel queued requests in the CDQ for requests originating in the host that issued the CANCEL command and matching the REQUEST or USERID filter criteria.
 - Queued requests in the CDQ are managed by the master scheduler for the CDQ group. XCF messaging is used to cancel these requests by sending the cancel command to the master scheduler. These queued requests in the master scheduler must have originated from the host issuing the CANCEL command and match the filter criteria of the REQUEST and USERID parameters. When matching, the requests are canceled. The generated messages on the master scheduler are returned to the cancel host using XCF messaging and the cancel host reports them.
 - When the cancel host is also the master scheduler, no XCF messaging is used to cancel the queued requests, as the CDQ queues are located on the master scheduler host.
 - To cancel an active request in the CDQ, issue CANCEL TCBADDRESS on the host that this request is active in.
- The ALTERPRI REQUEST or USERID commands change the priority of queued requests in the CDQ for requests originating in the host that issued the ALTERPRI command and matching the REQUEST or USERID filter criteria.
 - Queued requests in the CDQ are managed by the master scheduler for the CDQ group. XCF messaging is used to alter the priorities of these requests by sending the ALTERPRI command to the master scheduler. These queued requests in the master scheduler must have originated from the host issuing the ALTERPRI command and match the filter criteria of the REQUEST and USERID

parameters. When matching, the requests priority is altered. The generated messages on the master scheduler are returned to the ALTERPRI host using XCF messaging, and the ALTERPRI host reports them.

- When the ALTERPRI host is also the master scheduler, no XCF messaging is used to alter the priority of the queued requests, as the CDQ queues are located on the master scheduler host.

Processing of failures in the common dump queue

The CDQ group support uses XCF facilities to detect when members lose their connection to the CDQ. This could be due to one or more of the following reasons:

- A host requested a disconnection with the SETSYS CQUEUE(DUMP(DISCONNECT))
- An XCF-related failure caused the member to be removed from the group
- A z/OS image failed

In all of these cases the host is considered to be no longer defined to the group.

When the host that is now not defined was the master scheduler, all remaining hosts in the CDQ group are notified that the master scheduler is now not defined. From the hosts in the CDQ group that are master scheduler candidates, a host takes the master scheduler responsibilities. For each of these hosts, any work that was placed on the CDQ prior to this event is placed on the CDQ, managed by the new master scheduler host. This may result in some volumes being processed again.

When the host that is now not defined is not the master scheduler, the master scheduler host is notified that this host is no longer defined to the group. The CDQ in this case searches for requests previously selected to run on this now not defined host, and fails them with an ARC1327I or ARC1829I message, with a return code of 105. This indicates that a host in the common queue group became not defined, with the return code indicating that the work was selected by the master scheduler to run in this host of the CDQ, but the master scheduler cannot determine if the work completed.

Common recover queue

The common recover queue (CVQ) is a queue of DFSMSHsm volume restore requests from FRRECOV COPYPOOL, FRRECOV TOVOLUME, and RECOVER TOVOLUME with FROMDUMP commands that is shared by multiple DFSMSHsm hosts. See Table 48 on page 474, for details on which command variations are supported. The common recover queue is implemented through the cross-system coupling facility (XCF) to balance the volume restore workload across the resources in each DFSMSHsm host in an HSMplex. Individual DFSMSHsm hosts are defined to XCF as a member of a group to be used for volume restore processing. By distributing volume restore work to all the members in the group, the number of available tasks to perform the work is increased which also improves throughput of the volume restore function as the workload is no longer concentrated on a single host's address space.

Table 48. CVQ Scope. The CVQ Scope table uses three columns where the first row is the header for each column.

Command/Function	Qualifications	Supported in CVQ
FRRECOV COPYPOOL	None. From dump determined by HSM for latest version	YES
	FROMDUMP specified	YES
	FROMDASD specified or latest version from DASD determined by HSM	N/A
FRRECOV TOVOLUME (Individual volume restore)	FROMCOPYPOOL From dump determined by HSM for latest version	YES
	FROMCOPYPOOL FROMDUMP specified	YES

Table 48. CVQ Scope. The CVQ Scope table uses three columns where the first row is the header for each column. (continued)

Command/Function	Qualifications	Supported in CVQ
	FROMCOPYPOOL FROMDUMP APPLYINCREMENTAL specified	YES
	FROMCOPYPOOL FROMDASD specified or latest version from DASD determined by HSM	N/A
FRRECOV DSNAME	None	NO
RECOVER DSNAME	FROMDUMP	NO
RECOVER DSNAME	From dump not specified	NO
RECOVER *	TOVOLUME FROMDUMP	YES
	TOVOLUME FROMDUMP APPLYINCREMENTAL	YES

The CVQ scope table shows that the FRRECOV, and RECOVER commands volume restores, as well as the optional APPLYINCREMENTAL, are eligible to be processed in the CVQ. The DSNAME function command is not eligible.

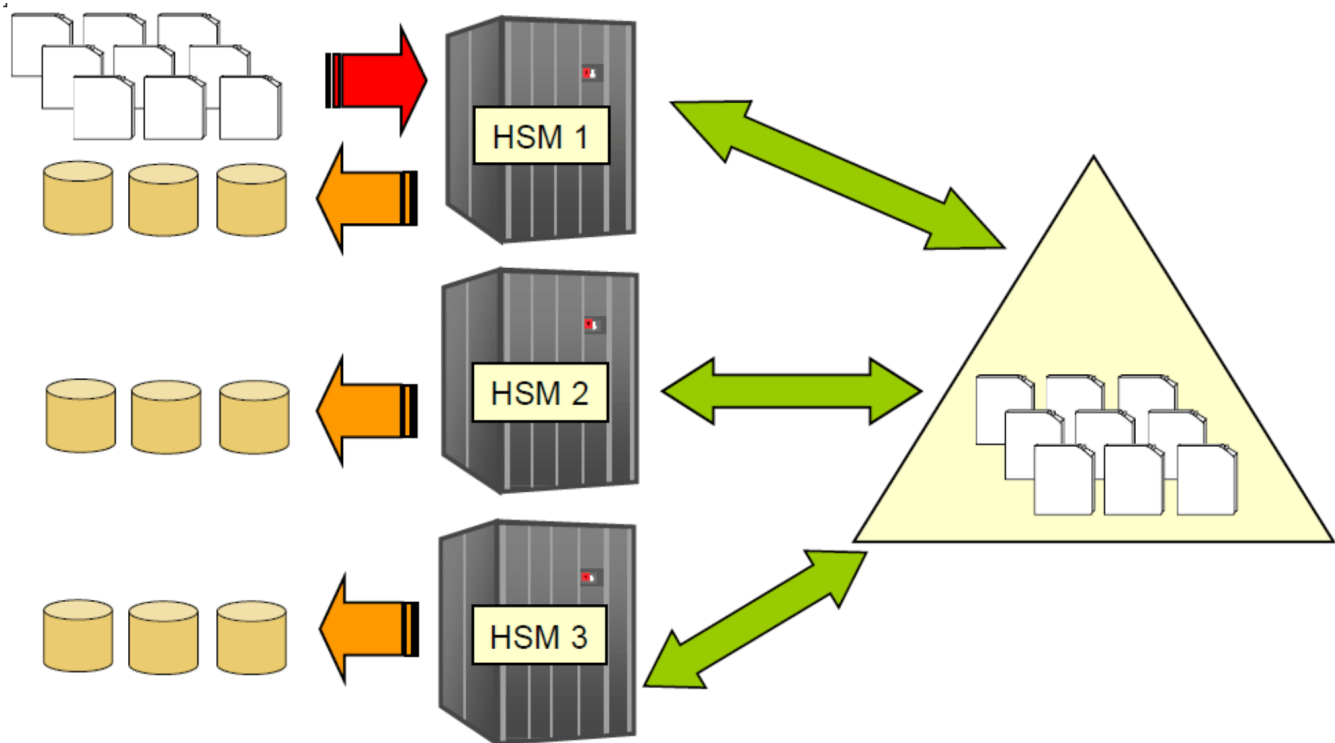


Figure 80. Common Recover Queue

Figure 80 on page 475, shows that the CVQ consists of one or more DFSMSHsm hosts that are connected to an XCF defined group, where submitted volume restore requests will be distributed and processed among the volume restore resources available in the CVQ group, with results returned back to the host where the request originated.

Figure 81 on page 476, shows the CVQ group allows for flexible configurations. This provides the capability to:

- Define multiple queues in the same HSMplex

- Allow group members to both receive and process requests, only process requests, or only receive requests.

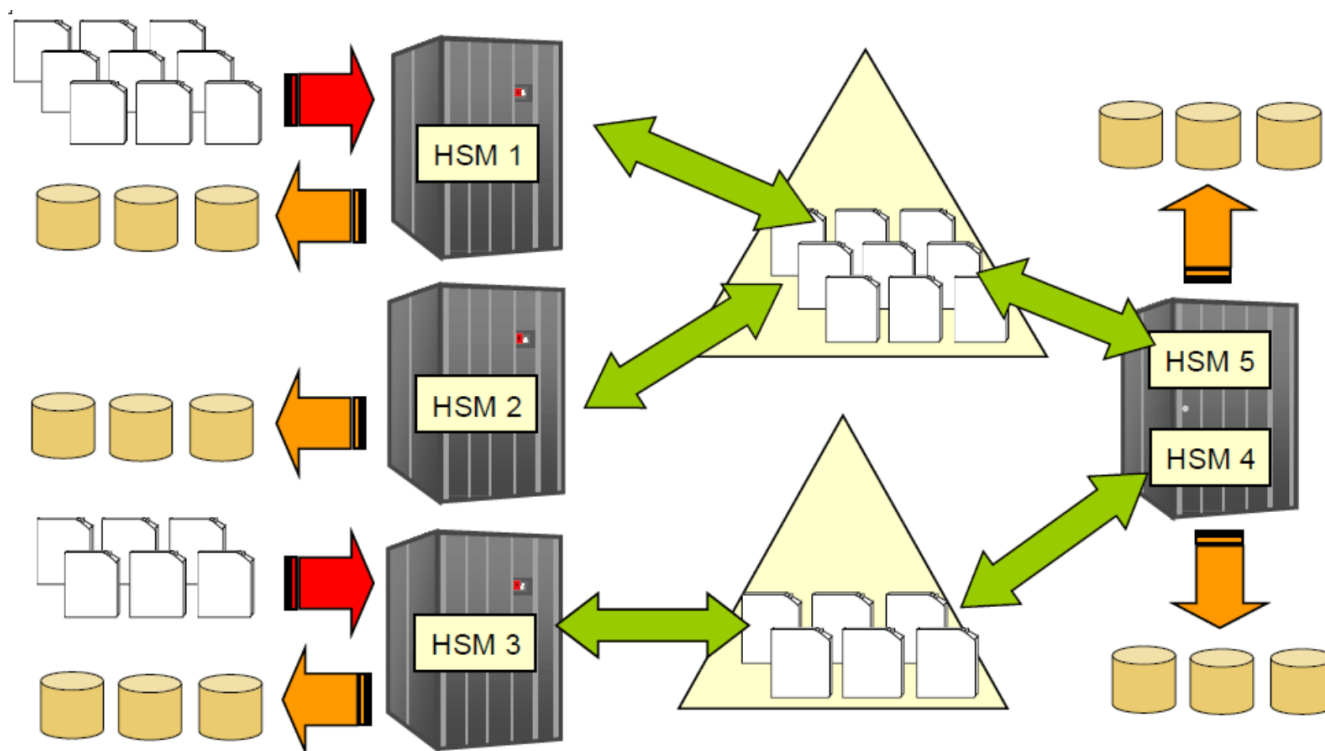


Figure 81. CVQ -- Flexible Configurations

In CVQ environments:

- A CVQ can not span an HSMplex
- The priority of requests is supported in a CVQ
- A host can be connected to only one CVQ
- The progress and status messages are recorded on the host processing the request
- A CVQ can coexist with a CDQ environment with the same host or subset of hosts
- The CVQ flexible configurations allow group members to receive and process requests, only process requests, or only receive requests

Results are returned to the submitting host, but progress and status messages are recorded on the host processing the request. By using the message simplification support along with CVQ processing, each CVQ group member will write all DFSMSshm and DFSMSdss messages related to the processing of a particular copy pool (FRRECOV COPYPOOL FROMDUMP) or individual volume restore from a copy pool (FRRECOV TOVOLUME FROMDUMP) to a single shared message data set. This simplifies diagnostics as all messages are collected from each host into a single data set. The data set is managed by the Master Scheduler (MS) host for the CVQ. See the SETSYS FASTREPLICATION(MESSAGE DATASET(YES)) option in Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSshm control parameters,” on page 1057.

Types of hosts used in a common recover queue

The master scheduler (MS) is the DFSMSshm host that manages all of the volume restore requests in the Common Recover Queue (CVQ). All volume restore requests are sent to the master scheduler, and it assigns work to all of the other hosts eligible in the group to process requests from the CVQ. Each group needs a master scheduler to manage CVQ processing for the group; otherwise, volume restore requests are processed locally.

The other principal types of DFSMSshm hosts are the submitting host and the processing host.

Submitting host

Host that receives requests from commands and, using XCF messaging, sends it to the master scheduler host. Performs command complete processing and posts the user that the command has completed.

Master scheduler (MS)

- Accepts requests from submitting host members and locally submitted requests, and maintains them as a single common queue of all requests for the CVQ function. The MS can be the submitting host. In this case, the command is not sent
- Assigns the requests to the available tasks across all of the eligible hosts (processing hosts) that are connected to the CVQ, while balancing the requests among all the hosts.
- Manages "work completed" messages received from a processing host to support functions like stacking.
- Sends "command complete" messages back to the submitting host.

Processing host

- Receives assigned work requests from the master scheduler and places them on its local queue for processing.
- Sends "work complete" messages back to the master scheduler with specific protocols to support functions like stacking.
- The processing host could be the submitting host and/or the master scheduler host.

Figure 82 on page 478, shows a CVQ group where:

- HSM1 is the submitting and processing host
- HSM2 is the master scheduler and processing host. It receives all requests, manages them on a single queue, and distributes the requests among the eligible hosts connected to the common queue, through XCF messaging.
- HSM3 is a processing host.

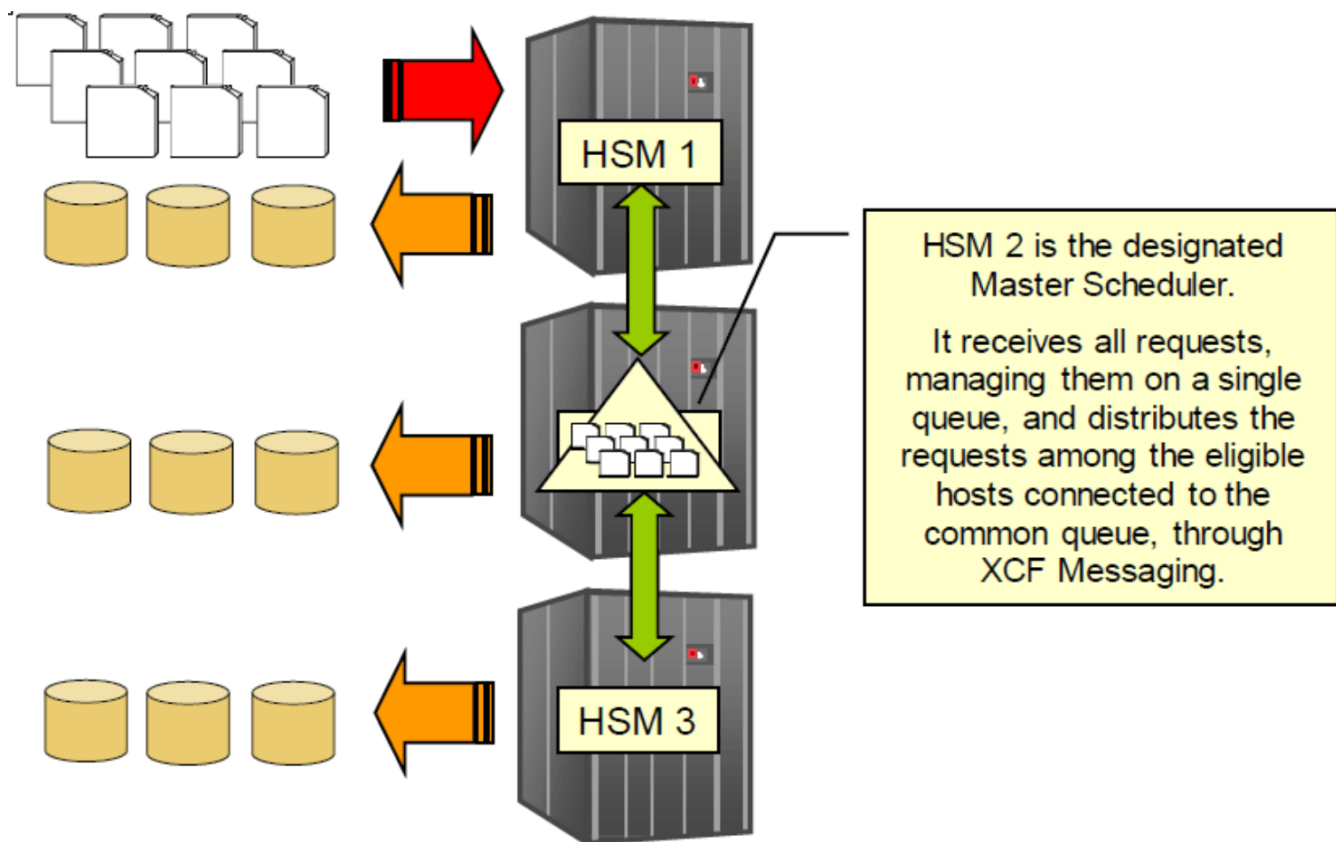


Figure 82. CVQ Host Types

Connecting to a common recover queue

The SETSYS COMMONQUEUE(RECOVER(CONNECT(*basename*))) is used to define the CVQ group name to XCF and to connect this DFSMSHsm host to it. The *basename* is a four-character suffix to the HSM designated group name prefix. It has the following format: ARCV*basename*.

The first DFSMSHsm host to issue the SETSYS command with CONNECT option will define the XCF group, and connect itself to this group, while subsequent DFSMSHsm hosts will connect to the XCF group defined by the first host.

For example, a DFSMSHsm host is defined as a member of the XCF group named, ARCVQUE1, as follows:

```
SETSYS CQUEUE(RECOVER(CONNECT(QUE1))
```

The current SETSYS setting and the MS status can be displayed with QUERY SETSYS (message ARC1500I) and QUERY ACTIVE (message ARC1560I) respectively.

Master scheduler

The first member to join the CVQ group that is a master scheduler candidate becomes the master scheduler host for the CVQ group.

The master scheduler candidate (MSC) setting can be defined:

- In conjunction with the CONNECT option of the SETSYS command:

```
SETSYS COMMONQUEUE(RECOVER(CONNECT(basename) MSC(YES|NO))
```

MSC defaults to YES.

- Separately, to set the MSC state before the CONNECT option, or to change it afterward:


```
SETSYS COMMONQUEUE(RECOVER(MSC(Y))
```

For example, an MSC(N) option on the SETSYS command issued for each host prior to the CONNECT option ensures that no host becomes a master scheduler candidate. In this case, the CVQ group status is DISABLED, and no CVQ processing occurs. All work is processed locally. After all SETSYS commands with the CONNECT option have been issued, the MSC state for one or more hosts could be changed through the MSC(Y) option, to allow that host to take on the master scheduler responsibilities. A DFSMSHsm host with a MSC(Y) state is eligible to take over when the current master scheduler host fails, or disconnects from the group.

It is recommended that all of your DFSMSHsm hosts be defined to be master scheduler candidates. This ensures that one host will take on the master scheduler responsibilities when a CVQ group is defined, and ensures that another host will take on the master scheduler responsibilities once the current one is removed from the group due to a disconnection or system errors. However, if a system might not have enough capacity to perform this work, you may want to define it to not be a master scheduler candidate.

Query commands

To display the current SETSYS setting, use the QUERY SETSYS command and look for message ARC1500I. To display the master scheduler status, use the QUERY ACTIVE command and look for message ARC1560I.

Using the DISPLAY XCF,GROUP command, the output for this command will identify the group as ARCVQUE1 and identify the member name as HOST#*hsmid* for this host.

For more information

For more information, refer to Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.

Placing requests on the common recover queue

The common recover queue (CVQ) is a queue of DFSMSHsm volume restore requests. These volume restore requests include all requests from RECOVER *, FRRECOV COPYPOOL, FRRECOV TOVOLUME when FROMDUMP is specified, or when DFSMSHsm determines that the most current version is from a dump (volume restores). APPLYINCREMENTAL can be applied. However, data set recovers are not eligible for the CVQ. A host that participates in CVQ processing places its volume restore requests onto the CVQ for processing. The following items apply to placement status:

- To verify that a host is connected to the CVQ, issue the QUERY ACTIVE command. Look for CONNECTION STATUS of CONNECTED in message ARC1560I for the Common Recover Queue.
- HOLD RECOVER or HOLD FRRECOV(TAPE) issued on a host prevents tasks for this host from being selected by the master scheduler as a processing host.
- HOLD RECOVER or HOLD FRRECOV(TAPE) issued on the host that issued the volume restore request places that request on the CVQ, when the master scheduler is not disabled. Note that the MASTER SCHEDULER STATUS is QUERY ACTIVE, *not* DISABLED(HELD).

Both WAIT and NOWAIT type requests are placed as the master scheduler is available to assign and process requests in the CVQ. Tasks for this host subsequently are not assigned by the master scheduler for processing.

- HOLD RECOVER and FRRECOV(TAPE) issued on the master scheduler host prevents it from assigning and processing requests. Note that NOWAIT request types are placed on the CVQ (even though the function was held on the master scheduler), while WAIT type requests are failed. A HOLD of a specific type (RECOVER or FRRECOV) affects only the assignment of that type by the master scheduler. To keep the master scheduler responsibilities intact, that is, to prevent tasks on the MS host from being used by the CVQ, issue an SETSYS MAXDUMPRECOVERTASKS(0). Alternatively, you can transition the MS responsibilities to another host with SETSYS CQUEUE(RECOVER(MSC(N))), prior to issuing the HOLD.

- HOLD RECOVER and FRRECOV(TAPE), issued on all the hosts in the CVQ places the master scheduler in the DISABLED state. Use the messages issued in response to the QUERY ACTIVE to verify this state. Note that the MASTER SCHEDULER STATUS is DISABLED(HELD). When DISABLED:
 - New WAIT request types are failed locally
 - New NOWAIT request types are queued locally
 - Queued WAIT request types in the CVQ master scheduler host are failed
- SETSYS MAXDUMPRECOVERTASKS(0) issued on the host that issued the volume restore request places the request on the CVQ. Tasks for this host subsequently are not assigned by the MS for processing.
- Any request that is not submitted to the master scheduler host (for example, due to errors) is placed on the local queue of the host that issued the request.
- Requests submitted on the master scheduler host are placed on its local queue
- Requests that are sent to the MS host from other members are placed on the local MS queue, then selected to run locally on the MS, or assigned to be processed on another member in the CVQ group.
- The default priority of WAIT volume restore requests over NOWAIT volume restore requests does not change with the CVQ. However, the ALTERPRI command supports change requests that are currently queued to the CVQ.

Selecting requests from the common dump queue

The master scheduler is the DFSMSHsm host that manages all of the volume restore requests in the CVQ. All volume restore requests are sent to it, and it assigns work to all of the other hosts eligible to process requests from the CVQ. The following items apply to selecting requests to be processed by the CVQ:

- The master scheduler selects hosts to process requests based on the current utilization of available tape resources in each host relative to each other. The first host with the lowest utilization will be selected to process the request.
- A host's resources are not available if the HOLD RECOVER, FRRECOV or SETSYS EMERGENCY commands has been issued or if the SETSYS MAXDUMPRECOVERTASKS(0) has been issued. In each case the system will indicate back to the MS with XCF messaging that this host has no resources available.
- Selected hosts that are not the Master Scheduler are assigned the work with XCF messaging. The assigned processing host selects the task to use.
- When the master scheduler selects its own volume restore task to process a CVQ request, the task is attached and assigned the work.

Processing requests off of the common recover queue

The master scheduler selects a host member of the CVQ to process a request placed on the CVQ. This host could be the master scheduler itself, another member in the CVQ and even the member that placed the request on the CVQ that the request originated from. This selected member is the processing host and is responsible for selecting an available task to use for the copy to tape processing that it will perform. The following items apply to a processing host in the CVQ:

- The assigned processing host selects an available task to process the request.
- Status and progress messages along with diagnostic trace data that is generated are recorded on the processing host. A single copy pool recover request may generate many different volume restores with each one processed by individual CVQ group members. Messages and diagnostic data are placed on the host processing the request, making it difficult to put together the message flow of an individual volume restore from all the processing hosts.
- By using the message simplification support along with CVQ processing, each CVQ group member will write all DFSMSHsm and DFSMSdss messages related to the processing of a particular copy pool to a single shared message data set that is managed by the MS host. This simplifies the understanding of the message flow for the original fast replication copy pool command as all messages are collected from each host into this single data set. XCF messages are used to perform this function. Refer to the SETSYS

FASTREPLICATION(MESSAGE DATASET(YES)) option in Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.

- As volumes complete processing, the selected processing host, in conjunction with the MS host, determines if another volume restore is needed by the currently mounted tape. If this is the case, that task is assigned to process the additional volume restore. This is accomplished with XCF messaging.
- The processing host can be both the submitting host and MS host, either the submitting host or the MS host. And it can be neither the submitting host nor the MS host.
- When the SETSYS COMMONQUEUE(RECOVER(CONNECT(*basename*))) is specified, the preference by the system will be to utilize the available recover tasks as defined by MAXDUMPRECOVERTASKS on all the recover group members in the HSMplex. However, the number of concurrent tasks is limited by the stacking of the volumes on the dump tapes. For example, if 100 volumes are being recovered, and all volumes are on a single tape, then only a single recover task will be used.

Using QUERY, CANCEL and ALTERPRI in the common recover queue

The QUERY, CANCEL and ALTERPRI commands support the Common Recover Queue (CVQ).

The following items apply to QUERY processing:

- QUERY SETSYS displays the current SETSYS settings for the CVQ on the host that issued the QUERY command. Message ARC1500I displays the SETSYS settings on this host.
- QUERY ACTIVE displays the current state of the CVQ and active requests running it.
 - The state and active requests reported on are those that originated from the host issuing the QUERY command
 - Active requests reported could be active in any member in the CVQ. XCF messaging is used to retrieve active requests by sending the QUERY command to the Master Scheduler. These requests could be running in any member of the CVQ group and must match the filter criteria of the query. In this case, the active requests must have originated from the host issuing the QUERY command. The generated messages on the MS are returned to the query host using XCF and in turn the query host issues them. The query host then goes on to report its own active requests.
 - When the query host is also the MS, no XCF messaging is used to obtain the active requests as the CVQ queues are located on the MS host.
- QUERY REQUEST or USER display requests that originated from the host issuing the QUERY command that match the REQUEST or USER filter criteria.
 - Active requests reported could be active in any member in the CVQ and queued in the MS. XCF messaging is used to retrieve these requests by sending the QUERY command to the Master Scheduler. These active requests could be running in any member of the CVQ group or queued in the MS and must match the filter criteria of the query. In this case, the requests must have originated from the host issuing the QUERY command and match the REQUEST or USER criteria on the QUERY command. The generated messages on the MS are returned to the query host using XCF messaging and in turn the query host reports them. The query host then goes on to report its own requests.
 - When the query host is also the MS, no XCF messaging is used to obtain the queued and active requests as the CVQ queues are located on the MS host.
- QUERY WAITING displays the number of requests that are queued in the CVQ that originated from the host issuing the QUERY command.
 - Queued requests in the CVQ will be queued in the MS. XCF messaging is used to retrieve these queued request numbers by sending the QUERY command to the Master Scheduler. These queued requests in the MS must match the filter criteria of the query. In this case, the requests must have originated from the host issuing the QUERY command. The generated messages on the MS are returned to the query host using XCF messaging and in turn the query host reports them. The query host then goes on to determine its own queued request numbers, combines them with what was returned from the CVQ, then reports the totals.
 - When the query host is also the MS, no XCF messaging is used to obtain the queued request numbers as the CVQ queues are located on the MS host.

- **QUERY COMMONQUEUE(RECOVER)** displays all queued and active requests being managed by the CVQ. Use this query to get a snapshot of all CVQ activity from any host connected to the CVQ.
 - Queued and active requests in the CVQ are managed by the MS for the CVQ group. XCF messaging is used to retrieve these requests by sending the **QUERY** command to the Master Scheduler. These queued and active requests in the MS have no matching criteria to satisfy. All requests will be reported. The generated messages on the MS are returned to the query host using XCF messaging and in turn the query host reports them.
 - When the query host is also the MS, no XCF messaging is used to obtain the queued and active requests as the CVQ queues are located on the MS host. The filter criteria is all queued requests running in the CVQ regardless of where the volume restore request command originated.
 - Messages **ARC1562I** and **ARC1563I** are reported for the active and queued requests in the CVQ.
- The **CANCEL REQUEST** or **USERID** command cancels queued requests in the CVQ for requests originated in the host that issued the **CANCEL** command that also matches the **REQUEST** or **USERID** filter criteria.
 - Queued requests in the CVQ are managed by the MS for the CVQ group. XCF messaging is used to cancel these requests by sending the cancel command to the Master Scheduler. These queued requests in the MS must have originated from the host issuing the **CANCEL** command and match the filter criteria of the **REQUEST** and **USERID** parameters. When matching, the requests are cancelled. The generated messages on the MS are returned to the cancel host using XCF messaging and in turn the cancel host reports them.
 - The **QUERY COMMONQUEUE(RECOVER)** command can be used to determine the requests running in the CVQ which will also identify the **HOST** id of where each request originated from.
 - When the cancel host is also the MS, no XCF messaging is used to cancel the queued requests as the CVQ queues are located on the MS host.
 - To cancel an active request in the CVQ, issue the **CANCEL TCBADDRESS** on the host that this request is active in.
- The **ALTERPRI REQUEST** or **USERID** command changes the priority of queued requests in the CVQ for requests originated in the host that issued the **ALTERPRI** command that also matches the **REQUEST** or **USERID** filter criteria.
 - Queued requests in the CVQ are managed by the MS for the CVQ group. XCF messaging is used to alter the priorities of these requests by sending the **ALTERPRI** command to the Master Scheduler. These queued requests in the MS must have originated from the host issuing the **ALTERPRI** command and match the filter criteria of the **REQUEST** and **USERID** parameters. When matching, the requests priority is altered. The generated messages on the MS are returned to the **ALTERPRI** host using XCF messaging and in turn the **ALTERPRI** host reports them.
 - The **QUERY COMMONQUEUE(RECOVER)** command can be used to determine the requests running in the CVQ which will also identify the **HOST** id of where each request originated from.
 - When the **ALTERPRI** host is also the MS, no XCF messaging is used to alter the priority of the queued requests as the CVQ queues are located on the MS host.

Disconnecting from the common recover queue

SETSYS COMMONQUEUE(RECOVER(DISCONNECT)) is used to request that a member of the CVQ group that this host is currently connected too is to leave the group. See the syntax of this command for specifics on other options and defaults. The following items apply to this disconnect process:

- All CVQ requests being processed on this host and requests that originated from this host and selected to be processed in the CVQ are waited upon for completion before disconnecting the host from the CVQ. This host will no longer be selected to process new volume restore work from the CVQ.
- A **HOLD RECOVER** or **FRRECOV** issued prior to the disconnect command will prevent queued requests from being selected to be processed. After the disconnect has completed, a **RELEASE RECOVER** or **FRRECOV** will allow the queued requests to be processed locally.

- A disconnection of the host that is currently the Master Scheduler will wait for all CVQ active and queued work to complete prior to disconnecting. Once disconnected another member in the CVQ group that has been previously defined as a Master Scheduler candidate will automatically take over the responsibilities for it. A HOLD RECOVER or FRRECOV prior to a disconnection will prevent queued requests from being selected for processing. A subsequent RELEASE RECOVER or FRRECOV after the disconnecting will allow these requests to be processed locally.
- The QUERY ACTIVE command response in message ARC1560I will verify the disconnection.
- Volume restore resources available on this host after the disconnection has completed are not known or available to the remaining CVQ members. All volume restore requests received will be placed on this host's local queue and processed locally.
- For shutdown procedures, a DISCONNECT should first be issued on each host being removed from the HSMplex to allow CVQ requests to complete first.
- The DISCONNECT FORCE option should be reserved for emergency situations. With FORCE specified, requests that originated from this host and selected to be processed in the CVQ are not waited upon for completion before disconnecting. Messages will be issued to indicate that a request has been failed but still may complete in the CVQ.

For more information, refer to [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.](#)

Processing of failures in the common recover queue

The CVQ group support uses XCF facilities to detect when members lose the connection to the CVQ. The connection might be lost for one or more of the following reasons:

- An host requested a disconnection with the SETSYS COMMONQUEUE(RECOVER(DISCONNECT))
- An XCF related failure causes it to be removed from the group
- A z/OS® image fails

When the host that is now not defined was the master scheduler, then master scheduler responsibilities are managed by a master scheduler candidate from the CVQ group. All hosts would have been notified that the MS was the one that went not defined. For each of these hosts, any work that was placed on the CVQ prior to this event is placed on the CVQ, and managed by the new master scheduler host. This may result in some volumes being processed again.

When the host that is now not defined is not the master scheduler, the master scheduler host is notified that this host is no longer defined to the group. The CVQ in this case searches for requests previously selected to run on this now not defined host, and fails them with ARC1127I or ARC1829I, and with return code 105. This indicates that a host in the common queue group is not defined, and that the work was selected by the master scheduler to run in this host of the CVQ, but the host cannot determine if the work completed.

Chapter 11. Availability management procedures

This topic details specific backup, recovery, and full-volume dump procedures associated with DFSMSHsm availability management.

Backup and recovery procedures

This topic tells you how to perform the following functions:

- Interrupt and restart backup processing
- Define a new automatic backup start window
- Back up and recover a catalog

Interrupting and restarting backup processing

To stop the first phase of automatic backup, which is the backup of the control data sets, you must use the STOP command.

You can use the following methods to interrupt automatic backup processing during its last three phases. The last three phases of automatic backup processing consist of the following functions:

- Movement of backup versions
- Backup of migrated data sets
- Backup of DFSMSHsm-managed volumes having the automatic backup attribute

There are four methods you can use to interrupt automatic backup processing during its last three phases:

- **Holding backup processing**—You can hold backup processing by specifying the BACKUP, BACKUP(AUTO), or ALL parameter of the HOLD command. If you hold backup processing with the ENDOFDATASET parameter while DFSMSHsm is moving backup versions or backing up migrated data sets, processing ends after DFSMSHsm has finished processing the current backup version. If you hold backup processing with the ENDOFVOLUME parameter while DFSMSHsm is moving backup versions or backing up migrated data sets, processing ends after the current phase of automatic backup has completed. If you hold backup processing with the ENDOFVOLUME parameter while DFSMSHsm is backing up managed volumes, DFSMSHsm continues processing the current volumes and does not process any new volumes. Otherwise, DFSMSHsm finishes processing the current data set. DFSMSHsm does not start any new volume backup tasks.
- **Disabling backup processing**—You can disable backup processing by specifying the NOBACKUP parameter of the SETSYS command. If you disable backup processing while DFSMSHsm is moving backup versions or backing up migrated data sets, processing ends after DFSMSHsm has finished processing the current backup version. If you disable backup processing while DFSMSHsm is backing up DFSMSHsm-managed volumes, DFSMSHsm finishes processing the current data set and does not start any new volume backup tasks.
- **Placing DFSMSHsm in emergency mode**—You can place DFSMSHsm in emergency mode by specifying the EMERGENCY parameter of the SETSYS command. If you place DFSMSHsm in emergency mode while DFSMSHsm is moving backup versions or backing up migrated data sets, processing ends after DFSMSHsm has finished processing the current data set.

If you are in a multiple DFSMSHsm-host environment and have enabled the secondary host promotion function, another DFSMSHsm host takes over the unique functions of the primary DFSMSHsm host that you have put in emergency mode. Those functions include automatic movement of backup versions from ML1 to tape, and automatic backup of migrated data sets. If promotion occurs during an automatic backup window (as defined on the primary host), and it is not past the latest start time, the promoted host takes over where the primary host left off. If the promoted host was an automatic backup host before being promoted, it performs backups of DFSMSHsm-owned volumes after performing the

automatic movement of backup versions from ML1 to tape, and automatic backup of migrated data sets. If the promoted host was *not* an automatic backup host before being promoted, it does not back up DFSMSHsm-owned volumes.

If you place DFSMSHsm in emergency mode while DFSMSHsm is backing up DFSMSHsm-managed volumes, DFSMSHsm finishes processing the current data set before ending the volume backup tasks that are in progress and does not start any new volume backup tasks.

- **Stopping DFSMSHsm**—You can stop DFSMSHsm by entering the MVS or DFSMSHsm STOP command. If you shut down DFSMSHsm while DFSMSHsm is moving backup versions or backing up migrated data sets, processing ends after DFSMSHsm has finished processing the current data set. If you shut down DFSMSHsm while DFSMSHsm is backing up DFSMSHsm-managed volumes, DFSMSHsm finishes processing the current data set before ending the volume backup tasks that are in progress and does not start any new volume backup tasks. If you issue the DFSMSHsm STOP command while DFSMSHsm is backing up the control data sets or the journal data set, DFSMSHsm finishes processing the current data set and does not start backing up any other control data sets or the journal data set.

If you are in a multiple DFSMSHsm-host environment, have enabled secondary host promotion, and you have issued a STOP PROMOTE command, another DFSMSHsm host takes over the unique functions of the primary DFSMSHsm host that you have stopped. Those functions include automatic movement of backup versions from ML1 to tape, and automatic backup of migrated data sets. If promotion occurs during an automatic backup window (as defined on the primary host), and it is not past the latest start time, the promoted host takes over where the primary host left off. If the promoted host was an automatic backup host before being promoted, it performs backups of DFSMSHsm-owned volumes after performing the automatic movement of backup versions from ML1 to tape, and automatic backup of migrated data sets. If the promoted host was *not* an automatic backup host before being promoted, it does not back up DFSMSHsm-owned volumes.

You can restart automatic backup processing by entering the corresponding commands if certain conditions are met:

- RELEASE BACKUP or RELEASE ALL
- SETSYS BACKUP
- SETSYS NOEMERGENCY
- START DFSMSHSM (MVS command)

If secondary host promotion did not continue backup processing where the primary DFSMSHsm host left off, automatic backup restarts at the point of interruption if the date of the defined start time for the current automatic backup window is the same as the date that automatic backup last started from the beginning (same cycle date). When automatic backup restarts, it uses the same set of daily backup volumes it was using when it first started. Those daily backup volumes are the ones assigned to the day in the backup cycle that was in effect when automatic backup started.

After you restart automatic backup processing, DFSMSHsm continues with the next DFSMSHsm-managed volume under automatic backup control. However, there are a few considerations:

- If you issued a DFSMSHsm STOP command and then a START DFSMSHsm command, DFSMSHsm can start automatic backup at the point of interruption only:
 - If you add the primary volumes under automatic backup control in the same order in each startup of DFSMSHsm
 - If you do not change the volumes associated with the storage groups
- If you interrupted a BACKVOL PRIMARY command, DFSMSHsm can resume backup at the start of the next volume only if you issued HOLD BACKUP or SETSYS NOBACKUP. You must issue RELEASE BACKUP, RELEASE ALL, or SETSYS BACKUP during the same startup of DFSMSHsm if you want the BACKVOL PRIMARY command to resume processing.
- DFSMSHsm does not restart a BACKVOL command issued for one volume.

Related reading

For more information about using the secondary host promotion function, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Defining a new automatic backup start window

As you increase the number of volumes that DFSMSHsm backs up, you might want to change the automatic backup start window. In a test environment, you might want daily automatic backup to run more than once on a single day. This topic describes how to define a new start window.

Defining a new start window when automatic backup is running

You can change the automatic backup start window at any time, even when automatic backup is currently running on the affected DFSMSHsm host.

If automatic backup is running but completes before the newly defined start time is reached, automatic backup processing starts again at the newly defined start time.

If automatic backup is running and processing continues beyond the newly defined start time, automatic backup processing does not start again until the next time the newly defined start time occurs.

If you specify a quiesce time for the new automatic backup start window while automatic backup is running and the current time is outside of the newly specified window, automatic backup processing stops.

If automatic backup is released or DFSMSHsm is restarted during the automatic backup start window, automatic backup starts, unless prevented by one of the commands discussed in [“Interrupting and restarting backup processing”](#) on page 485, or by denying permission to the SETSYS REQUEST command.

Defining a new start window when automatic backup is not running

If you define a planned start time that is later than the time that automatic backup last ended, automatic backup starts at the new planned start time if it has not already processed all the requested volumes on this day. Otherwise, automatic backup starts at the new planned start time on the next day.

If you define a planned start time that is earlier than the time that automatic backup last ended, automatic backup starts only if it has not run to completion on this day.

Backing up and recovering a catalog

There are no special requirements to back up a catalog. Either the catalog is backed up during volume backup of the volume it is on, or it can be backed up by the BACKDS command. Although the catalog is open when the backup occurs, DFSMSHsm enqueues on the catalog to prevent it from being updated while DFSMSHsm makes the backup version.

Because this enqueue prevents access to the catalog by any other user, for the duration of the backup, it is imperative that the elapsed time of the backup be kept as short as possible.

Defining the user catalog with as large a data control interval size as possible dramatically reduces the elapsed time to back up a user catalog. Therefore, the user catalogs should be defined with a data control interval size of one half track.

Use the following procedure to recover a backup version of a catalog:

1. Purge all jobs that allocate the volume.
2. Issue the F CATALOG,OPEN(*volser*) command to determine which catalogs on the target volume are allocated.
3. Issue, for each catalog that is open:

```
F CATALOG,UNALLOCATE(catalogname.)
```

to cause the catalog address space to close the catalog and deallocate it.

4. Issue the RECOVER command to DFSMSHsm.

If DFSMSdss is installed, aliases of the catalog are backed up and recovered. When DFSMSHsm attempts to recover a catalog, the following rules apply:

- If the catalog does not exist, it is defined, along with its aliases, from the backup version.
- If the catalog is empty, any aliases for that catalog in the system catalog are not deleted. Aliases that were backed up are defined for the recovered catalog if they do not already exist in the system catalog.
- If the catalog is not empty, it is deleted and redefined from the backup version. Any aliases for that catalog in the system catalog are not deleted. Aliases that were saved with the backup version are not added to the system catalog.

The recovery of a catalog has the following restrictions:

- A catalog cannot be recovered with a new name.
- Aliases in the catalog are not backed up or recovered unless DFSMSdss is installed on the system.
- If you need to recover the master catalog, you must recover it on a different z/OS image.
- You must be a DFSMSHsm-authorized user.
- Recovery of a non-SMS managed catalog fails with message ARC1001I, return code 58, reason code 30, when the catalog is recovered to a volume that is not the same as the volume it was backed up from. (Recovery to another volume would cause DFSMSHsm to lose RACF discrete-profile security protection).
- If the catalog contains entries for VSAM data sets that are protected by RACF discrete profiles, IBM recommends that you recover the catalog to the same volume it was on when it was backed up. If the catalog is recovered to a different volume, all VSAM data sets that were protected by a discrete profile are now either unprotected or protected by a generic profile that might not provide the level of access and security desired.

The RACF profile of a VSAM data set reflects the catalog volume. DFSMSHsm does not update these profiles when it recovers a catalog. To restore the level of protection that a discretely-protected VSAM data set had before the catalog was recovered to a different volume, you have to update the data set's discrete profile to indicate the new volume. The task of deciding whether to update a discretely-protected VSAM data set's discrete profile, and making the update, can be left to the owner of the data, if an installation so desires.

When recovering non-SMS-managed catalogs, you can use the TOVOLUME parameter to specify the volume from which the catalog was backed up.

When recovering SMS-managed catalogs, you can assign a storage class with the GUARANTEED SPACE attribute as a means of requesting that the catalog is recovered to its original volume.

- The HRECOVER command will fail if an ALIAS is substituted for the data set name of a catalog, even if the user is DFSMSHsm-authorized.

Backing up and recovering RLS user catalogs

If you use DFSMSHsm to back up and recover RLS user catalogs, ensure that the DFSMSHsm authorized user identification (UID) has been granted READ access to the IGG.CATLOCK FACILITY class profile and ALTER access to the user catalogs. DFSMSHsm invokes DFSMSdss to back up and recover RLS ICF user catalogs. During recover of an RLS user catalog, DFSMSHsm specifies the BCSRECOVER(LOCK) parameter, and DFSMSdss will perform LOCK and UNLOCK on behalf of DFSMSHsm when the user catalog is preallocated. The RECOVER command will fail if DFSMSHsm UID has insufficient authority, or if the IGG.CATLOCK FACILITY class profile has not been defined.

Related reading

For more information on using RLS catalogs and IGG.CATLOCK FACILITY class profile, see [z/OS DFSMS Managing Catalogs](#). For more information on the DFSMSdss BCSRECOVER(LOCK|SUSPEND) parameter, see [z/OS DFSMSdss Storage Administration](#).

Restoring a catalog from a dump copy

You cannot use DFSMSHsm to restore a catalog from a dump copy. DFSMSHsm issues a LOCATE for the catalog name before invoking DFSMSdss, which causes the catalog to be opened. DFSMSdss does not restore a catalog that is open. You can, however, use DFSMSdss directly to restore a catalog from a DFSMSHsm-owned dump copy.

Use the following procedure to restore a catalog:

1. Use the procedure in topic [“Restoring data sets or volumes directly with DFSMSdss” on page 458](#) to find the appropriate dump copy and to invoke DFSMSdss.
2. Ensure that all dump tape volumes are available at the time of the restore.
3. Follow the procedure, with one exception, described in [“Backing up and recovering a catalog” on page 487](#) for recovering a backup version of a catalog. The exception is in the last step. Instead of starting DFSMSHsm and recovering the catalog, use the DFSMSdss RESTORE command to restore the catalog.

Note: You cannot restore a catalog that was dumped with compression, as described in [“Invoking full-volume dump compression” on page 457](#).

Full-volume dump procedures

This topic explains:

- Interrupting and restarting dump procedures
- Defining a new automatic dump start window

Interrupting and restarting automatic dump processing

You can use the following methods to interrupt automatic dump processing during the three phases of automatic dump. These three phases are:

- Automatic deletion of expired dump copies
- Automatic dump of DFSMSHsm-managed or owned volumes
- Deletion of excess volume table of contents (VTOC) copy data sets

There are four methods you can use to interrupt automatic dump processing:

- **Hold dump processing** — You can hold dump processing by specifying the DUMP, DUMP(AUTO), or ALL parameter of the HOLD command. If you hold dump processing with the ENDOFDATASET parameter while DFSMSHsm is automatically deleting expired dump copies or excess dump VTOC copy data sets, processing ends after DFSMSHsm has finished processing the current dump generation. If you hold dump processing with the ENDOFVOLUME parameter while DFSMSHsm is automatically deleting expired dump copies or excess VTOC copies, processing ends after the current phase of automatic dump has completed. If you hold dump processing with the ENDOFVOLUME parameter, DFSMSHsm continues processing the current volumes and does not process any new volumes. Otherwise, DFSMSHsm dump processing ends the next time DFSMSdss reads or writes a record. DFSMSHsm does not start any new volume dump tasks.
- **Disable dump processing** — You can disable dump processing by specifying the NOBACKUP parameter of the SETSYS command. If you disable dump processing while DFSMSHsm is automatically deleting expired dump copies or excess dump VTOC copy data sets, processing ends after DFSMSHsm has finished processing the current dump generation. If you disable dump processing while DFSMSHsm is dumping DFSMSHsm-managed volumes, DFSMSHsm dump processing ends the next time DFSMSdss reads or writes a record. DFSMSHsm does not start any new volume dump tasks.
- **Place DFSMSHsm in emergency mode** — You can place DFSMSHsm in emergency mode by specifying the EMERGENCY parameter of the SETSYS command. If you place DFSMSHsm in emergency mode while DFSMSHsm is automatically deleting expired dump copies or excess dump VTOC copy data sets, processing ends after DFSMSHsm has finished processing the current dump generation.

If you are in a multiple DFSMSHsm-host environment, and have enabled the secondary host promotion function, another DFSMSHsm host takes over the unique functions of the primary DFSMSHsm host that you have put in emergency mode. Those functions include automatic deletion of expired dump copies and excess dump VTOC copy data sets. If promotion occurs during an automatic dump window (as defined on the primary host), and it is not past the latest start time, the promoted host restarts automatic dump processing.

If you place DFSMSHsm in emergency mode while DFSMSHsm is dumping DFSMSHsm-managed volumes, DFSMSHsm dump processing ends the next time DFSMSdss reads or writes a record. DFSMSHsm does not start any new volume dump tasks.

- **Stop DFSMSHsm processing** — You can stop DFSMSHsm by entering the MVS or DFSMSHsm STOP command. If you shut down DFSMSHsm while DFSMSHsm is deleting expired dump copies or excess dump VTOC copy data sets, processing ends after DFSMSHsm has finished processing the current dump generation. If you shut down DFSMSHsm while DFSMSHsm is dumping DFSMSHsm-managed volumes, DFSMSHsm dump processing ends the next time DFSMSdss reads or writes a record. DFSMSHsm does not start any new volume dump tasks.

If you are in a multiple DFSMSHsm-host environment, have enabled secondary host promotion, and you have issued a STOP PROMOTE command, another DFSMSHsm host takes over the unique functions of the primary DFSMSHsm host that you have stopped. Those functions include automatic movement of backup versions from ML1 to tape, and automatic backup of migrated data sets. If promotion occurs during an automatic backup window (as defined on the primary host), and it is not past the latest start time, the promoted host takes over where the primary host left off. If the promoted host was an automatic backup host before being promoted, it performs backups of DFSMSHsm-owned volumes after performing the automatic movement of backup versions from ML1 to tape, and automatic backup of migrated data sets. If the promoted host was *not* an automatic backup host before being promoted, it does not back up DFSMSHsm-owned volumes.

You can restart automatic dump processing by entering the corresponding commands:

- RELEASE DUMP or RELEASE ALL
- SETSYS BACKUP
- SETSYS NOEMERGENCY
- START DFSMSHSM (MVS command)

After you restart automatic dump processing, DFSMSHsm continues with the next DFSMSHsm-managed volume under automatic dump control. Automatic dump restarts at the point of interruption if less than 24 hours have passed since automatic dump was last started from the beginning and if the time is still within the dump start window.

Related reading

For more information about using the secondary host promotion function, see [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Defining a new automatic dump start window

As you increase the number of volumes that DFSMSHsm dumps, you might want to change the automatic dump start window. In a test environment, you might want automatic dump to run more than once on a single day.

This topic describes:

- Defining a new start window when automatic dump is running
- Defining a new start window when automatic dump is not running

Defining a new start window when automatic dump is running

You can change the automatic dump start window at any time, even when automatic dump is currently running on the particular DFSMSHsm host. Automatic dump runs when the new planned start time occurs if the currently running automatic dump completes before the planned start time of your newly defined start window. If the currently running automatic dump runs past the new planned start time, automatic dump will not start automatically until the next time the planned start time occurs. However, if automatic dump is released or DFSMSHsm is restarted during the newly defined automatic dump start window, automatic dump starts unless it is prevented by one of the commands discussed in [“Interrupting and restarting automatic dump processing” on page 489](#) or by denying permission to the SETSYS REQUEST command.

Defining a new start window when automatic dump is not running

If you define a planned start time that is later than the time that automatic dump last ended, automatic dump starts at the planned start time.

If you define a planned start time that is earlier than the time that automatic dump last ended, automatic dump starts only if it has not run to completion on this day.

Chapter 12. Maintaining DFSMSHsm control data sets

DFSMSHsm maintains three control data sets:

- backup control data set (BCDS)
- migration control data set (MCDS)
- offline control data set (OCDS)

These control data sets and the DFSMSHsm journal are what make it possible for you to rely on DFSMSHsm's record keeping to perform all of its activities. Therefore, you need backup copies of them to allow you to recover data should the active control data sets become lost or damaged.

DFSMSHsm provides for automatic backup of the control data sets and the journal each time that automatic backup is performed for the system. In addition, you can back up the control data sets by command.

The primary host is the only host that normally performs an automatic CDS backup. If the primary host fails, a secondary host can take over the functions of the failed primary host.

To enable you to recover a lost or damaged control data set, DFSMSHsm maintains a journal of each transaction that occurs to the control data sets. Should a control data set be damaged or lost, you can recover a backup version of the control data set and apply the transactions recorded in the journal to bring the control data set back to current status.

In addition to showing how to manage the control data sets automatically, the method of issuing commands necessary to determine the condition of the data sets and to repair them if necessary is discussed.

Related reading

- For more information about secondary host promotion, see the sysplex topic in [*z/OS DFSMSHsm Implementation and Customization Guide*](#).
- For more information about control data set placement and sizing, see [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Tasks for maintaining the control data sets

The tasks for controlling automatic backup of the control data sets require adding DFSMSHsm commands to the ARCCMDxx member. The tasks for retrieving the backup control data sets require the invocation of CDS recovery utilities. The following tasks maintain the control data sets:

- Code SETSYS CDSVERSIONBACKUP to tell DFSMSHsm how to back up the data sets automatically.
- Manually back up the control data sets using the BACKVOL CDS command, when automatic backup is not used.
- Recover the control data sets by using the enhanced CDS recovery utilities.
- Issue a DFSMSHsm command to determine if the control data sets match the contents of the computing system catalogs and the VTOCs of the DASD volumes.
- Issue commands to repair errors in the control data sets.

Specifying how DFSMSHsm backs up the control data sets

You can use the CDSVERSIONBACKUP parameter of the SETSYS command to control backup of the control data sets. Within the CDSVERSIONBACKUP parameter, subparameters specify the number of backup versions to keep for the control data sets, the device type on which to store the backup versions, whether to run CDS backup in a parallel or nonparallel mode, the names of the backup data sets, and the data mover.

Note: If you are converting from a SETSYS NOCDVERSIONBACKUP environment, you no longer need the MIGCATBU, BAKCATBU, and OFFCATBU DD statements in your DFSMSHsm startup procedure.

Allocating backup data sets

Whether you choose DFSMSdss or IDCAMS as the data mover when copying the control data sets, the backup data sets are allocated in the following manner:

- If backing up to DASD, you must preallocate the backup data sets.
- If backing up to tape, the data sets are dynamically allocated by DFSMSHsm.

The backup data set names are given a low-level qualifier of either Dxxxxxxx (if DFSMSdss is used for copying) or Vxxxxxxx (if IDCAMS is used for copying), where xxxxxxx is the version number. Backup data set names for multiclusder CDS data sets are given a low-level qualifier of DSy.Dxxxxxxx, where y is the volume number and xxxxxxx is the version number. When CDS version backup fails, the data set name of the failed backup is given a low-level qualifier of Xxxxxxxx, where xxxxxxx is the version number. This data set naming convention supports the use of automated CDS-recovery procedures that determine the latest good control data set backup versions from catalog information.

Choosing the data mover to be used

The SETSYS CDSVERSIONBACKUP subparameter DATAMOVER allows you to indicate which of the following datamovers that DFSMSHsm can use to back up the control data sets. (The DFSMSHsm journal is always backed up by sequential I/O.)

- DATAMOVER(DSS) — tells DFSMSHsm to use DFSMSdss logical dump to back up the control data sets
- DATAMOVER(HSM) — tells DFSMSHsm to use IDCAMS EXPORT to back up the control data sets

When you use DFSMSdss—rather than IDCAMS—to copy the data, concurrent copy can be used, and the control data sets will be validated as they are backed up. Concurrent copy allows the CDS backup to complete as soon as the journal has been backed up, so that DFSMSHsm can begin processing requests before the CDS backups are completed.

Rule: If you access the control data sets in RLS mode, you must use DFSMSdss as the data mover.

If you use IDCAMS, you need to examine the number of records copied each day (check the record count in message IDC0005I from IDCAMS EXPORT), because a broken VSAM index for a CDS will likely result in significant data loss in the backup copy. This loss can go unnoticed for a considerable period of time.

Recommendation: DFSMSHsm development strongly recommends that you specify DATAMOVER(DSS), which provides advantages of validation during backup as well as (with the concurrent copy function) reduced serialization of the control data sets. DATAMOVER(HSM) is retained as an option for purposes of coexistence with previous releases.

DFSMSdss as data mover

You can use DFSMSdss and the concurrent copy function to back up the control data sets. Concurrent copy lets you make a *point in time* copy of a control data set even while it is being updated. DFSMSdss also validates the CDS while the CDS is being backed up. This function is similar to AMS EXAMINE.

The use of the concurrent copy function will reduce the duration of complex-wide serialization of the DFSMSHsm CDS resources when performing CDS version backup and, when SETSYS JOURNAL(RECOVERY) is specified, non-intrusive journal backup is used to further reduce the duration (see [“Using non-intrusive journal backup”](#) on page 496). This time reduction increases the availability of DFSMSHsm functions and non-intrusive journal backup allows DFSMSHsm activity to continue during the journal backup. To achieve the highest level of DFSMSHsm availability, you should place the control data sets on SMS-managed devices that support the concurrent copy function, and you should use DFSMSdss to back up the control data sets. If DFSMSdss backs up to tape, each CDS (or each cluster if multi-cluster control data sets are used) is copied to an individual tape drive (in parallel). Use of concurrent copy is required in order to use non-intrusive journal backup.

If a control data set is on an SMS-managed device that supports the concurrent copy function, and the management class attribute BACKUP COPY TECHNIQUE is specified as a value other than STANDARD, DFSMSdss establishes a concurrent copy session. (See [“Concurrent copy management class support”](#) on page 495 for information about the BACKUP COPY TECHNIQUE management-class attribute.) Otherwise, DFSMSHsm directs DFSMSdss to dump the data set without the benefit of concurrent copy. All control data sets, if they are SMS-managed, should be specified with the same SMS backup copy technique.

After all resources have been set up (data sets allocated and tapes mounted), the control data sets are reserved. When using RLS in a multihost environment, DFSMSHsm does an exclusive enqueue rather than a reserve. The reserves, or enqueues, prevent other DFSMSHsm functions that require the control data sets from running.

DFSMSHsm removes the CDS serialization after the following actions have been done:

1. The journal has been backed up.
2. The journal has been nulled (emptied).
3. DFSMSdss indicates that it has established a concurrent copy session for each control data set to be backed up.

All current DFSMSHsm-supported backup DASD and tape devices are supported by the concurrent copy function.

Concurrent copy management class support

When using DFSMSdss as the data mover, the BACKUP COPY TECHNIQUE management-class attribute that is assigned to the control data sets informs DFSMSHsm whether to direct DFSMSdss to use the concurrent copy function when backing up those data sets. (The attribute is used for a similar purpose during automatic or volume backup processing.)

The BACKUP COPY TECHNIQUE attribute is used as follows:

- CONCURRENT PREFERRED (P) - specifies that the concurrent copy facility is preferred. If a concurrent copy session can be established, the data set is backed up using the concurrent copy facility. Otherwise, data set is backed up by DFSMSdss logical dump without concurrent copy.
- CONCURRENT REQUIRED (R) - specifies that the concurrent copy facility is required. If a concurrent copy session cannot be established, the data set will not be backed up.
- STANDARD (S) - specifies that the concurrent copy facility is not to be used, even though it may be available for the data set. Normal DFSMSdss logical dump backup processing is used without concurrent copy.
- VIRTUAL PREFERRED (VP) - specifies that the virtual concurrent copy is preferred. If a virtual concurrent copy session cannot be established the data set is backed up by DFSMSdss logical dump without concurrent copy.
- VIRTUAL REQUIRED (VR) - specifies that the virtual concurrent copy is required. If a virtual concurrent copy session cannot be established, the data set will not be backed up.
- CACHE PREFERRED (CP) - specifies that the cache-based concurrent copy is preferred. If a cache-based concurrent copy session cannot be established the data set is backed up by DFSMSdss logical dump without concurrent copy.
- CACHE REQUIRED (CR) - specifies that the cache-based concurrent copy is required. If a cache-based concurrent session cannot be established, the data set will not be backed up.

If the management class associated with a data set specifies that the concurrent copy BACKUP COPY TECHNIQUE be used, DFSMSdss is invoked specifying the CONCURRENT parameter.

Note: Use these additional considerations when using cache-based concurrent copy to process large data sets or volumes, or when using multiple tasks to process a large number of data sets:

- Use concurrent copy only during periods of light update activity for the data sets or volumes involved.
- Spread multiple concurrent copy jobs across as many LPARs as possible.

- Ensure that jobs and address spaces that will use concurrent copy are assigned a WLM service class with a high execution velocity. Never assign a discretionary goal to concurrent copy work.
- Ensure that volumes that will be processed will not be reserved by another system during the concurrent copy.

DFSMSHsm as data mover

When DFSMSHsm (IDCAMS) is used as the data mover, the control data sets are not validated during backup, and concurrent copy is not available. Thus you need to examine the number of records copied each day, because a broken VSAM index for a CDS will likely result in significant data loss in the backup copy. This loss can go unnoticed for a considerable period of time.

To specify DFSMSHsm as the datamover, add the following command to the ARCCMDxx member for processing unit 1: SETSYS CDSVERSIONBACKUP (DATAMOVER (HSM))

Using non-intrusive journal backup

There are two types of journal backup methods which should be considered: *quiesced* and *non-intrusive*.

During *quiesced* journal backup, all DFSMSHsm activity is quiesced while the journal and the control data sets are backed up.

During *non-intrusive* journal backup, all DFSMSHsm activity is allowed to continue while DFSMSHsm creates a backup of the preexisting journal records. After the initial backup of the journal completes, all DFSMSHsm activity is quiesced and DFSMSHsm completes the backup of the journal and backs up the control data sets using concurrent copy (this is a requirement of non-intrusive journal backup). The overall elapsed time for non-intrusive journal backup is approximately the same as quiesced journal backup.

Because all DFSMSHsm activity is quiesced during quiesced journal backup, user and production jobs that require recall or recovery resources wait until the entire CDS and journal backup is complete. Therefore, the impact of journal backup on DFSMSHsm availability and performance should be considered when planning a backup schedule. Non-intrusive journal backup can be used to reduce the impact on DFSMSHsm availability and performance.

Message ARC0750I indicates which method is in use and why non-intrusive journal backup was not performed. All requirements to use non-intrusive journal backup must be satisfied.

Enabling non-intrusive journal backup

To enable and use non-intrusive journal backup, the non-intrusive journal backup requirements must be satisfied. See [Requirements to use non-intrusive journal backup](#).

Disabling non-intrusive journal backup

There is no method of disabling non-intrusive journal backup. Non-intrusive journal backup will be used instead of quiesced journal backup as long as the non-intrusive journal backup requirements are satisfied.

Requirements to use non-intrusive journal backup

Requirements that must be satisfied to enable and use non-intrusive journal backup are:

- The SETSYS JOURNAL(RECOVERY) setting is in effect.
 - Add the SETSYS JOURNAL(RECOVERY) command to the ARCCMDxx parmlib member of SYS1.PARMLIB and restart DFSMSHsm.
 - Issue the SETSYS JOURNAL(RECOVERY) command after DFSMSHsm is started.
- The control data sets are configured to be backed up using concurrent copy.

Related reading

- For more information about using DFSMSdss as the data mover and configuring the use of concurrent copy, see [“DFSMSdss as data mover” on page 494](#).
- For more information about using the SETSYS command, see Chapter 65, [“SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#).

Specifying how many backup versions to keep

DFSMSHsm keeps the backup copies of the control data sets as versions, where each version is a set of CDS backup data sets and a journal backup data set. The version number of each backup data set is shown in the low-level qualifier of the data set names. Unless all backups fail, the CDS backup version number is incremented by one, and the new backup data sets are created and cataloged each time you back up the control data sets.

Note: As a result of errors during backup of an individual CDS, you may have one or more invalid backup data sets in a backup version. If you back up your control data sets to DASD and fail to attend to any CDS backup failures, you can eventually lose the ability to recover the control data sets.

Whenever a new CDS backup version is created, the oldest backup version is eliminated from the system (rolled off). If you back up to tape, the oldest backup version is uncataloged. If you back up to DASD, the oldest backup version is reused and renamed to the new version.

You must decide how many versions to keep on your system, and then tell DFSMSHsm that number by using the BACKUPCOPIES subparameter of the SETSYS CDSVERSIONBACKUP command. You should keep a minimum of three CDS backup versions. Specify that number to DFSMSHsm with the following command:

```
SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(3))
```

Specifying the names of the backup data sets

Use subparameters of the SETSYS CDSVERSIONBACKUP command to specify the data set names that are to be assigned to the backup data sets. You can use up to 35 characters in the data set name, including periods, but you must not end the name with a period. DFSMSHsm appends a qualifier to the data set name that you specify. The form of this qualifier depends on the data mover used to perform the backup:

- *DSy.Vnnnnnnn* — V specifies that IDCAMS was the data mover
- *DSy.Dnnnnnnn* — D specifies that DFSMSdss was the data mover
- *DSy.Xnnnnnnn* — X specifies that the backup failed, regardless of the data mover

Note: *nnnnnnn* is the version number. *y* is the volume for a multicluster CDS. *DSy* does not exist if the CDS is only one volume.

If you do not specify the data set names, DFSMSHsm generates names of *uid.BCDS.BACKUP*, *uid.MCDS.BACKUP*, *uid.OCDS.BACKUP*, and *uid.JRNL.BACKUP* for the data sets. DFSMSHsm substitutes the UID from the startup procedure for *uid*.

The following commands, used in the example system and added to the ARCCMDxx member in processing unit 1, demonstrate DFSMSHsm processing when the data set names are not specified:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(HSM) -  
  BACKUPCOPIES(3) -  
  BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -  
  MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -  
  OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -  
  JRNLBACKUPDSN(BHSM.JRNL.BACKUP))
```


Specifying the device type for backup of the control data sets

The control data sets and journal data set can be backed up to either DASD or tape. Both DASD and tape have certain advantages in relation to each other. In deciding whether to use DASD or tape for backing up your control data sets and journal data set, consider the following factors:

- **Mounting:** You do not have to mount DASD. For tape backup, you must mount a tape for each control data set and the journal.
- **Allocation:** With DASD, you do not have allocation problems that are caused by no available drives. With tape, the allocation request for a scratch tape fails if no tape drives are available.
- **SMS-managed:** With DASD, the backup copies of your control data sets can be SMS managed.
- **Preallocation:** With tape, you need not preallocate backup data sets.

Whether you back up the control data sets to DASD or to tape, DFSMSHsm issues messages ARC0742I and ARC0743I for each data set that is backed up. If the backups are in nonparallel mode, the messages are issued in pairs. However, if the backups are in parallel mode, all of the ARC0742I messages may be issued before the first ARC0743I message is issued. Parallel processing makes it possible to start all the backups before the first backup is completed.

If you decide to back up the control data sets and journal data set to DASD, you must preallocate the data sets. Also, allocate the different versions of the backup data sets on different volumes. This better ensures that you can recover your backup versions.

In the example system, backup is shown going to DASD. Therefore, the command that is added to the ARCCMDxx member in processing unit 1 is modified as follows:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -
  BACKUPCOPIES(3) -
  BACKUPDEVICECATEGORY(DASD) -
  BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
  MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
  OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
  JRNLBACKUPDSN(BHSM.JRNL.BACKUP))
```

Unlike backup of user data sets, backup of the control data sets must be to only one kind of I/O device. You cannot use a mixture of DASD and tape.

Specifying DASD for backup

If you specify BACKUPDEVICECATEGORY(DASD), parallel processing is always performed.

Because the backup of the control data sets to DASD is in parallel mode, all of the resources necessary for the backup are allocated before the control data sets and the journal are reserved. This means that DFSMSHsm cannot perform its other functions until either backup is complete or (if concurrent copy is used) concurrent copy sessions have been established and journal backup is complete.

Specifying tape for backup

Parallel backup CDS processing to tape allows the control data set and journal backups to be performed concurrently, thereby reducing the time required for the CDS backup function.

If you specify BACKUPDEVICECATEGORY(TAPE), your datamover is DFSMSHsm, and you want parallel backup processing, then you must specify PARALLEL, because NOPARALLEL is the default. When using DFSMSdss as the data mover and going to tape, parallel processing must be specified.

If CDS backup is invoked while the control data sets are accessed in RLS mode, DFSMSdss must be the datamover. If the backup is directed to tape, the PARALLEL parameter of the SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY) must be specified.

When backing up to tape in parallel mode, one tape device for each data set to be backed up is required. For instance, if all three control data sets and the journal are to be backed up, then four tape drives are required.

When the backup of the control data sets is in parallel mode, all of the resources necessary for the backup are allocated before the control data sets and the journal are reserved.

Protecting CDS backup tapes

The DFSMSHsm CDS backup tapes are supported for expiration protection only, as specified by the EXPIRATIONDATE or RETENTIONPERIOD subparameters of the SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(TAPE)) command. The SETSYS TAPESECURITY command does not affect the CDS backup tapes, and cannot be used to assign RACF or password protection as it can with other DFSMSHsm-owned tapes. However, the DFSMSHsm ARCCBEXT installation exit can be used to assign RACF protection to the CDS backup tapes, and the DFSMSHsm ARCTVEXT installation exit can be used to remove RACF protection from the tapes when you want to return them to scratch.

Related reading

For more information about DFSMSHsm installation exits, see [z/OS DFSMS Installation Exits](#).

Invoking the CDS backup exit (ARCCBEXT)

After CDS backup completes, the *control data set backup exit* (ARCCBEXT) is invoked, if it is active.

Related reading

For more information about the ARCCBEXT exit, see [z/OS DFSMS Installation Exits](#).

Backing up to multiple DASD data sets

If you decide to back up the control data sets and journal data set to DASD, you must first allocate and catalog a backup data set for each backup version of the control data sets. If the CDS is multicluster, additional backup version data sets are required. See [“Considerations when using multicluster control data sets”](#) on page 502.

The following example describes the process of backing up the control data sets and journal data set to multiple DASD backup data sets. The example assumes you have single-cluster control data sets.

1. The following command has been specified to indicate the number of preallocated data sets for each control data set and the journal data set:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -
  BACKUPCOPIES(3) -
  BACKUPDEVICECATEGORY(DASD) -
  MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
  BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
  OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
  JRNLBACKUPDSN(BHSM.JRNL.BACKUP))
```

BACKUPCOPIES(3) indicates the number of preallocated data sets for each backup data set.

2. Three backup data sets are allocated for each control data set and the journal data set backup copy. Assume the following 12 data sets are allocated:

Copy Number	Names of Allocated Backup Data Sets
1	BHSM.MCDS.BACKUP.V0000001 BHSM.BCDS.BACKUP.V0000001 BHSM.OCDS.BACKUP.V0000001 BHSM.JRNL.BACKUP.V0000001
2	BHSM.MCDS.BACKUP.V0000002 BHSM.BCDS.BACKUP.V0000002 BHSM.OCDS.BACKUP.V0000002 BHSM.JRNL.BACKUP.V0000002
3	BHSM.MCDS.BACKUP.V0000003 BHSM.BCDS.BACKUP.V0000003 BHSM.OCDS.BACKUP.V0000003 BHSM.JRNL.BACKUP.V0000003

Note:

- a. The final qualifier of each backup data set must start with V0000001 and increase by 1 for each required backup data set. Because you chose "DSS" as the data mover in the SETSYS CDSVERSIONBACKUP command, DFSMSHsm changes the final qualifier of its output data sets to begin with the letter D (D0000001). The renaming of the backup data set final qualifier happens automatically when you change data movers.
 - b. If the journal is allocated as a large format sequential data set and is larger than 64K tracks, when you preallocate the journal backup data set, you must allocate the journal backup data set as a large format sequential data set, which is large enough to contain a copy of the original large format sequential journal.
3. DFSMSHsm backs up the control data sets and journal data set by doing the following:
- a. It issues an exclusive enqueue against the control data set resource and, if it is in a multiple DFSMSHsm-host environment, reserves the volumes containing the journal data set and each control data set being used.

When using RLS in a multiple DFSMSHsm-host environment, DFSMSHsm does an exclusive enqueue rather than a reserve.

- b. It invokes DFSMSdss to logically dump each of the control data sets, using concurrent copy if available. DFSMSdss informs DFSMSHsm when a concurrent copy session has been established, or when it has finished the dump of a control data set.
- c. It copies the journal to the backup data set and nulls the journal data set. When the journal has been backed up and all the concurrent copy sessions have been established or all the control data sets have been backed up, DFSMSHsm releases the volume reserves (if using RLS there are no reserves), and dequeues the control data set resource.

When DFSMSHsm backs up the control data sets and journal data set, it determines the final qualifier of the backup data sets. If the version number in the final qualifier matches the version number in the final qualifier of an existing data set, DFSMSHsm uses the existing data set. In this example, DFSMSHsm finds an existing backup data set the first three times it backs up the control data sets, and changes the first letter of the last data set name qualifier from V to D (V0000001 becomes D0000001). The fourth time DFSMSHsm backs up the control data sets, it cannot find an existing version 4 backup data set. Therefore, DFSMSHsm backs up the control data sets to the backup data set with the final qualifier of D0000001.

- d. If DFSMSHsm backs up the control data sets to DASD volumes and needs to rename the backup data set, it does so. In this example, after DFSMSHsm successfully backs up these data sets for the fourth time, it renames the backup data set by changing the final qualifier of D0000001 to a final qualifier of D0000004. Even if the renaming fails, the backup copy is still valid.
- e. DFSMSHsm optionally calls the control data set backup user exit ARCCBEXT, allowing you to process the backup version after DFSMSHsm creates it.

Note: If you change the specification for the number of backup copies kept to *reduce* the number of copies, DFSMSHsm does not delete the excess older backup versions that were made before you reduced the number of copies.

If you change the specification for the number of backup copies kept to *increase* the number of copies, you must first allocate and catalog additional backup data sets. Start with a final qualifier that is one more than the last final qualifier of your backup data sets on DASD.

Related reading

For more information about allocating the MCDS, BCDS, OCDS, and journal data set, and their backup copies, see [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Backing up to multiple tape data sets

If you decide to back up the control data sets and journal data set to multiple tape data sets, the process is a little different. When you specify the following command, you do not have to allocate and catalog backup data sets:

```
SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(copies) -
  BACKUPDEVICECATEGORY(TAPE))
```

You can choose to specify a unit name, density, and retention period or expiration date.

The following example describes the process of backing up the control data sets and journal data set to multiple tape backup data sets.

1. The following command is specified:

```
SETSYS CDSVERSIONBACKUP (BACKUPCOPIES(3) -
  DATAMOVER(DSS) -
  BACKUPDEVICECATEGORY(TAPE))
```

BACKUPCOPIES(3) indicates the maximum number of cataloged backup data sets for each control data set and the journal data set.

2. DFSMSHsm backs up the control data sets and the journal data set. DFSMSHsm puts only one backup data set on a specific tape. Because you do not have to preallocate backup data sets when you use tape, DFSMSHsm dynamically allocates a new backup data set. The default backup data set name is the same for tape as for DASD. Therefore, if you have not specified the backup data set name with the SETSYS CDSVERSIONBACKUP command, DFSMSHsm uses the default backup data set name and the correct final qualifier for the new backup data set. In this case—the first time DFSMSHsm backs up the four data sets—the backup data sets have the following names:

```
HSM.MCDS.BACKUP.D0000001
HSM.BCDS.BACKUP.D0000001
HSM.OCDS.BACKUP.D0000001
HSM.JRNL.BACKUP.D0000001
```

The second time DFSMSHsm backs up the four data sets, it uses a final qualifier of D0000002.

```
HSM.MCDS.BACKUP.D0000002
HSM.BCDS.BACKUP.D0000002
HSM.OCDS.BACKUP.D0000002
HSM.JRNL.BACKUP.D0000002
```

3. The third time DFSMSHsm backs up the four data sets, it dynamically allocates a third set of backup data sets and uses a final qualifier of D0000003.

After DFSMSHsm successfully backs up the control data sets and the journal data set for the fourth time, it uncatalogs the set of backup data sets whose final qualifier is D0000001 and invokes installation exit ARCTVEXT for the rolled-off version of each successfully backed-up data set.

Note: If the backup is not successful for a particular data set, the oldest version of that data set is not rolled off, and DFSMSHsm invokes the tape installation exit, ARCTVEXT, that allows you to reuse the bad backup tapes.

4. DFSMSHsm optionally invokes the control data set backup user exit ARCCBEXT, allowing you to process the backup version after DFSMSHsm creates it.

Note:

1. The maximum number of tapes DFSMSHsm uses for one backup version of one control data set or the journal is 18.
2. If you change the specification for the number of backup copies kept to reduce the number of copies, DFSMSHsm does not delete or uncatalog the excess older backup versions that were made before you reduced the number of copies.

Considerations in a multiple DFSMSHsm host environment

The following are considerations in a multiple DFSMSHsm-host environment:

- All hosts sharing DFSMSHsm control data sets must specify the same options for the SETSYS CDSVERSIONBACKUP command, as the information is stored in the MCDS. The last DFSMSHsm host to issue the SETSYS CDSVERSIONBACKUP command controls how all the DFSMSHsm hosts back up the control data sets.

An alternative to having all DFSMSHsm hosts specifying the same options for the SETSYS CDSVERSIONBACKUP command is to have only one DFSMSHsm host (typically the host identified as the primary DFSMSHsm host) specifying the command.

- The NOCDSVERSIONBACKUP parameter of the SETSYS command is no longer valid. If SETSYS NOCDSVERSIONBACKUP is specified, message ARC0103I is issued, indicating an unsupported parameter.
- If backing up the control data sets to DASD, make the preallocated data sets accessible to any DFSMSHsm host doing the backups.
- If backing up to tape, some of your tape data sets won't be rolled off if they were created on one z/OS image, deleted on another z/OS image, and cataloged other than on a shared user catalog.
- A CDSQ=RLS DFSMSHsm environment will use XCF to communicate to all hosts in an HSMPLX that CDS Backup is starting and needs the ARCENQG/ARCCAT resource.

Considerations when using multicluster control data sets

A multicluster MCDS or BCDS can be defined when the CDS size increases beyond one volume. If you have defined a multicluster CDS, you must make the following changes to your CDS backup procedure:

- When backing up your control data sets to DASD, preallocate each volume within the multicluster CDS using a different naming convention. Each low-level qualifier (version number) must be preceded by .DSy, where y is a number from 1 to 4 and represents the number of volumes in the multicluster CDS. For instance, if you have split the MCDS into four volumes, split the BCDS into two volumes, and are keeping only one backup copy, the following data sets would be preallocated (assuming the choice of DFSMSdss for the data mover):

```
HSM.MCDS.BACKUP.DS1.D0000001
HSM.MCDS.BACKUP.DS2.D0000001
HSM.MCDS.BACKUP.DS3.D0000001
HSM.MCDS.BACKUP.DS4.D0000001
HSM.BCDS.BACKUP.DS1.D0000001
HSM.BCDS.BACKUP.DS2.D0000001
HSM.OCDS.BACKUP.D0000001
HSM.JRNL.BACKUP.D0000001
```

- The interface to the ARCCBEXT installation exit has been extended to include the names of the backup data sets for the multicluster CDS.
- When you back up your control data sets to tape, the new naming convention (.DSy) applies to the created tape data set names.

Note: The CDS backup tapes created under the old naming convention are not rolled off automatically. Message ARC0747E is issued to indicate that roll-off processing cannot be performed.

- Messages ARC0742I, ARC0743I, and ARC0744E indicate which volume in the multicluster CDS was being processed.
- When you back up your control data sets to tape in parallel, a tape drive is needed for each cluster in the multicluster CDS. For instance, if you split the MCDS into four volumes, split the BCDS into two volumes, have defined an OCDS on one volume, and are journaling, eight tape drives are required to back up the control data sets and the journal.

Related reading

- For more information about how to implement a multicluster CDS, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about the ARCCBEXT installation exit, see [z/OS DFSMS Installation Exits](#).

Nulling the journal

If the journal data set fails to back up during the CDS version backup function, the journal will not be nulled. Conversely, the journal is always nulled if the backup of the journal data set is successful, regardless of the success or failure of the backup of the control data sets.

You can null the journal by using the following command: `BACKVOL CDS(NULLJOURNALONLY)`

`NULLJOURNALONLY` is an optional subparameter telling the CDS version backup function to only null the journal. No backup copies of the control data sets or journal are made.

Changing the names of the backup data sets

You can change the naming conventions for the backup data sets by performing the following steps:

- Issue the `SETSYS CDSVERSIONBACKUP` command using the subparameter for the data set names assigned to the backup data sets. Specify the new naming convention used to back up the control data sets.

Rule: You must change the naming convention for the backup data sets in all procedures that DFSMSHsm uses. For example, update the `ARCCMDxx` member of the DFSMSHsm startup procedure if you have specified the `SETSYS CDSVERSIONBACKUP` command.

- Rename any current backup versions of the control data sets to the new naming convention if the backup data sets are on DASD. Also, you must rename any “unused” backup versions.
- Uncatalog the old backup data sets so the tape management function can reuse the tape if the backup data sets are on tape. During backup of the control data set, you might receive message `ARC0747E`, which indicates DFSMSHsm could not uncatalog the oldest version using the new backup names. The tape backup copies are successfully made.

Changing backup device from tape to DASD

If you started your backup of the control data sets and the journal data set to tape and want to change to DASD, do the following:

1. Uncatalog the backup versions that reside on tape. If this is not done, attempts to back up the control data sets to DASD will fail.

The expiration date specified by the `EXPIRATIONDATE` or `RETENTIONPERIOD` parameter is still in effect on your backup tapes. Your tape management system returns the tapes to scratch according to the expirations date. Note that some expiration dates, such as 99365 (which prevents the tape from ever expiring), may require manual deletion through your tape management system.

2. Specify the following command:

```
SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(DASD))
```

You can also change the number of backup copies. However, this example uses the same number as in the tape example; that is, three backup copies.

3. Allocate your backup data sets on DASD. The sample job `ALLOCBK1`, which is located in the DFSMSHsm starter set, allocates and catalogs the control data set backup version data sets on DASD volumes.

The final qualifier must again increase by 1 for each set of backup data sets. However, do not start with a final qualifier of D0000001. Instead, start with a final qualifier that is 1 more than the last final qualifier of your backup data sets on tape.

Issue the following command to determine the last final qualifier for the backup data sets:

```
QUERY CDSVERSIONBACKUP
```

In this example, the last final qualifier is D0000050.

When you allocate your backup data sets, you decide to use the default backup data set names. Because you are backing up all three control data sets and the journal data set, you must allocate 12 backup data sets:

Data Set	Names of Allocated Backup Data Sets
MCDS	HSM.MCDS.BACKUP.D0000051 HSM.MCDS.BACKUP.D0000052 HSM.MCDS.BACKUP.D0000053
BCDS	HSM.BCDS.BACKUP.D0000051 HSM.BCDS.BACKUP.D0000052 HSM.BCDS.BACKUP.D0000053
OCDS	HSM.OCDS.BACKUP.D0000051 HSM.OCDS.BACKUP.D0000052 HSM.OCDS.BACKUP.D0000053
Journal	HSM.JRNL.BACKUP.D0000051 HSM.JRNL.BACKUP.D0000052 HSM.JRNL.BACKUP.D0000053

DFSMShsm now backs up the control data sets and journal data set to backup data sets on DASD volumes.

Related reading

For more information about the sample job ,ALLOCBK1, and the DFSMShsm starter set, see [z/OS DFSMShsm Implementation and Customization Guide](#).

Changing backup device from DASD to tape

If you started your backup of the control data sets and journal data set to DASD and want to change to tape, specify the following command:

```
SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(TAPE))
```

If you are using DFSMSdss as the data mover, specify:

```
SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(TAPE(PARALLEL)))
```

You may change the number of backup copies. If you do not change the number of backup copies, the backup copies that are currently cataloged on DASD are uncataloged by DFSMShsm.

Note: DFSMShsm does not delete the DASD backup copies. The only function performed by DFSMShsm is to uncatalog these data sets. If you feel that you do not need the backup copies on DASD, you may scratch these data sets from your system after the appropriate number of backup copies have been made on a tape device media.

If you have already uncataloged the DASD backup copies, or if you change the number of backup copies, the attempt to uncatalog the DASD backup copies may fail during the backup of the control data sets and journal data set. An ARC0747E message is issued for each data set that could not be uncataloged, and the ARC0741I message at the completion of the backup of the control data sets and journal data set indicates that the backup was UNSUCCESSFUL. However, this unsuccessful status indicates only that backup versions were not successfully uncataloged. The tape backup copies will have been successfully

made, and the final qualifier for these backup copies will have been successfully saved to be used for the next backup of the control data sets and journal data set. This scenario requires no action on your part.

Using the QUERY command to get CDS information

Issue the DFSMSHsm QUERY CDSVERSIONBACKUP or the QUERY SETSYS commands to determine whether DFSMSHsm or DFSMSdss is being used to back up the control data sets.

Issue the QUERY CONTROLDATASETS command to display the CDS serialization technique in use, and the key boundaries being used for non-key-range multicluster control data sets.

Backing up the control data sets manually

Occasionally, it may become necessary for you to manually back up the control data sets. This manual backup could be needed if CDS version backup fails to get a good copy of the control data sets, or if you receive a message that the journal is full. DFSMSHsm provides parameters for the BACKVOL command that allow you to back up all the control data sets manually:

```
BACKVOL CDS(DATAMOVER(HSM | DSS)
  BACKUPDEVICECATEGORY(DASD | TAPE(NOPARALLEL | PARALLEL)))
```

where:

DATAMOVER(HSM | DSS)

specifies which CDS backup data mover should be used when backing up the control data sets.

BACKUPDEVICECATEGORY(TAPE | DASD)

specifies the device on which the backup copies are recorded.

TAPE(NOPARALLEL | PARALLEL)

specifies whether one or more tape devices are allocated. If DATAMOVER(DSS) is specified, TAPE(NOPARALLEL) cannot be specified.

Note:

1. If you do not specify DATAMOVER with the BACKVOL CDS command, the last DATAMOVER specified by SETSYS CDSVERSIONBACKUP is the default.
2. If you do not specify BACKUPDEVICECATEGORY with the BACKVOL CDS command, the last BACKUPDEVICECATEGORY specified by SETSYS CDSVERSIONBACKUP is the default.
3. If you specify BACKVOL CDS DATAMOVER(DSS) without BACKUPDEVICECATEGORY, and the DFSMSHsm CDS version backup environment indicates TAPE(NOPARALLEL), the BACKVOL command terminates with an error message indicating that the CDSVERSIONBACKUP options of DATAMOVER(DSS) and TAPE(NOPARALLEL) are in conflict.
4. Use caution if you change device types with the BACKUPDEVICECATEGORY parameter. See the topics on [“Changing backup device from tape to DASD”](#) on page 503 and [“Changing backup device from DASD to tape”](#) on page 504.
5. Using DATAMOVER(DSS) provides advantages of validation during backup as well as (with the concurrent copy function) reduced serialization of the control data sets. DATAMOVER(HSM) is retained as an option for purposes of coexistence with previous releases.
6. In a single DFSMSHsm host environment, DFSMSHsm obtains an exclusive enqueue on the control data sets. In a multiple DFSMSHsm host environment, DFSMSHsm reserves the volumes, or if using RLS, does an exclusive enqueue on a SYSTEMS wide global resource which will prevent other HSMs in an HSMplex from accessing the control data sets.

Possible failures during CDS and journal backup

If a CDS or journal backup fails, DFSMSHsm documents the failure with message ARC0744E. This topic explains how to proceed after the most likely type of error, and more general suggestions for other error conditions.

Growth of your control data sets

Over time, the number of data sets migrated and backed up, the number of tape volumes owned by DFSMSHsm, and so on, typically increase, causing your control data sets to grow in size. A control data set may grow such that the space allocated to its DASD backup copy is no longer enough, producing return code 12, 13, 18, 47, or 48 in message ARC0744E during an attempted backup.

Assuming from analyzing messages from DFSMSdss or Access Method Services that a backup copy is full, proceed as follows:

1. Issue the QUERY CDSVERSIONBACKUP command. The resulting message ARC0376I, for example,

ARC0376I

```
BACKUPCOPIES=3,
BACKUPDEVICECATEGORY=DASD,
LATESTFINALQUALIFIER=V0000170,
DATAMOVER=HSM
```

gives you the latest version qualifier, and how many backup copies need to be reallocated.

2. Examine the message for the control data set that is identified in the ARC0744E message and determine which backup copies (MCDS, BCDS, OCDS, or JRNL) must be made larger. Assume that the backup copies of the BCDS must be reallocated.
3. Rename the backup data sets *uid.BCDS.BACKUP.V0000168* through *uid.BCDS.BACKUP.V0000170* so they can be kept until there are successful backups.
4. Compare the (used) size of the current BCDS to its size when the backup data sets for the BCDS were last allocated. This allows you to get a rough idea of how much larger the backup data sets must be.
5. Use the ratio from step “4” on page 506, and reallocate backup data sets *uid.BCDS.BACKUP.V0000168* through *uid.BCDS.BACKUP.V0000170* to their new larger sizes.
6. Issue the commands

```
RELEASE BACKUP
BACKVOL CDS
```

If the failed backup was for the journal, DFSMSHsm will have inhibited journaling. A successful BACKVOL CDS command should back up the journal, then null the online journal.

7. Issue commands to RELEASE the other DFSMSHsm functions that you are using once the BACKVOL command is successful.
8. Delete the backup data sets that were renamed in step “3” on page 506 when DFSMSHsm makes three successful backups. A successful backup generates the following message as part of the output from a QUERY CDSV:

ARC0376I

...

```
LATESTFINALQUALIFIER=V0000173,
```

Other failures

By considering the return code in message ARC0744E, you can determine which type of failure occurred:

- If there is an indication of an error in the structure or contents of the control data set itself, perform the following steps:
 1. Take the action to correct the error as indicated by the error message number and the associated return code. This information is contained in the message documentation.
 2. You might need to recover the CDS to a previous level.
- If there is an indication of an error in the system environment outside of the control data set (for example, an output tape could not be allocated), perform the following steps:

1. Stop DFSMSHsm if necessary (if, for example, the error was fragmented storage such that IDCAMS could not be loaded).
 2. Correct the environmental condition, if necessary.
 3. Restart DFSMSHsm, if you stopped it.
 4. Issue the BACKVOL CDS command to make up for the backup that failed.
- If the non-intrusive journal backup method was intended, but was not used, perform the following steps:
 1. Review the output of message ARC0750I, which is issued at the start of journal backup. This message will indicate which journal backup method was used and why non-intrusive journal backup was not used.
 2. Verify all requirements to use non-intrusive journal backup are satisfied. For a list of requirements, see [“Using non-intrusive journal backup”](#) on page 496.

When the conditions preventing non-intrusive journal backup are corrected, future CDS backup processing will use non-intrusive journal backup.

- If there is an indication of a probable program error, gather the necessary data for contacting the IBM Support Center.

Related reading:

- For more information about the action to take regarding error messages and return codes, see [z/OS MVS System Messages, Vol 2 \(ARC-ASA\)](#).
- For more information about how to recover the CDS to a previous level, see [“Recovering control data sets using the enhanced CDS recovery function”](#) on page 508).

Reasons for recovering the control data sets

Occasionally a control data set becomes unusable by DFSMSHsm. For example, a hardware failure or a software logical failure may occur after which DFSMSHsm is unable to use a control data set.

When the control data sets are backed up, they constitute a snapshot of the conditions in the system at that time. After the control data sets are backed up, as DFSMSHsm continues to process data sets and volumes, the control data sets are continually changing. To make it possible for you to return the control data sets to their condition just before the error, DFSMSHsm makes a journal record of each transaction that occurs in the control data sets. If a control data set becomes lost or damaged, you can return it to its correct condition by recovering a backup version of the control data set and applying the appropriate journal.

The CDS recovery process has variations based on which data mover has been used to back up the control data sets. The low-level (final) qualifier of the name of the control data set backup data set indicates which data mover has been used to back it up. A low-level qualifier of Vnnnnnnn indicates that DFSMSHsm backed up the data set. A low-level qualifier of Dnnnnnnn indicates that DFSMSdss backed up the data set, and a low-level qualifier of Xnnnnnnn indicates that CDS version backup of the data set failed, regardless of whether DFSMSHsm or DFSMSdss was doing the backup.

I/O errors

When an I/O error occurs on the control data sets, the following situations can exist:

- If the I/O error is caused by an equipment check on the real DASD containing the affected data set, recover the data set to a different volume. Call your service representative to repair the drive on which the equipment check has occurred.
- If the I/O error is caused by a data check on the real DASD containing any control data set, you can use the enhanced CDS recovery function to recover the backup version of the affected data set to a different volume. You can use Device Support Facilities (a program to perform operations on disk volumes) to analyze, inspect, and assign alternate tracks and then recover the affected control data set to the same

volume. Or, you can rename the affected control data set and define a new control data set on the same volume.

Recovering control data sets using the enhanced CDS recovery function

The recommended procedure for recovery is called the "enhanced CDS recovery function", which allows you to recover a control data set to the point of failure (to the point where the hardware failure occurred or to the point where the software failure was recognized).

The enhanced CDS recovery function consists of the following two utility programs:

ARCBJRNL

The journal copy utility that backs up the journal—using as input either the active journal or the active journal plus multiple journal backup copies—and prepares the resultant backed up copy that is sorted and used as input to the import utility.

Rule: Because there may still be residual data in the journal, you must run the sort job regardless of how many records that ARCBJRNL reports were processed.

ARCIMPRT

The import utility that uses a sorted version of the backed up and sorted journal or journals from the journal copy utility and an exported control data set to recover the control data set to the point of failure.

Requirements for using the enhanced CDS recovery function

To successfully use the enhanced CDS recovery function, you must be sure that:

- You have sufficient DASD to contain the rebuilt control data set and the backed up journal.
- DFSMSHsm is not active on any z/OS image in the complex while the control data set is being recovered.
- You have specified SETSYS JOURNAL for normal DFSMSHsm operation, because an up-to-date journal is necessary for the enhanced CDS recovery function.
- You run the ARCIMPRT utility separately for each specific control data set to be recovered because the parameter passed to ARCIMPRT is for a specific control data set to be recovered. Also, the EXPORT DD statement must be changed to the particular exported control data set being used.

If you are recovering a multicluster CDS, you need recover only the specific volume which failed; you do not need to recover all the volumes within the multicluster CDS.

In other words, if a migration control data set is to be recovered, then the parameter passed to ARCIMPRT must be MCDS or MIGRATIONCONTROLDATASET, and the EXPORT DD statement *dsname* must refer to the portable-exported-migration CDS (your backup copy of the control data set).

Implementing enhanced CDS recovery using JCL

This topic describes the steps necessary to recover a CDS. You can recover a CDS either from the most recent backup copy or from an older backup copy. An older backup copy is needed if your most recent backup copy is unusable. Both types of recoveries are essentially the same except for Step 1 (Create the Backup Journal).

DFSMSHsm should be down when processing this procedure.

If multiple journal backup copies are needed to recover the journal, and therefore recover a control data set, the following procedure should be used, because it minimizes the time DFSMSHsm is down:

1. Use the ARCBJRNL utility (with multiple backup copies as input) and the ARCIMPRT utility to recover the control data set to a point just before what the active journal contains.
2. Bring down DFSMSHsm.

3. Use the ARCBJRNL utility (with only the active journal as input) and the ARCIMPRT utility to finish recovering the control data set. When you run this step you need to make an exported copy of the MCDS created by ARCIMPRT in step 1 as the input to ARCIMPRT in this step.
4. Bring DFSMSHsm up.

Steps 1, 2, and 4 can be run separately as shown or can be combined into one job.

The following sample JCL shows the steps required to recover a control data set:

1. Create the backup journal.

The backup journal can be created by using either of the two following JCL examples.

This example uses only the active journal as input and is used if recovering from the most recent backup copy of your CDS. If the most recent backup copy is a failed copy (last qualifier is Xnnnnnnn), follow the procedures which use an active journal and one or more backup journals.

```
//ARCBJRNL JOB ,MSGLEVEL=(1,1),MSGCLASS=A,REGION=1024K
//* -----
//* CREATE DATA SET CONTAINING RECORDS FROM THE
//* ACTIVE JOURNAL
//* -----
//STEP01 EXEC PGM=ARCBJRNL
//STEPLIB DD DSN=location where program ARCBJRNL resides,
//          DISP=SHR
//SYSPRINT DD SYSOUT=*
//JRNLO DD DSN=online journal data set name,
//        DISP=OLD
//JRNLBK DD DSN=backup journal name,
//          UNIT=unit name,
//          VOL=SER=volume where backup journal will reside,
//          SPACE=adequate space to hold journal,
//          DISP=(,CATLG,DELETE)
```

Note:

- a. You can indicate what input—the active journal, journal backup copies, or both—is used during ARCBJRNL utility processing by using the PARM keyword on the EXEC statement. The keywords and their meanings are:
 - JRNLOONLY — Only the active journal is used as input
 - JRNLBKBACKUP — Only journal backup data sets are used as input
 - JRNLBKBACKUPS — Only journal backup data sets are used as input
 - BOTH — Both the active journal and journal backup data sets are used as input
- b. If PARM is not specified, only the active journal is used as input to ARCBJRNL processing.
- c. When using multiple journal backup data sets for input to ARCBJRNL processing, list these data sets in ascending version order in the ARCBJRNL BKUPJRNL DD DSN= statements.
- d. The following DD statements are required:
 - i) JRNLO DD statement pointing to the current online journal being used by DFSMSHsm.
 - ii) BKUPJRNL DD statement pointing to the journal backup copies. If recovering using multiple journal backup copies you must concatenate them in ascending order. In addition if the journal backups were written to tape you must ensure you have enough tape drives available for the number of journal backup copies you will be concatenating as DFSMSHsm will allocate all of these data sets at the same time.
 - iii) JRNLBK dd statement pointing to the merged journal dataset created by ARCBJRNL which will be used as input to step 2. The following example uses the active journal and three existing journal backup copies as input. This example is used if recovering from an older backup copy of your CDS. In this case, the backup CDS is from 4 backups ago.

The following example uses the active journal and three existing journal backup copies as input. This example is used if recovering from an older backup copy of your CDS. In this case, the backup CDS is from 4 backups ago.


```
//ARCBJRNL JOB ,MSGLEVEL=(1,1),MSGCLASS=A,REGION=1024K
//* -----
//* CREATE DATA SET CONTAINING JOURNAL FROM JOURNAL
//* BACKUP COPY DATA SETS AND THE ACTIVE JOURNAL
//* -----
//STEP01 EXEC PGM=ARCBJRNL,PARM='BOTH'
//STEPLIB DD DSN=location where program ARCBJRNL resides,
// DISP=SHR
//SYSPRINT DD SYSOUT=*
//JRNLO DD DSN=online journal data set name,
// DISP=SHR
//BKUPJRNL DD DSN=backup copy name and version,DISP=SHR
// DD DSN=backup copy name and version,DISP=SHR
// DD DSN=backup copy name and version,DISP=SHR
//JRNLBK DD DSN=backup journal name,
// UNIT=unit name,
// VOL=SER=volume where backup journal will reside,
// SPACE=adequate space to hold journal,
// DISP=(,CATLG,DELETE)
```

2. Sort the journal backup file.

After you have created the journal backup file using the JCL in step 1, sort the file by KEY using the EQUALS parameter, as in the following JCL.

Note: The following JCL assumes you have a SORTD cataloged procedure. The SORTD cataloged procedure can be created by running the DFSORT SORTPROC job that DFSORT distributes in SYS1.SAMPLIB. If you do not have a SORTD procedure, modify the EXEC card in the JCL to identify the sort program you want to use. Ensure that the program is logically equivalent to DFSORT.

```
//SORT JOB
//* -----
//* SORT THE BACKED UP JOURNAL
//* -----
//STEP02 EXEC SORTD
//SORTLIB DD .....
//SYSOUT DD SYSOUT=*
//SORTWK01 DD .....
//SORTWK02 DD .....
//SORTWK03 DD .....
//SORTWKnn DD .....
//SORTIN DD DSN=backup journal name from STEP 1,
// DISP=SHR
//SORTOUT DD DSN=sorted backup journal name,
// UNIT=unit name,
// VOL=SER=volume where sorted backed up journal will reside,
// SPACE=adequate space to hold sorted backed up journal,
// DCB=(RECFM=VBS,BLKSIZE=0,LRECL=6560),
// DISP=(,CATLG,DELETE)
//SYSIN DD *
SORT FIELDS=(17,44,CH,A)
OPTION EQUALS
/*
```

3. The next step depends on the data mover used to create the backup copy. If the backup copy was made by DFSMSHsm, that backup copy is already in the form of a portable exported data set. If this is the case, step 3 is not required.

If the backup copy has been made by DFSMSdss:

- a. Restore the backup version (either the most recent backup version or the last good backup version) using the following example JCL:

```
//JOB1 JOB accounting information,REGION=nnnK
//STEP1 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//TAPE DD DSN=uid.CDS.BACKUP.D0000106,DISP=(OLD,KEEP)
//DASD DD UNIT=3390,VOL=SER=222222,DISP=OLD
//SYSIN DD *
RESTORE DATASET(-
INCLUDE(uid.dsname))-
INDDNAME(TAPE)-
OUTDDNAME(DASD)-
REPLACE
/*
```

- b. Export the data set restored in step “3.a” on page 510 using the following example JCL:


```
//EXPORT JOB accounting information,REGION=nnnnK
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//OUTDD1 DD DISP=(,CATLG,DELETE),BUFNO=26,BLKSIZE=28332,
//        SPACE=(CYL,(200,20),RLSE),
//        UNIT=3390,VOL=SER=222222,
//        DSN=temp_dsname
//SYSIN DD *
EXPORT dsname -
OUTFILE(OUTDD1) -
TEMPORARY
/*
```

4. Import and merge the control data set.

After you have sorted the journal backup file by doing step 2 and you have an exported copy of the subject CDS, run ARCIMPRT to create a control data set by using the following JCL.

Note:

- If you need to increase the size of the control data set, issue the DEFINE command before the IMPORT command.
- Any valid IDCAMS command may be specified in the SYSIN stream. These commands are passed directly to IDCAMS (program ARCIMPRT does not modify these commands). Therefore, the DELETE command is used to delete the control data set before the data set is imported.
- The IMPORT command must use the INFILE parameter; using the INDATASET parameter will result in an error return.

The following sample shows how to re-create a migration control data set. To re-create a backup control data set or an offline control data set, make changes to the "PARM=" in the EXEC statement and to the EXPORT DD statement *dsname* that reflects which control data set exported copy to use. Also make changes to the OUTDATASET name in the IDCAMS control statement that reflects which control data set is to be re-created.

```
//ARCIMPRT JOB
/* -----
/* RE-CREATE THE MCDS
/* Note the parameter in the following execute statement
/* specifying which CDS to recover.
/* Also, any valid IDCAMS command may be specified in the
/* SYSIN stream below.
/* -----
//STEP03 EXEC PGM=ARCIMPRT,PARM='MCDS'
//STEPLIB DD DSN=where ARCIMPRT resides,
//        DISP=SHR
//SYSPRINT DD SYSOUT=*
//AMSDUMP DD SYSOUT=*
//JOURNAL DD DSN=name of sorted backup journal from STEP 2,
//        DISP=SHR
//EXPORT DD DSN=name of portable exported MCDS from STEP 3,
//        DISP=SHR
//SYSIN DD *
DELETE (MCDS cluster name) CLUSTER PURGE
IMPORT INFILE(EXPORT) OUTDATASET(MCDS cluster name)
/*
```

- Restart DFSMSHsm from the processing unit from which you issued ARCIMPRT.
- Verify that the control data sets are correct. See [“Determining if control data sets are accurate”](#) on page 517.
- Fix any errors in the control data sets. See [“Repairing errors in the control data sets”](#) on page 517.
- If you are recovering the MCDS, issue a QUERY CDSVERSIONBACKUP to ensure that the latest final qualifier is correct. You can determine the final qualifier of the latest backup copy from your catalog using the LISTCAT LEVEL command, as shown in the following example:

```
LISTCAT LEVEL(uid)
```

If these qualifiers do not match, use the FIXCDS patch command to correct the field at offset X'B1' in the MHCR record to reflect the latest version in the catalog, as shown in the following example:

```
FIXCDS S MHCR PATCH(X'B1' 0000007)
```


9. Use the CONTROLDATASETS parameter of the BACKVOL command to get a current backup copy of the control data sets and journal data set.
10. Restart DFSMSHsm in the other processing units.

ARCBJRNL and ARCIMPRT return codes

Table 49 on page 512 and Table 50 on page 512 describe the return codes that are returned by the ARCBJRNL and ARCIMPRT utilities. These codes are returned in general register 15 and can be referred to in your JCL.

Table 49. ARCBJRNL Return Codes

Return code	Meaning
0	Normal return.
4	No journal ARCCR control record (MSGARC0461I).
8	Only journal ARCCR control record (MSGARC0461I).
28	Get macro I/O error.
32	Put macro I/O error.
40	Online journal OPEN error.
42	The online journal data set has a format 8 DSCB.
44	Backup journal OPEN error.
46	The journal backup input data set has a format 8 DSCB.
48	EOV or abend error.
50	I/O error occurred on journal backup input data set.
52	OBTAIN macro failed for the journal backup input data set.
54	Open error occurred on journal backup input data set.
56	RDJFCB macro failed for the journal backup input data set.
58	ABEND error occurred processing journal backup input data set.
64	JRNLO DD DUMMY record was found for the online journal data set.
68	RDJFCB macro failed.
72	OBTAIN macro failed.
80	Journal control record and data set type are inconsistent.
84	Insufficient level of DFSMSdfp product is installed.
99	Input parameter error (invalid value specified on PARM keyword).

Table 50. ARCIMPRT Return Codes

Return code	Meaning
0	Normal return.
4	IDCAMS message: minor problems but function completed.
8	IDCAMS message: function completed but some processing bypassed.
12	IDCAMS message: entire function could not be performed.
16	IDCAMS message: severe error or problem encountered.

Table 50. ARCIMPRT Return Codes (continued)

Return code	Meaning
18	Invalid backup journal used as input to ARCIMPRT (MSGARC0463I).
20	Invalid or no parameter passed to ARCIMPRT (MSGARC0462I).
22	Backup journal file open error.
23	Backup journal file has a format 8 DSCB.
24	Backup journal file I/O error.
25	Export file has a format 8 DSCB.
26	Export file open error.
27	No CDS records found in exported CDS.
28	Export file I/O error.
30	Sysprint file open error.
32	Sysprint file I/O error.
34	Did not read a header record preceding the CDS backup copy (this can occur if the IMPORT command uses the INDATASET parameter instead of INFILE).
68	RDJFCB macro failed.
72	OBTAIN macro failed.

Recovering multicluster non-key range control data sets

ARCIMPRT has been enhanced to recover multicluster non-key-range data sets. It uses the key boundaries to determine which journal records need to be applied to the cluster being recovered.

If you reorganize your control data sets, you should immediately back them up. If you do not, and you attempt to perform a CDS recovery using a CDS backup version created before the CDS was reorganized, the recovery fails with message ARC0466I. Then the following steps must be taken to restore your multicluster non-key-range CDS:

1. The cluster being recovered must be returned to the state in which it existed after the reorganization. For example, if the control data sets were split from two to three, then the job or jobs used to perform the split should be rerun. Only the cluster or clusters being recovered should be returned to the state they were in following the reorganization.
2. Reissue the ARCIMPRT job using the new FORCE parameter. The FORCE parameter causes ARCIMPRT to use the latest key boundaries in use, instead of those that existed at the time of the CDS version backup. To use the FORCE parameter on the MCDS, you would issue:

```
PARM= ' MCDS , FORCE '
```

After the recovery, message ARC0467I is issued identifying the cluster that was recovered.

Control data set recovery procedure using CDSVERSIONBACKUP with UPDATEC

For performance reasons, you should use the enhanced CDS recovery function instead of the UPDATEC command to recover your control data sets. However, if you want to use UPDATEC, use the following procedure:

1. Use the CDSVERSIONBACKUP parameter of the QUERY command to find out the initial characters and the final qualifier of the backup data set name. Save the QUERY information for use in steps [“5” on page 514](#) and [“11” on page 515](#). You will receive the following messages from the query:

ARC0375I

```
CDSVERSIONBACKUP,

MCDSBACKUPDSN=uid.MCDS.BACKUP,

BCDSBACKUPDSN=uid.BCDS.BACKUP,

OCDSBACKUPDSN=uid.OCDS.BACKUP,

JRNLBACKUPDSN=uid.JRNL.BACKUP
```

ARC0376I

```
BACKUPCOPIES=3,

BACKUPDEVICECATEGORY=DASD,

LATESTFINALQUALIFIER=V0000007,

DATAMOVER=HSM
```

Recommendation: If DFSMSHsm is not active, you should issue the following IDCAMS LISTCAT command to determine the latest version: LISTCAT LEVEL(*uid*)

Also note whether DFSMSHsm or DFSMSdss was the data mover (by looking at the prefix letter of the low-level qualifier).

2. Stop DFSMSHsm in all processing units.
3. Use the Access Method Services DELETE command to delete the control data set that has the error. The following example shows the DELETE command using one of several parameters that are available.

```
DELETE CLUSTER(uid.MCDS)
```

4. Re-create the control data set that was deleted in step “3” on page 514. When you re-create the control data set, increase the space allocation for that control data set if more space is required. For more information, refer to the *z/OS DFSMSHsm Implementation and Customization Guide*.
5. The next step depends on the data mover used to create the backup version.

- If the data mover was DFSMSdss, restore the latest backup version into the re-created data set using the following JCL example:

```
//JOB1 JOB accounting information,REGION=nnnK
//STEP1 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//TAPE DD DSN=uid.MCDS.BACKUP.D0000106,DISP=(OLD,KEEP)
//DASD DD UNIT=3390,VOL=SER=222222,DISP=OLD
//SYSIN DD *
        RESTORE DATASET(-
        INCLUDE(uid.dsname))-
        INDDNAME(TAPE)-
        OUTDDNAME(DASD)-
        REPLACE
/*
```

- If the data mover was DFSMSHsm, use the Access Method Services IMPORT command to recover the latest backup version of the affected control data set by using the following JCL example:


```
//IMPORT JOB , 'USERID' MSGCLASS=A,REGION=768K
//STEP1 EXEC PGM=IDCAMS
//DEST DD DSN=uid.MCDS,DISP=OLD
//SOURCE DD DSN=uid.MCDS,DISP=OLD,AMP=AMORG
//DASD DD DSN=uid.MCDS.BACKUP.V0000106,
// DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
IMPORT-
INFILE(DASD)-
OUTFILE(SOURCE)-
INTOEMPTY-
PURGE
/*
```

6. Restart DFSMSHsm in emergency mode, EMERG=YES, in the processing unit from which you will issue the UPDATEC command.

Recommendation: On the restart, ensure that your ARCCMDxx parmlib member is set correctly with either of the following specification:

- SETSYS EMERG is specified.
- Neither SETSYS EMERG nor SETSYS NOEMERG is specified.

If you follow either of the above recommendations, then the EMERG=YES specification will not be overridden by your ARCCMDxx member.

7. Use the UPDATEC command to combine the entries from the journal data set with the entries from the recovered control data set. (See [“Updating recovered control data sets using UPDATEC”](#) on page 516.) When you use this command, the backup version number of the journal data set is one greater than that of the backup version control data set used in step [“5”](#) on page 514. You must concatenate the journal backup data set name that you received in the ARC0375I message in step [“1”](#) on page 513 with the version number, and use that as the journal parameter of the UPDATEC command.

If you need to use multiple backup versions of the journal data set to recover a control data set, be sure you apply the backup versions in the correct order (oldest to newest).

8. Stop and restart DFSMSHsm in emergency mode, EMERG=YES, from the processing unit from which you issued the UPDATEC command.

Recommendation: On the restart, ensure that your ARCCMDxx parmlib member is set correctly with either of the following specification:

- SETSYS EMERG is specified.
- Neither SETSYS EMERG nor SETSYS NOEMERG is specified.

If you follow either of the above recommendations, then the EMERG=YES specification will not be overridden by your ARCCMDxx member.

9. Verify that the control data sets are correct. (See [“Determining if control data sets are accurate”](#) on page 517.)
10. Fix any errors in the control data sets. (See [“Repairing errors in the control data sets”](#) on page 517.)
11. If you are recovering the MCDS, issue a QUERY CDSVERSIONBACKUP to ensure that the latest final qualifier is the same as that shown in step 1. If you do not have the output from the QUERY command in step 1, you can determine the final qualifier of the latest backup copy from your catalog using the LISTCAT LEVEL command, as shown in the following example:

```
LISTCAT LEVEL(uid)
```

If these qualifiers do not match, use the FIXCDS patch command to correct the field at offset X'B1' in the MHCR record to reflect the latest version in the catalog, as shown in the following example:

```
FIXCDS S MHCR PATCH(X'B1' 0000007)
```

12. Use the CONTROLDATASETS parameter of the BACKVOL command to get a current backup copy of the control data sets and journal data set.

13. If the backup of the control data sets was successful, you can issue a SETSYS NOEMERG command, and release DFSMSHsm's automatic functions with a RELEASE ALL command.
14. Restart DFSMSHsm in the other processing units.

Updating recovered control data sets using UPDATEC

For performance reasons, you should use the enhanced CDS recovery function instead of the UPDATEC command to recover your control data sets. If you do use UPDATEC, the forms of the command are:

UPDATEC BACKUPCONTROLDATASET

UPDATEC BACKUPCONTROLDATASET JOURNAL(*dsname*)

UPDATEC MIGRATIONCONTROLDATASET

UPDATEC MIGRATIONCONTROLDATASET JOURNAL(*dsname*)

UPDATEC OFFLINECONTROLDATASET

UPDATEC OFFLINECONTROLDATASET JOURNAL(*dsname*)

UPDATEC ALL

UPDATEC ALL JOURNAL(*dsname*)

The forms with no journal specified use the current (active) journal data set to update the specified data set. The forms with a journal specified use a backed up journal data set requested by *dsname* to update the specified data set. For example, the command UPDATEC BACKUPCONTROLDATASET applies the transactions recorded in the active journal data set to the backup control data set. Thus, if the backup control data set were a copy of the last backup version of the backup control data set, the UPDATEC command would modify it to the state of the backup control data set before it was lost or damaged.

If the most-recent backup version of a control data set is also damaged, you can still recover the control data set by a three-step process:

1. Import the next-most-recent backup version of the data set.
2. Run UPDATEC with the journal that corresponds to the next-most-recent backup version. This journal is the most-recent *backed up* journal.
3. Run UPDATEC with the current journal.

Note: The UPDATEC command may be used to recover a damaged control data set. However, UPDATEC can take a long time to run. Because the process holds an exclusive enqueue and reserve on all the control data sets, no other DFSMSHsm functions (such as RECALL) can be run when UPDATEC is running. Therefore, a complex will not be able to do any DFSMSHsm work until UPDATEC finishes. In a JES3 environment, the whole complex may be unavailable until UPDATEC finishes processing.

VSAM considerations for the control data sets

The three control data sets are VSAM data sets and as such need to be periodically compressed and reorganized. To redistribute records or compress the control data sets, use the Access Method Services EXPORT and IMPORT commands. To move the control data sets to another volume or to increase their space allocations, use the Access Method Services REPRO, DELETE, and DEFINE commands.

Immediately after you reorganize your control data sets you should back them up.

Related reading

For more information about VSAM control data sets, see the topic about VSAM considerations for control data sets in [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Determining if control data sets are accurate

During recovery processing, a control data set can lose synchronization with the actual conditions in the system. DFSMSHsm provides a command allowing you to determine how closely the control data sets match the actual system conditions and to repair minor discrepancies.

The AUDIT command detects and reports discrepancies about data set information from:

- Computing system catalogs
- DFSMSHsm control data set records
- The VTOC of the DASD storage volume on which the data set resides

An example of the AUDIT command is:

```
AUDIT DATASETCONTROLS(MIGRATION) -  
      NOFIX OUTDATASET(outputname) REPORT(ERRORS)
```

In processing this command, DFSMSHsm reads each data set record in the MCDS. DFSMSHsm compares the information about each data set to the information contained in the system catalogs and in the VTOCs of any DASD volumes that the catalogs or the MCDS indicates the data set resides on. DFSMSHsm does not try (because of the NOFIX parameter) to reconcile any differences that it finds. Instead, DFSMSHsm sends a report of any errors to the data set specified by the OUTDATASET parameter.

Recommendation: The OUTDATASET contains executable FIXCDS commands to create or correct CDS records, but those CDS changes are only a subset of the corrective actions taken by AUDIT. In addition, the CDS records may be read multiple times during the course of an audit. If the NOFIX option is specified, the changes to a record generated on its initial read are not reflected in that record when it is read a subsequent time, which may result in incorrect actions being taken. For these reasons, the later submission of output from an AUDIT NOFIX as a substitute for an AUDIT FIX is generally discouraged.

The AUDIT command helps determine how closely the control data sets match the actual system conditions. There are Access Method Services commands that verify the structure of the CDS. The Access Method Services DIAGNOSE and EXAMINE commands can be used to validate the VSAM volume data set data structures and analyze the structural consistency of the index and data component of a key-sequenced data set.

Related reading

For more information about the AUDIT command, see [Chapter 30, “AUDIT command: Auditing DFSMSHsm,”](#) on page 737.

Repairing errors in the control data sets

For various reasons, the records in the control data set may occasionally need to be changed. You use the FIXCDS command to display or change control data set records. With the FIXCDS command, you can specify a specific record type, a specific data set, and a specific kind of change to make.

Related reading

For more information about the FIXCDS command, see [z/OS DFSMSHsm Diagnosis](#).

Procedures for handling a journal EOV error

DFSMSHsm issues message ARC0026E specifying EOV ERROR ON JOURNAL when the Journal data set becomes full. If this error is received, begin CDS backup processing by using the BACKVOL CDS command or the SETSYS command with the AUTOBACKUPSTART parameter. It is preferable that the backup of the control data sets be initiated on the system that encountered the error. This creates backup copies of the MCDS, BCDS, OCDS and nulls the journal data set so that all the space in it becomes available. If you are in a multiple DFSMSHsm-host environment, issue the RELEASE ALL command on each system that received the ARC0026E message after the completion of the backup of the data set.

Recovering from a journal I/O error

A journal I/O error may have two causes. The first cause is a corrupted journal record header (MLOG). To recover from this type of error, do the following:

Action	Command
1. Put DFSMSHsm in emergency mode.	SETSYS EMERGENCY
2. Issue the CONTROLDATASETS parameter of the BACKVOL command to back up the control and journal data sets.	BACKVOL CONTROLDATASETS
Note: This step may fail with another I/O error. If it does, see “Example 2: Replacing a damaged journal” on page 522 for more information on how to replace a damaged journal.	
3. Stop DFSMSHsm.	F DFSMSHSM,STOP
4. Use IEBGENER to copy some data into the journal data set to overwrite the existing control record.	
5. Start DFSMSHsm.	S DFSMSHSM

The second cause of a journal I/O error is a hardware error. If overwriting the journal record header does not correct the journal I/O error, a hardware malfunction probably caused the error.

To correct a hardware-caused journal I/O error, you may have to either 1) allocate a new journal data set on the same volume in a different location or 2) allocate a new journal data set on a different volume.

To allocate a new journal data set on the same DASD volume, do the following:

Action	Command
1. Put DFSMSHsm in emergency mode.	SETSYS EMERGENCY
2. Issue the CONTROLDATASETS parameter of the BACKVOL command to back up the control and journal data sets.	BACKVOL CONTROLDATASETS
Note: This step may fail with another I/O error. If it does, see “Example 2: Replacing a damaged journal” on page 522 for more information about how to replace a damaged journal.	
3. Stop DFSMSHsm.	F DFSMSHSM,STOP
4. Rename the old journal data set	
5. Allocate a new journal data set	
6. Delete the old journal data set	
7. Start DFSMSHsm.	S DFSMSHSM

To allocate a new journal data set on a different DASD volume, follow the instructions in the next topic [“Moving the journal” on page 518](#).

Moving the journal

To move the journal data set from one DASD device to another, do the following:

Action	Command
1. Put DFSMSHsm in emergency mode.	SETSYS EMERGENCY
2. Issue the CONTROLDATASETS parameter of the BACKVOL command to back up the control and journal data sets.	BACKVOL CONTROLDATASETS
3. Stop DFSMSHsm.	F DFSMSHSM,STOP
4. Delete the old journal data set	
5. Allocate a new journal data set on a different DASD device.	
6. Start DFSMSHsm.	S DFSMSHSM

Steps to ensure that the control data sets are accurate

After you recover the control data sets, you can do the following to ensure that they are accurate:

1. Issue the AUDIT commands to determine if any discrepancies exist between the data set records in the recovered control data sets and the data set records in the computing system catalog.
2. Issue the AUDIT commands with the FIX parameter to fix any errors that AUDIT can correct.
3. Issue the FIXCDS command to fix any minor problems encountered in the recovered control data sets.

Control data set backup and recovery examples

This topic provides some examples of recovering your control data sets and journal data set.

Recovery examples using the enhanced CDS recovery function

The following examples help you recover your control data sets or journal data sets using the enhanced CDS recovery function.

Example 1: Recovering the MCDS from DASD

This example shows how you set up your environment, how to recover the MCDS, and the messages that are issued during the backup and recovery processing.

Set up your environment with the following SETSYS command:

```
SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(3) -
    BACKUPDEVICECATEGORY(DASD))
```

Issue the QUERY CDSVERSIONBACKUP command to find out what the last final qualifier was for the backup data sets. DFSMSHsm issues the following messages:

ARC0375I

CDSVERSIONBACKUP,

MCDSBACKUPDSN=*uid*.MCDS.BACKUP,

BCDSBACKUPDSN=*uid*.BCDS.BACKUP,

OCDSBACKUPDSN=*uid*.OCDS.BACKUP,

JRNBACKUPDSN=*uid*.JRNL.BACKUP

ARC0376I

BACKUPCOPIES=3,

BACKUPDEVICECATEGORY=DASD,

LATESTFINALQUALIFIER=V0000000,

DATAMOVER=HSM

Note: Since the LATESTFINALQUALIFIER begins with a V, the last data mover was DFSMSHsm.

Because no backup data sets are preallocated, you preallocate the following backup data sets:

uid.MCDS.BACKUP.V0000001 through *uid*.MCDS.BACKUP.V0000003

uid.BCDS.BACKUP.V0000001 through *uid*.BCDS.BACKUP.V0000003

uid.OCDS.BACKUP.V0000001 through *uid*.OCDS.BACKUP.V0000003

uid.JRNL.BACKUP.V0000001 through *uid*.JRNL.BACKUP.V0000003

Now you let DFSMSHsm execute to a point where DFSMSHsm has successfully backed up the control data sets and the journal data set ten times. The backup data sets have the final qualifiers of V0000008, V0000009, and V0000010.

As DFSMSHsm processes the MCDS, an I/O error occurs, so a part of the MCDS is no longer usable. To recover the MCDS, you would:

1. Issue QUERY CDSVERSIONBACKUP to find out what was the latest backup version. DFSMSHsm issues the following messages:

ARC0375I

CDSVERSIONBACKUP,

MCDSBACKUPDSN=*uid*.MCDS.BACKUP,

BCDSBACKUPDSN=*uid*.BCDS.BACKUP,

OCDSBACKUPDSN=*uid*.OCDS.BACKUP,

JRNLBKUPDSN=*uid*.JRNL.BACKUP

ARC0376I

BACKUPCOPIES=3,

BACKUPDEVICECATEGORY=DASD,

LATESTFINALQUALIFIER=V0000010,

DATAMOVER=HSM

Recommendation: If the QUERY CDSVERSIONBACKUP command is not successful or if DFSMSHsm is not active, issue the following IDCAMS LISTCAT command to determine the latest version: LISTCAT LEVEL(*uid*)

2. Issue a IDCAMS LISTCAT(*uid*) to determine if the backup copy is good. This determination is made by checking the low-level qualifier for a V or an X.

3. Stop DFSMSHsm in all processing units.
4. See [“Requirements for using the enhanced CDS recovery function”](#) on page 508 for information on using the enhanced CDS recovery function to recover the MCDS.
5. Restart DFSMSHsm in the processing unit.
6. Verify that the control data sets are correct (see [“Determining if control data sets are accurate”](#) on page 517).
7. Fix any errors in the control data sets (see [“Repairing errors in the control data sets”](#) on page 517).
8. Because you are recovering the MCDS, issue a QUERY CDSVERSIONBACKUP command to ensure that the latest final qualifier is the same as that shown in step 1. If you do not have the output from query command in step 1, you can determine the final qualifier of the latest backup copy from your catalog using the LISTCAT LEVEL command, as shown in the following example:

```
LISTCAT LEVEL(uid)
```

If these qualifiers do not match, use the FIXCDS patch command to correct the field at offset X'B1' in the MHCR record to reflect the latest version in the catalog, as shown in the following example:

```
FIXCDS S MHCR PATCH(X'B1' 0000010)
```

9. Issue the BACKVOL CDS command to ensure that all copies of the control data sets and the journal data set are current. During backup of the control data sets and the journal data set, DFSMSHsm issues the following messages:

ARC0740I

CDS BACKUP STARTING AT 04:15:35 ON 95/05/24 ON SYSTEM 381A TO DASD IN PARALLEL MODE, DATAMOVER=HSM

ARC0742I

BACKUP FOR MCDS STARTING AT 04:15:55 ON 95/05/24, BACKUP COPY TECHNIQUE IS UNDEFINED

ARC0743I

MCDS SUCCESSFULLY BACKED UP TO *uid*.MCDS.BACKUP.V0000008, ON VOLUMES VOL001,VOL002, TIME=04:20:03, DATE=95/05/24

ARC0742I

BACKUP FOR BCDS STARTING AT 04:25:55 ON 95/05/24, BACKUP COPY TECHNIQUE IS UNDEFINED

ARC0743I

BCDS SUCCESSFULLY BACKED UP TO *uid*.BCDS.BACKUP.V0000008, ON VOLUMES VOL003,VOL004, TIME=04:30:13, DATE=95/05/24

ARC0742I

BACKUP FOR OCDS STARTING AT 04:35:55 ON 95/05/24, BACKUP COPY TECHNIQUE IS UNDEFINED

ARC0743I

OCDS SUCCESSFULLY BACKED UP TO *uid*.OCDS.BACKUP.V0000008, ON VOLUME VOL005,
TIME=04:40:44, DATE=95/05/24

ARC0750I

BACKUP FOR JRNL STARTING AT 04:40:55 ON 95/05/24

ARC0743I

JRNL SUCCESSFULLY BACKED UP TO *uid*.JRNL.BACKUP.V0000008, ON VOLUME VOL006,
TIME=04:45:26, DATE=95/05/24

ARC0748I

LAST SUCCESSFUL CDS BACKUP-SET QUALIFIER IS V0000011

ARC0741I

CDS BACKUP ENDING AT 04:45:35 ON 95/05/24, STATUS=SUCCESSFUL

Note: Because you have specified three backup copies and you are backing up to DASD, DFSMSHsm replaces the V0000008 version of the backup data sets with V0000011.

If you issue the QUERY CDSVERSIONBACKUP command again, DFSMSHsm issues the following messages:

ARC0375I

CDSVERSIONBACKUP,

MCDSBACKUPDSN=*uid*.MCDS.BACKUP,

BCDSBACKUPDSN=*uid*.BCDS.BACKUP,

OCDSBACKUPDSN=*uid*.OCDS.BACKUP,

JRNLBKUPDSN=*uid*.JRNL.BACKUP

ARC0376I

BACKUPCOPIES=3,

BACKUPDEVICECATEGORY=DASD,

LATESTFINALQUALIFIER=V0000011,

DATAMOVER=HSM

10. Restart DFSMSHsm in the other processing units.

Example 2: Replacing a damaged journal

This example shows how you set up your environment, how to recover the journal, and some of the messages that are issued during the backup and recovery processing.

Set up your environment with the following SETSYS command:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -
  BACKUPCOPIES(4) -
  MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
  BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
  OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
  JRNLBACKUPDSN(BHSM.JRNL.BACKUP) -
  BACKUPDEVICECATEGORY(DASD))
```

You preallocate the following backup data sets:

```
BHSM.MCDS.BACKUP.D0000001
BHSM.BCDS.BACKUP.D0000001
BHSM.OCDS.BACKUP.D0000001
BHSM.JRNL.BACKUP.D0000001
```

```
BHSM.MCDS.BACKUP.D0000002
BHSM.BCDS.BACKUP.D0000002
BHSM.OCDS.BACKUP.D0000002
BHSM.JRNL.BACKUP.D0000002
```

```
BHSM.MCDS.BACKUP.D0000003
BHSM.BCDS.BACKUP.D0000003
BHSM.OCDS.BACKUP.D0000003
BHSM.JRNL.BACKUP.D0000003
```

```
BHSM.MCDS.BACKUP.D0000004
BHSM.BCDS.BACKUP.D0000004
BHSM.OCDS.BACKUP.D0000004
BHSM.JRNL.BACKUP.D0000004
```

Assume DFSMSHsm has successfully backed up the control data sets and the journal data set ten times. The backup data sets have the final qualifiers of D0000007, D0000008, D0000009, and D0000010.

As DFSMSHsm processes the journal data set, an I/O error occurs. DFSMSHsm disables the journal data set and issues the following message:

```
ARC0026A JOURNALING DISABLED DUE TO I/O ERROR.
          MIGRATION, BACKUP, DUMP, AND RECYCLE HELD.
```

Issue a BACKVOL CDS command to back up the control data sets and the journal data set. DFSMSHsm successfully backs up the control data sets and issues message ARC0743I for each backup. The message indicates that DFSMSHsm made the backup to the backup data set whose final qualifier is D0000007 (oldest version).

When DFSMSHsm tries to back up the journal, another I/O error occurs. Because DFSMSHsm cannot successfully complete the journal data set backup function, it renames the journal backup data set to X0000011. Then DFSMSHsm issues the following message:

```
ARC0744E JRNL COULD NOT BE BACKED UP,
          RC=retcode, REASON=reascode
          MIGRATION, BACKUP, DUMP, AND RECYCLE HELD.
```

In addition, DFSMSHsm issues the following message to indicate the latest final backup data set qualifier (D0000011). DFSMSHsm waits for the completion of the CDS backups and then nulls (empties) the journal data set if no CDS backup failures occur.

```
ARC0748I LAST SUCCESSFUL CDS BACKUP-SET QUALIFIER IS
          D0000011
```

At this point, assume the following conditions exist:

- The latest backup versions of the control data sets are backup data sets whose final qualifier is D0000011.
- The latest good backup version for the journal data set is a backup data set whose final qualifier is D0000010.
- The current control data sets are usable.

If you want to continue processing with the journal, you must stop DFSMSHsm, delete and reallocate the journal data set to avoid getting another I/O error during processing of the data set. This process is described in “Reasons for recovering the control data sets” on page 507. A hardware problem could be causing the I/O error on the journal data set, so be sure you do not re-allocate the journal data set to where it was allocated before the I/O error occurred.

After you restart DFSMSHsm, issue BACKVOL CDS to back up the control data sets and the new, empty journal data set. DFSMSHsm backs up the control data sets and the journal data set to the backup data sets whose final qualifiers are D0000008. If DFSMSHsm successfully backs up the control data sets and the journal data set, DFSMSHsm renames the backup data sets with the final qualifier of D0000012 (or V0000012 if IDCAMS is the data mover).

Example 3: Recovering the MCDS from tape when the latest MCDS backup version is unusable

This example shows how to set up your environment, how to recover the MCDS when the most recent backup version is unusable, and the processing that occurs when recovering the MCDS.

Set up your environment with the following SETSYS command:

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) -  
  BACKUPCOPIES(6) -  
  MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -  
  BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -  
  OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -  
  JRNBACKUPDSN(BHSM.JRNL.BACKUP) -  
  BACKUPDEVICECATEGORY(TAPE))
```

Assume DFSMSHsm has successfully backed up the control data sets and the journal data set six times, and the latest final qualifier is D0000006. Therefore, backup data sets with final qualifiers of D0000001 through D0000006 exist on tape volumes.

If an error occurs so DFSMSHsm cannot access the MCDS, you:

1. Issue QUERY CDSVERSIONBACKUP and observe that the latest final qualifier is D0000006.
2. Using DFSMSdss, restore the MCDS backup version of D0000006. Assume that, when you attempt this restore, the restore fails. You must now use MCDS backup version BHSM.MCDS.BACKUP.D0000005.
3. Using DFSMSdss, restore the MCDS backup version of D0000005.
4. Using IDCAMS, export the D0000005 backup version of the MCDS that was restored in step 3. The following example shows JCL that can be used to do the export:

```
//EXPORT JOB ,MSGLEVEL=(1,1),MSGCLASS=A,REGION=1024K  
//STEP1 EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//OUTDD1 DD DISP=(,CATLG,DELETE),BUFNO=26,BLKSIZE=28332,  
//          SPACE=(CYL,(200,20),RLSE),  
//          UNIT=3390,VOL=SER=333333,  
//          DSN=BHSM.MCDS.BACKUP.TEMP  
//SYSIN DD *  
EXPORT BHSM.MCDS.BACKUP.D0000005 -  
  OUTFILE(OUTDD1) -  
  TEMPORARY  
/*
```

Where:

BHSM.MCDS.BACKUP.TEMP is the data set name of the exported copy being created.

BHSM.MCDS.BACKUP.D0000005 is the data set name of the control data set being exported.

BLKSIZE=28332 is the capacity of half a track on a 3390 device.

5. Stop DFSMSHsm in all processing units.
6. Create the backup journal file using the online journal and the latest backup copy of the journal. Use ARCBJRNL and PARM=' BOTH ', JRNLO DD for the online journal, and BKUPJRNL DD for the journal backup copy version D0000006 to create the backup journal file. See [“Implementing enhanced CDS recovery using JCL” on page 508](#) for an example of the JCL needed to create the backup journal file.
7. Sort this newly created backup journal file. See [“Implementing enhanced CDS recovery using JCL” on page 508](#) for an example of the JCL needed to sort the backup journal file.
8. Import and merge the exported MCDS from step 4 and the sorted backup journal file from step 7 using ARCIMPRT. Again, see [“Implementing enhanced CDS recovery using JCL” on page 508](#) for an example of the JCL needed to do the import and merge.
9. Use the PATCH parameter of the FIXCDS command to update the MHCR to version 6, the latest control data set backup version.
10. Issue the BACKVOL CDS command to obtain a new set of backup copies of the control data sets and journal data set. When DFSMSHsm successfully backs up these data sets, the latest backup copies are those data sets whose final qualifier is D0000007.

DFSMSHsm uncatalogs the D0000001 backup version. Only versions D0000002 through D0000007 remain cataloged.

Performance considerations in a multiple DFSMSHsm host environment

Review the following performance consideration if you have configured your system to use a multiple DFSMSHsm host environment:

- ARCGPA and ARCCAT resources are held when a DFSMSHsm function requires an update to or read of a control data set (CDS) record. While the resources are held, all other functions must wait for the controlling function to complete its task. In some cases, the wait is long and can delay new DFSMSHsm request or functions from starting. For example, when a CDS backup task appears to be hung, it might be waiting for a large data set recall task to complete. Furthermore, new recall requests wait for both the original recall request and the CDS backup to complete before being processed. Specifically:
 - In a record level sharing (RLS) environment, a CDS backup must wait for all DFSMSHsm functions across the HSMplex to complete.
 - In a non-RLS environment, a CDS backup must wait for all DFSMSHsm functions within the same LPAR as the DFSMSHsm host performing the CDS backup to complete.

To minimize this delay, DFSMSHsm hosts in the same HSMplex (RLS environment) or within the same LPAR (non-RLS environment) allow CDS backup to begin immediately after all pending CDS updates are complete. The host performing the CDS backup sends a notification to the other hosts using cross-system coupling facility (XCF) services. All hosts processing functions and tasks that might delay CDS backup complete pending CDS updates, suspend new CDS updates, and release all resources (such as ARCGPA and ARCCAT) necessary for CDS backup to begin. After CDS backup is complete, suspended functions and tasks resume.

- Because all DFSMSHsm activity is quiesced during quiesced journal backup, user and production jobs that require recall or recovery resources wait until the entire CDS and journal backup is complete. Therefore, the impact of journal backup on DFSMSHsm availability and performance should be considered when planning a backup schedule.

Non-intrusive journal backup can be used to reduce the impact on DFSMSHsm availability and performance when concurrent copy is available for CDS backup and the SETSYS JOURNAL(RECOVERY) setting is in effect. This method of journal backup does not hold resources during journal backup, which allows DFSMSHsm activity to continue. Resources are held only at the end of journal backup while changes to the journal that occurred during creation of the initial backup copy are appended to the

backup copy. The control data sets are then backed up using concurrent copy. For more information about the non-intrusive journal backup method, see the topic [“Using non-intrusive journal backup”](#) on page 496.

- When there are multiple HSMplexes within a z/OS sysplex, the startup procedure keyword PLEXNAME should be specified. For more information about the SETSYS PLEXNAME command, see [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,”](#) on page 1057.

MASH configuration considerations

When running in a multiple address space DFSMSHsm (MASH) configuration where XCF services are not available, the following should be considered:

- Long running functions on other DFSMSHsm hosts (including those that start before automatic CDS backup) will not release the resources required to allow CDS backup to obtain an exclusive lock. This prevents starting CDS backup.
- The enqueue prevents all data set or volume-type functions from accessing the control data sets while they are being backed up. Therefore, you should carefully consider the best time to start backup of the control data sets.

Chapter 13. Managing DFSMSHsm media

As time passes, conditions in your computer system change such that you will need to manage the DFSMSHsm-owned storage media in your system. For example:

- When the data on backup or ML2 tapes expires, you then want to consolidate the valid data scattered over many tapes.
- You may expect many data sets to be migrated and want to make room on the migration volumes for the new data sets before it is done automatically.
- You may be replacing one DASD device type with another DASD or tape device type and want to move the data sets that are on level 0, level 1, and level 2 volumes to the new device types.
- You may want to remove volumes from DFSMSHsm management or ownership.

Moving all data sets from one level 0 volume to other level 0 volumes

Device conversion of level 0 volumes is performed expeditiously by DFSMSdss. However, the following capabilities exist and are useful under some conditions.

You can use DFSMSHsm to move single data sets or all data sets from one level 0 volume to other level 0 volumes. The processing for non-SMS-managed data sets differs from that for SMS-managed data sets.

To move BDAM data sets between volumes of different track sizes, you might consider migrating and then recalling that data set using the DAOPTION.

The commands for moving SMS-managed data sets are:

```
MIGRATE DATASETNAME(dsname) CONVERT
MIGRATE DATASETNAME(dsname) CONVERT(volser2)
MIGRATE VOLUME(volser1 MIGRATE(0)) CONVERT
MIGRATE VOLUME(volser1 MIGRATE(0)) CONVERT(volser2)
```

The commands for moving all non-SMS-managed data sets from one level 0 volume to one or more other level 0 volumes are:

```
MIGRATE DATASETNAME(dsname) CONVERT
MIGRATE DATASETNAME(dsname) -
  CONVERT(volser2 unittype2)
MIGRATE VOLUME(volser1 MIGRATE(0)) CONVERT
MIGRATE VOLUME(volser1 MIGRATE(0)) -
  CONVERT(volser2 unittype2)
MIGRATE VOLUME(volser1 MIGRATE(0)) CONVERT -
  UNIT(unittype1)
MIGRATE VOLUME(volser1 MIGRATE(0)) -
  CONVERT(volser2 unittype2) UNIT(unittype1)
```

Processing SMS-managed data sets

The main difference in the processing of SMS-managed data sets is that you cannot direct the movement to a specific volume. Although you can specify a volume, SMS-processing determines the actual volume that the data set resides on after being moved. The process for moving data sets migrates the data sets

to ML1 volumes and recalls them to level 0 volumes. At the beginning of processing for MIGRATE VOLUME commands, DFSMSShsm obtains a list of the management classes active in the system. See [“DFSMSShsm retrieval of management classes and storage groups”](#) on page 154 for a discussion of the exception. DFSMSShsm does not obtain a list of management classes for MIGRATE DATASETNAME commands. As it processes each data set, DFSMSShsm checks the value specified for the data set by the COMMAND OR AUTOMIGRATE attribute in the management class. If the value of the attribute is BOTH, DFSMSShsm processes the data set.

In recalling each data set, DFSMSShsm invokes the automatic class selection (ACS) services of DFSMSdftp and passes the information described in [“Automatic recall”](#) on page 83.

If SMS is installed and active, the ACS routine may return to DFSMSShsm a storage class and, optionally, a management class. If the ACS routine returns a storage class, DFSMSShsm passes the data set name to DFSMSdftp, along with its storage class name and management class name (if present). DFSMSdftp selects a volume and places the data set on it. The volume selected is not necessarily the target volume specified by the MIGRATE command. The catalog entry is updated to reflect the current status.

To prevent SMS from selecting the source volume as the target volume for the recall, change the status attribute for the volume in the storage group. Suitable status attributes are DISABLENEW and QUIESCENEW.

If all the data sets on a volume are being processed (MIGRATE VOLUME is specified), any data set that should be expired is expired without being migrated.

Processing non-SMS-managed data sets

There are two differences in processing for non-SMS-managed data sets:

- No storage class, management class, and data class are received.
- Data sets are recalled to the volume specified.

The target unit type need not be specified if the volume has been added to DFSMSShsm by an ADDVOL command.

For non-SMS-managed data sets, DFSMSShsm migrates the data sets to an ML1 volume and then recalls them to the volume specified by *volser2*. If *volser2* is not specified, DFSMSShsm selects the level 0 volumes to recall the data sets to in the manner specified in [“Recall”](#) on page 119. If *volser2* has not been identified to DFSMSShsm, use *unitttype2* to specify the type of unit that *volser2* can be mounted on. If *volser1* has not been identified to DFSMSShsm, use the UNIT(*unitttype1*) parameter to specify the type of unit that *volser1* can be mounted on. To prevent selection of the source volume as the target volume for the recall, you must do two things:

- Specify the volume to DFSMSShsm with NOAUTORECALL in the ADDVOL command
- Remove the volume from any pools of which it is a member

With BDAM data sets, when you specify a target volume on a different device type during DASD conversion, DFSMSShsm recalls the data set you specify if the data on a track of the source volume fits on a track of the target volume. However, you are responsible for data addressability.

DFSMSShsm maintains the same number of tracks for BDAM data set space during DASD conversion to a different device type. There is no change in the number of records per track, so space is wasted when you convert a BDAM data set to a DASD with a larger track size. DFSMSShsm assumes that any user program accessing BDAM data depends on having the same relative track address.

DFSMSShsm does not compact data sets or migrate them to SDSP data sets when it does DASD conversion.

If you specify the CONVERT parameter in a direct-to-tape environment, the command fails.

Examples of moving data sets between level 0 volumes

Consider what happens if you use the MIGRATE CONVERT command in the system shown in [Figure 83](#) on page 529.

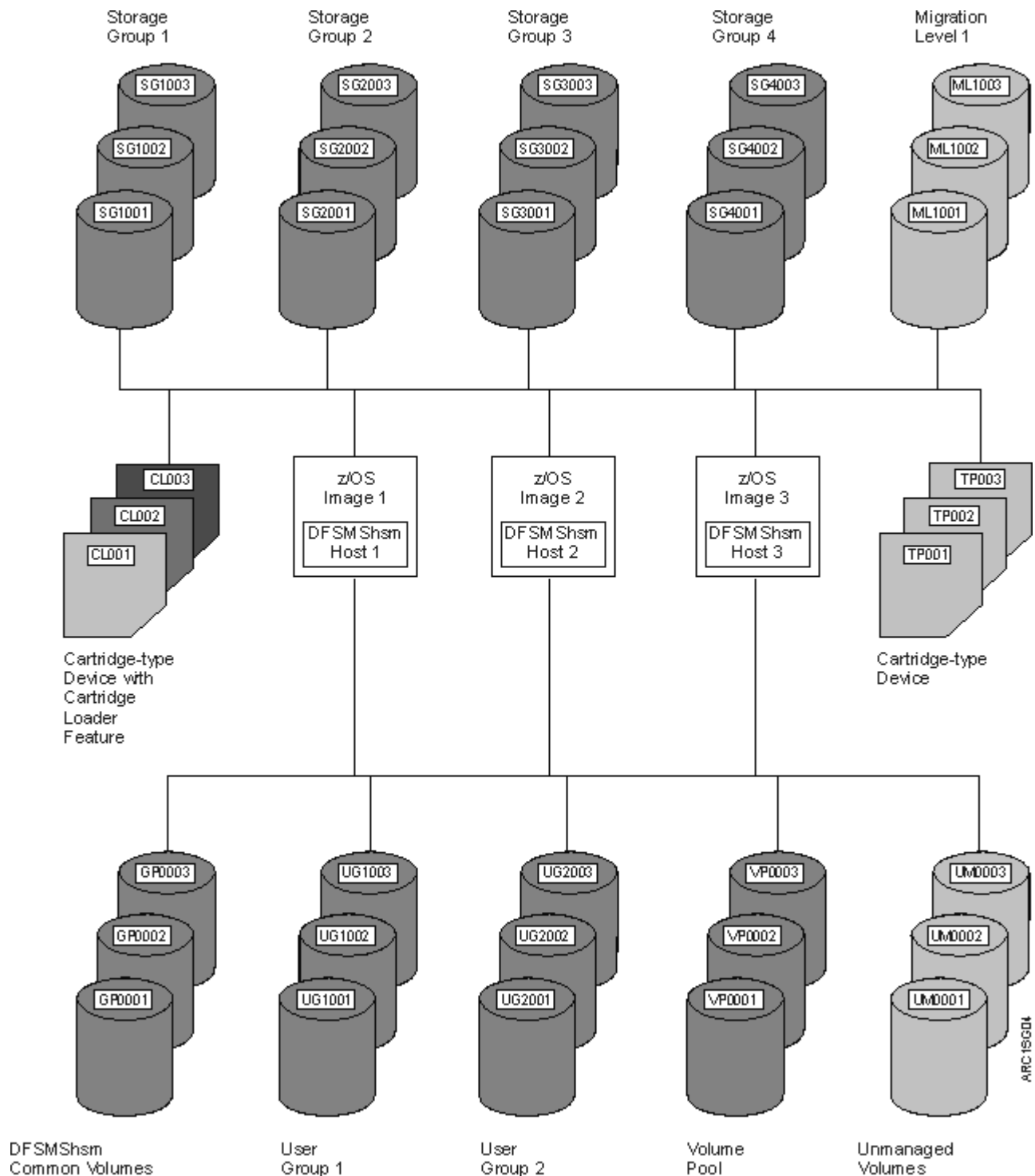


Figure 83. Example System for MIGRATE CONVERT

SMS-managed volume

Assume that you want to remove volume SG2002 from the system and replace it with a new volume. You must do three things:

- Prevent new allocations to volume SG2002
- Remove the data sets that DFSMSHsm can process from the volume
- Remove the data sets that DFSMSHsm cannot process from the volume

To prevent new allocations to the volume, you must change the status of the volume in the storage group. For example, you can establish the volume status in the storage group definition as shown in [Figure 84](#) on [page 530](#).

```
STORAGE GROUP NAME: STGGP2
VOLUME SERIAL NUMBER: SG2002
```

SYSTEM NAME		SMS VOL STATUS
-----		-----
PROCUN1	====>	DISNEW
PROCUN2	====>	DISNEW
PROCUN3	====>	DISNEW

Figure 84. SMS Volume Status

To remove the data sets that DFSMSHsm can process, you can issue the following command:

```
MIGRATE VOLUME(SG2002 MIGRATE(0)) CONVERT
```

DFSMSHsm migrates all of the data sets that it can process off of volume SG2002 and recalls them to other SMS-managed volumes. The DFSMSdfp automatic class selection routine selects the volume for data set recall.

To move any data sets that DFSMSHsm cannot process, you can use DFSMSdss.

Non-SMS-managed volume

Assume that you want to remove all data sets from volume UG2003. You still must prevent allocations to the volume, remove data sets that DFSMSHsm can process from the volume, and remove any data sets that DFSMSHsm cannot process from the volume.

Preventing users other than DFSMSHsm from allocating to the volume is a normal MVS process. You can prevent DFSMSHsm from selecting the volume for recall of data sets by removing UG2003 from the user group with the following command in all z/OS images:

```
DEFINE POOL(GROUP2 VOLUMES(UG2001 UG2002))
```

Because you have volume UG2003 dedicated to GROUP2, you have specified the NOAUTORECALL parameter of the ADDVOL command. Thus, because the volume does not belong to any pool and has the attribute of NOAUTORECALL, DFSMSHsm cannot recall any data sets to it.

To move the data sets that DFSMSHsm can process, issue the following command:

```
MIGRATE VOLUME(UG2003 MIGRATE(0)) CONVERT
```

Notice that this is the same command used for the SMS-managed volume. DFSMSHsm again migrates and recalls all of the data sets on the volume that it can process. However, because the volume is not SMS managed (because the ACS routines indicate the data sets should be non-SMS-managed), DFSMSHsm selects the volumes to which the data sets are recalled. Because, presumably, all of the data sets on volume UG2003 belong to GROUP2, DFSMSHsm recalls them to either UG2001 or UG2002.

To move any data sets that DFSMSHsm cannot process, you can use DFSMSdss.

Moving some data sets from migration level 1 volumes

Two commands, FREEVOL and MIGRATE, are available to move data from ML1 volumes. You can move data sets from individual ML1 volumes with the FREEVOL command, or you can move or delete data sets from all ML1 volumes with the MIGRATE command. If you intend to make more space available on ML1 volumes, use the MIGRATE ML1 command.

Note: The "data" and "data sets" mentioned here and in the next few topics refer to data sets that have been created by DFSMSHsm. If your installation stores non-SMS-managed user or system data on ML1 volumes (refer to *z/OS DFSMSHsm Implementation and Customization Guide* for further restrictions), this data is not affected by FREEVOL and MIGRATE commands.

Use the FREEVOL AGE(0) command to empty DASD migration or DASD backup volumes in preparation for movement to new equipment or to replace partially damaged volumes. If you are attempting to empty an ML1 volume, there can be two types of data that need to be moved off the volume: backup versions and migrated data that must remain until it is backed up. Since backup of the migrated data only occurs when automatic backup is run on the primary DFSMSHsm host, ensure that all data sets are removed from the ML1 volume by running automatic backup on your primary DFSMSHsm host shortly before issuing the FREEVOL AGE(0) command.

Using the FREEVOL command for migration level 1 volumes

The FREEVOL command allows you to move data sets from a single ML1 volume to other ML1 volumes or to ML2 volumes. This command can be useful when converting to or from the use of SDSPs when the target is level 1. Selection of the target volumes can be manipulated by using the DRAIN parameter of the ADDVOL command. You can use the following commands to move data sets that exceed a certain age:

```
FREEVOL MIGRATIONVOLUME(volser) AGE(days)
FREEVOL MIGRATIONVOLUME(volser) AGE(days) -
  TARGETLEVEL(MIGRATIONLEVEL1)
FREEVOL MIGRATIONVOLUME(volser) AGE(days) -
  TARGETLEVEL(MIGRATIONLEVEL2)
FREEVOL MIGRATIONVOLUME(volser) AGE(days) -
  TARGETLEVEL(MIGRATIONLEVEL2(DASD))
FREEVOL MIGRATIONVOLUME(volser) AGE(days) -
  TARGETLEVEL(MIGRATIONLEVEL2(TAPE))
```

If you use the data set backup functions BACKDS, HBACKDS, ARCINBAK, or ARCHBACK and run short of ML1 space, you can move BACKDS versions from all ML1 volumes to daily backup volumes by using the FREEVOL ML1BACKUPVERSIONS command.

In all forms of the command used in this topic, the value for the AGE parameter must be greater than zero. Processing of the commands is different for SMS-managed and non-SMS-managed data sets.

Processing for SMS-managed data sets

At the beginning of processing, DFSMSHsm obtains a list of the active management classes in the system. See [“DFSMSHsm retrieval of management classes and storage groups” on page 154](#) for a discussion of the exception. DFSMSHsm then calculates the inactive age of each data set and compares it to the value of the LEVEL 1 DAYS NON-USAGE attribute in the management class with which the data set is associated. If the inactive age is greater than or equal to the value of the LEVEL 1 DAYS NON-USAGE attribute, the data set migrates to an ML2 volume, unless the value of the attribute is NOLIMIT. If the attribute value is NOLIMIT, the data set does not migrate.

The TARGETLEVEL parameter does not determine the level to which SMS-managed data sets migrate. However, if the MIGRATIONLEVEL2 subparameter is specified with a device category of TAPE or DASD, the data set migrates to an ML2 volume of the category specified. If the MIGRATIONLEVEL2 subparameter does not specify a device category, the data set migrates to a level 2 volume of the category specified by the TAPEMIGRATION parameter of the SETSYS command.

Processing for non-SMS-managed data sets

For non-SMS-managed data sets, DFSMSHsm compares the time since the data set was last opened against the value specified with the AGE parameter. If the time since the data set was last opened is equal to or more than the value specified with the AGE parameter, DFSMSHsm migrates the data set to a volume of the level specified by the TARGETLEVEL parameter. If the MIGRATIONLEVEL2 subparameter

of the TARGETLEVEL parameter specifies a device category, the data set migrates to an ML2 volume of the category specified. If the MIGRATIONLEVEL2 subparameter does not specify a device category, the data set migrates to an ML2 volume of the category specified by the TAPEMIGRATION parameter of the SETSYS command.

Using the MIGRATE command for migration level 1 volumes

With the MIGRATE command you can either migrate data sets from all the ML1 volumes or delete migrated data sets from individual ML1 volumes. The commands to delete the migrated data sets do not affect SMS-managed data sets.

Migrating data sets

The commands to migrate data sets are:

```
MIGRATE MIGRATIONLEVEL1
MIGRATE MIGRATIONLEVEL1 DAYS(days)
```

For each form of the command, SMS-managed data sets and non-SMS-managed data sets are processed differently.

Processing for SMS-managed data sets

For SMS-managed data sets, DFSMSHsm obtains a list of the management classes that are active in the system, unless a volume function is processing or has just completed. See [“DFSMSHsm retrieval of management classes and storage groups”](#) on page 154 for a discussion of the exception. It then compares the time since the data set was last opened to the value of the LEVEL 1 DAYS NON-USAGE attribute in the appropriate management class for each data set. If the time since the data set was last opened is equal to or more than the value in the LEVEL 1 DAYS NON-USAGE parameter, the data set migrates to ML2. Otherwise, the data set remains on the ML1 volume. If the value of the LEVEL 1 DAYS NON-USAGE attribute is NOLIMIT, the data set remains on the ML1 volume.

Processing for non-SMS-managed data sets

For non-SMS-managed data sets, the processing depends on whether the DAYS parameter is specified on the MIGRATE command.

If the DAYS parameter is *not* specified on the MIGRATE command, DFSMSHsm checks the time since the data set was last opened against the value of the MIGRATIONLEVEL1DAYS parameter of the SETSYS command. If the data set has not been opened for a number of days more than or equal to the value specified by MIGRATIONLEVEL1DAYS, the data set migrates to ML2.

If the DAYS parameter is specified on the MIGRATE command, DFSMSHsm checks the time since the data set was last opened against the value specified with the DAYS parameter. If the data set has not been opened for a number of days more than or equal to the value specified by the DAYS parameter, the data set migrates to ML2.

Deleting non-SMS-managed data sets from migration volumes

You can use the MIGRATE command to delete obsolete non-SMS-managed data sets from migration volumes. The forms of the command to delete data sets are:

```
MIGRATE VOLUME(volser DELETEBYAGE(days))
MIGRATE VOLUME(volser DELETEIFBACKEDUP(days))
```

If you specify DELETEBYAGE, DFSMSHsm deletes any non-SMS-managed data set that has not been opened for more days than is specified by DELETEBYAGE(*days*) from the migration volume. It does not delete any data set that has not passed the expiration date indicated in the data set's VTOC entry.

If you specify `DELETEIFBACKEDUP`, DFSMSHsm deletes any non-SMS-managed data set that has a valid backup version and that has not been opened for more days than is specified by `DELETEIFBACKEDUP(days)`. DFSMSHsm also marks the latest valid backup copy as a retired version. This command deletes the data set regardless of whether the data set has passed the expiration date indicated in the data set's VTOC entry.

Moving all the data sets from a migration level 1 volume

You can move all the data sets from an ML1 volume either to other ML1 volumes or to ML2 volumes. Use the following commands to move all data sets:

```
FREEVOL MIGRATIONVOLUME(volser)
FREEVOL MIGRATIONVOLUME(volser) AGE(0)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL1)
FREEVOL MIGRATIONVOLUME(volser) AGE(0) -
  TARGETLEVEL(MIGRATIONLEVEL1)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2)
FREEVOL MIGRATIONVOLUME(volser) AGE(0) -
  TARGETLEVEL(MIGRATIONLEVEL2)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2(DASD))
FREEVOL MIGRATIONVOLUME(volser) AGE(0) -
  TARGETLEVEL(MIGRATIONLEVEL2(DASD))
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2(TAPE))
FREEVOL MIGRATIONVOLUME(volser) AGE(0) -
  TARGETLEVEL(MIGRATIONLEVEL2(TAPE))
```

For this operation, the AGE parameter must be omitted or must be zero. Processing is different for SMS-managed data sets and for non-SMS-managed data sets. For either type of data set, DFSMSHsm moves all the migration copies on the volume. Small data sets that move to an ML1 volume can go to SDSP data sets.

Processing for SMS-managed data sets

At the beginning of processing, DFSMSHsm obtains a list of the management classes that are active in the system, unless a volume function is processing or has just completed. See [“DFSMSHsm retrieval of management classes and storage groups”](#) on page 154 for a discussion of the exception. DFSMSHsm then compares the time since each data set was last opened to the value specified with the LEVEL 1 DAYS NON-USAGE attribute in the management class with which the data set is associated. If the time since the data set was last opened is equal to or more than the value specified with the LEVEL 1 DAYS NON-USAGE attribute, the data set migrates to an ML2 volume. If either the time since the data set was last opened is less than the value specified with the LEVEL 1 DAYS NON-USAGE attribute or the attribute value is NOLIMIT, the data set migrates to another ML1 volume.

DFSMSHsm does not use the TARGETLEVEL parameter to determine the migration level to which to migrate an SMS-managed data set. However, if the TARGETLEVEL parameter specifies MIGRATIONLEVEL2 with a device category, any SMS-managed data set that migrates to ML2 migrates to a device of the category specified by the MIGRATIONLEVEL2 subparameter. If the MIGRATIONLEVEL2 subparameter does not specify a device category, any SMS-managed data set that migrates to ML2 migrates to a device of the category specified by the TAPEMIGRATION parameter of the SETSYS command.

Processing for non-SMS-managed data sets

For non-SMS-managed data sets, DFSMSHsm migrates all data sets from the ML1 volume to either:

- ML2 volumes if TARGETLEVEL is not specified
- The level of volumes specified by the TARGETLEVEL parameter

If a device category is specified by the MIGRATIONLEVEL2 subparameter of the TARGETLEVEL parameter, DFSMSHsm migrates the data sets to the category of device specified by the MIGRATIONLEVEL2 subparameter. If the MIGRATIONLEVEL2 subparameter does not specify a device category, DFSMSHsm migrates the data sets to the category of device specified by the TAPEMIGRATION parameter of the SETSYS command.

Moving data sets from a DASD migration level 2 volume to other migration level 2 volumes

You can use DFSMSHsm to move data sets from one DASD ML2 volume to other ML2 volumes. The FREEVOL command is used to move the data sets.

Use the following commands to move data sets between ML2 volumes:

```
FREEVOL MIGRATIONVOLUME(volser)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2(DASD))
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2(TAPE))
FREEVOL MIGRATIONVOLUME(volser) AGE(0)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2) AGE(0)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2(DASD)) AGE(0)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2(TAPE)) AGE(0)
FREEVOL MIGRATIONVOLUME(volser) AGE(days)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2) AGE(days)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2(DASD)) AGE(days)
FREEVOL MIGRATIONVOLUME(volser) -
  TARGETLEVEL(MIGRATIONLEVEL2(TAPE)) AGE(days)
```

For the forms of the command with no AGE specified or with AGE(0) specified, DFSMSHsm migrates all SMS-managed and non-SMS-managed data sets to other ML2 volumes. For the forms of the command with an age greater than 0 specified, DFSMSHsm moves only the data sets that are equal to or exceed the specified inactive age. The category of volume is specified by the TARGETLEVEL(MIGRATIONLEVEL2) parameter. If a device category is specified with the MIGRATIONLEVEL2 subparameter, the data sets migrate to volumes of that device category. If no device category is specified with the MIGRATIONLEVEL2 subparameter, the data sets migrate to the device category specified by the TAPEMIGRATION parameter of the SETSYS command.

Removing residual dump and backup VTOC copy data sets from migration level 1 volumes

During dump or backup (volume) processing, DFSMSHsm creates a data set (the VTOC copy data set), that contains selected information from the VTOC of the volume DFSMSHsm is dumping or backing up. These data sets are created on ML1 volumes. The VTOC copy data set drives the recovery of backed up data sets. Occasionally, VTOC copy data sets that are no longer needed remain as residual data sets on ML1 volumes. This happens when the primary volume that was dumped or backed up is removed from the system, and there is no longer a need to recover that volume.

To delete the VTOC copy data sets, use the DATA SET option from the ISMF primary option menu. This option can give you a list of the data sets associated with the primary volume that has been removed from the system. The DELETE command can then be used to remove the VTOC copy data sets.

The following search argument assumes that the SETSYS BACKUPPREFIX is (DFHSM) and that the volume serial number identifies the volume of interest. A data set name search argument of

```
'DFHSM.VTOC.*.%volser.*'
```

on the data set selection entry panel lists all the backup VTOC copy data sets. To find the dump VTOC copy data sets, change the second qualifier from VTOC to DUMPVTOC. Enter DELETE (on the data set list panel) in the line operator column next to the VTOC copy data sets you want to remove.

Preventing further migration to a DASD migration volume

Although the FREEVOL and MIGRATE commands can be used to empty migration volumes, the command does not prevent further use of the volumes. To prevent a migration volume from accepting any further migration copies, issue the following command:

```
ADDVOL volser UNIT(unittype) MIGRATION(DRAIN)
```

The DRAIN parameter prevents an ML2 volume from being associated with a key range, thus preventing the volume from being used for migration output. However, if an ML2 volume that has already been associated with a key range is given the DRAIN parameter in an ADDVOL command, that volume remains eligible to be used for migration output.

If you want to prevent such a volume from being used, you must first issue a DELVOL command with the UNASSIGN parameter; then issue an ADDVOL command with the DRAIN parameter. To determine which DASD ML2 volumes are associated with key ranges, use the QUERY MIGRATIONLEVEL2 command.

Moving data sets from DASD backup volumes

You can move all backup versions of SMS and non-SMS-managed data sets that meet a specified age criterion from DFSMSHsm daily backup volumes. Backup versions are moved from a specified daily backup volume to spill DASD or tape volumes. A parameter can be specified to prevent the movement of the most recent backup version for each data set.

Backup versions can be moved from a specific backup volume to spill tape volumes by using the following command:

```
FREEVOL BACKUPVOL(volser) TARGETLEVEL(SPILL(TAPE))
```

You can selectively move backup versions from a specific backup volume to the type of volumes set up by the SETSYS SPILL parameter where the AGE parameter is greater than zero. The command to move the data set is:

```
FREEVOL BACKUPVOL(volser) AGE(days) RETAINNEWEST
```


Removing volumes from DFSMSHsm management or control

You can use the DELVOL command to remove volumes that have been made known to DFSMSHsm by the ADDVOL command, or have automatically been made known to DFSMSHsm. Volumes are automatically ADDVOLed to DFSMSHsm in several ways; for example, an assignment of a scratch tape volume from DFSMSHsm management or control, or the use of an SMS-managed primary volume.

The recycle function and the DELVOL PURGE command are the only two ways to reuse a tape volume that has been written by DFSMSHsm.

When SMS storage groups are changed, use the DELVOL command to remove, from DFSMSHsm control, volumes that were removed from the SMS storage groups before daily automatic functions started. DELVOL these volumes on all DFSMSHsm systems that know about them. Do this so that the MVT entries will not be present.

Related reading

For more information, see Chapter 42, “DELVOL command: Removing a volume from DFSMSHsm management,” on page 841.

Removing dump copies and dump VTOC copy data sets from DFSMSHsm



Attention: If the dump copy you are removing is on a stacked dump tape and there are valid dump copies for other primary volumes on the tape, you cannot delete the dump tape (step 2) and you cannot delete the MCP record (step 3) without removing valid dump copies and losing data. The MCP record cannot be deleted until the dump tape expires.

Perform the following steps to discard dump copies associated with a volume that has been removed from the control of DFSMSHsm.

1. Obtain a list of the dump information associated with the DFSMSHsm-managed volume by issuing the following command. For *volser*, use the level 0 or ML1 volume that was dumped.

```
LIST VOLUME(volser) BCDS ALLDUMPS
```

2. Discard each dump copy by issuing the following command. For *volser*, use one of the dump volumes that is part of the dump copy. The PURGE, REASSIGN, UNASSIGN, and LASTCOPY parameters of the DELVOL DUMP command can be specified. You must specify COPYPOOLCOPY if the dump is part of a copy pool copy.

```
DELVOL volser DUMP
```

During DELVOL processing, when the last dump copy that requires the dump VTOC copy data set is deleted from the dump generation, the dump VTOC copy data set is deleted.

3. After you have removed all dump copies associated with a DFSMSHsm-managed volume, you can delete the MCP record. This BCDS record is deleted by issuing the following command. For *volser*, use the level 0 or ML1 volume that was dumped.

```
FIXCDS P volser DELETE
```

Note: The MCP record also contains information that is related to the backup VTOC copy data set. This data set drives the recovery of backed-up data sets. Therefore, do not delete the MCP record if you still want to recover the data that was backed up from the level 0 volume.

Related reading

- For more information about using the DELVOL command, see Chapter 42, “DELVOL command: Removing a volume from DFSMSHsm management,” on page 841.
- For more information about using the FIXCDS command, see [z/OS DFSMSHsm Diagnosis](#).

- For more information about using the LIST command, see [Chapter 53, “LIST command: Listing information from the MCDS, BCDS, and OCDS,”](#) on page 919.
- For information about how to delete VTOC copy data sets, see [“Removing residual dump and backup VTOC copy data sets from migration level 1 volumes”](#) on page 535.

JES3 considerations

These considerations apply only to non-SMS-managed volumes.

A DELVOL command for a volume is accepted at any time after initialization. However, use this command only when absolutely necessary. When you must delete a DFSMSHsm primary volume that JES3 setup might have reserved for use, leave the volume online if possible. This allows jobs to run that have already been set up to use the deleted volume as one of a pool. If the required data set is actually on another primary volume or if the data set is still migrated, the job can complete successfully. The volume is not selected for subsequent automatic recall of a data set because the volume has been deleted from DFSMSHsm.

Avoid the use of the DELVOL command unless you are certain that JES3 has not and will not associate or set up any migrated data sets for recall to the primary volume to be deleted. You cannot readily use the ADDVOL command to return the volume to DFSMSHsm's control because the ADDVOL command is processed only during initialization of DFSMSHsm in a JES3 environment.

An inadvertently used DELVOL command could prevent a job from running. If a migrated data set is associated with five or fewer pool volumes and one of the volumes is deleted from DFSMSHsm's control, the recall is directed to one of the remaining four volumes.

Consolidating valid data on tape volumes

As time goes by, data on your backup and ML2 tape volumes becomes invalid. Migrated data sets expire or are marked for deletion by DELETE and HDELETE commands. Backup versions roll off or are marked for deletion by BDELETE, HBDELETE, and EXPIREBV commands. To consolidate the valid data on fewer tapes, you can use the DFSMSHsm recycle process. You must perform two actions to use the recycle process:

1. You must define the parameters of the recycle process to DFSMSHsm. You perform the definition only once.
2. You must direct DFSMSHsm to begin the recycle process. You do this each time that you want recycle to run.

Defining the parameters of the recycle process

To define the parameters of the recycle process, you must specify the following information:

- The maximum percent of data remaining valid on a full (backup or ML2) or partial (nonassociated ML2) tape volume and its connected set before the volume is recycled
- The desired number of ML2 partial tape volumes not associated with any migration or recycle output task that remain after the recycle process
- The kind of unit used for output in recycling tape volumes
- Whether the recycle process waits for a tape allocation
- How many recycle tasks can run concurrently
- Whether to duplex recycle output. For ML2 tapes, specify SETSYS DUPLEX(MIGRATION(Y | N)). For backup tapes, specify SETSYS DUPLEX(BACKUP(Y | N)).

The SETSYS command is used for defining the default parameters of the recycle process. The command that is added to the ARCCMDxx member in each DFSMSHsm host for the example system is:


```
SETSYS ML2RECYCLEPERCENT(25) RECYCLEPERCENT(25) -
RECYCLEOUTPUT(BACKUP(ACL),MIGRATION(ACL)) -
MAXRECYCLETASKS(3) ML2PARTIALSNOTASSOCIATEDGOAL(8)
```

The following discussions of the SETSYS parameters related to recycle processing use the term *connected set*. A connected set is a single tape or a group of 2 or more tapes that has the following properties:

- The first tape has no valid data set spanning to a previous tape. (A spanning data set is a data set that is contained on more than one tape, and can span from 2 to 254 tape volumes.)
- The last tape has no valid data set spanning to a successive tape.
- All tapes have either a valid data set spanning to their previous tape, to their successive tape, or to both.
- All tapes are of one function type (for instance, all migration tapes).
- All tapes can be read on the same tape drive (for instance, a 3480X 18-track drive with IDRC compaction).
- A valid data set is contained entirely in only one connected tape set.
- DFSMSHsm recycles only those that are 254 or fewer volumes long. For an ML2 partial tape to be considered, it must be the only tape in its connected set.

Input tape drives that are allocated for connected sets remain allocated until all input tapes have been processed.

ML2RECYCLEPERCENT and **RECYCLEPERCENT** control the maximum amount of data that is *valid* on ML2 volumes and backup volumes, respectively, when DFSMSHsm automatically selects the volume for processing during recycling. In the example system, when DFSMSHsm detects that it has invalidated enough data to bring the amount of valid data on a volume and its connected set to 25% or less, it writes an informational message in the command activity log. When the RECYCLE command is issued for the appropriate category of volume, DFSMSHsm automatically selects each connected set of that category that has 25% or less valid data and recycles the complete connected set.

Note: For reel tapes, the percentage is valid blocks divided by total blocks written to this tape. For cartridge tapes, the percentage is valid blocks divided by estimated reuse capacity. This reuse capacity is estimated as the average total blocks written for each cartridge type.

The percentages for tapes can be displayed using either RECYCLE DISPLAY or LIST TTOC commands. (If you use the RECYCLE DISPLAY command, volumes which have 0 percent valid will not be listed. When such a volume is recycled, it will not be mounted, thus the volume does not have to be retrieved to be recycled.)

ML2PARTIALSNOTASSOCIATEDGOAL controls the desired number of partial ML2 tape volumes that are available. The recycle process considers all the ML2 partial tape volumes in the HSMplex that are not associated as output for any migration or recycle task and are not connected to a predecessor tape. If the number of those volumes is greater than eight, the excess volumes over eight are recycled. The tapes with the least amount of valid data (and that meet the other recycling criteria specified for the installation) are selected for recycling.

RECYCLEOUTPUT restricts the category of output device used by DFSMSHsm for writing the tape volume that consolidates the data from the tape volumes being recycled. Because you want the output tapes mounted on the 3490 devices with the automatic cartridge loader feature, you use the esoteric name that was specified with the USERUNITTABLE parameter of the SETSYS command.

MAXRECYCLETASKS specifies the maximum number of tape processing subtasks that DFSMSHsm will run. The default value is 2, and this value can be dynamically changed even while RECYCLE is running by using the SETSYS MAXRECYCLETASKS(*nn*) command (the minimum number of tasks is 1, the maximum number is 15). In the example system, MAXRECYCLETASKS is set to 3 subtasks (6 tape drives involved).

Note:

1. Each subtask requires two tape drives. If duplexing, each task requires three tape drives. One drive for the input tape, one for the original tape, and one for the alternate tape.

- Each subtask, when dealing with single-file-format tape cartridges, keeps the input tape drive allocated for as long as possible (certain error conditions will cause the drive to be deallocated).

RECYCLETAPEALLOCATION specifies whether DFSMSHsm is to wait for tape allocation to complete or is to proceed to process other requests. You have already specified NOWAIT in the initial overall SETSYS command.

Related reading

For more information about how to periodically deallocate an input unit during recycle processing and the effect of the SETSYS parameter RECYCLEINPUTDEALLOCFREQUENCY, see [“RECYCLEINPUTDEALLOCFREQUENCY: Allowing periodic deallocation of input unit during recycle processing” on page 1174.](#)

Initiating recycle processing

You can direct DFSMSHsm to recycle a single connected set or to recycle entire categories of volumes. Some commands that recycle tape volumes are:

```
RECYCLE ALL EXECUTE
RECYCLE ALL EXECUTE LIMIT(nnnn)
RECYCLE ALL EXECUTE PERCENTVALID(pct)
RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(include_range_list))
    - EXCLUDE(RANGE(exclude_range_list)))
RECYCLE ML2 EXECUTE
RECYCLE ML2 EXECUTE PERCENTVALID(pct)
RECYCLE ML2 VERIFY TAPELIST(FULLDSNAME(tapelist_dsn))
RECYCLE DAILY EXECUTE
RECYCLE DAILY EXECUTE PERCENTVALID(pct)
RECYCLE DAILY EXECUTE PERCENTVALID(pct) CHECKFIRST(N)
RECYCLE DAILY(day) EXECUTE
RECYCLE DAILY(day) EXECUTE PERCENTVALID(pct)
RECYCLE BACKUP EXECUTE
RECYCLE BACKUP DISPLAY TAPELIST (PREFIX(tapelist_prefix))
RECYCLE SPILL EXECUTE
RECYCLE SPILL EXECUTE PERCENTVALID(pct)
RECYCLE VOLUME(volser) EXECUTE
RECYCLE ALL OUTDATASET(dsname) EXECUTE
```

The BACKUP parameter allows both daily and spill backup tapes to be selected for input.

The EXECUTE parameter causes DFSMSHsm to perform the recycle function of moving valid data sets onto other tape volumes.

The LIMIT parameter quiescs the recycle process after the specified number of tapes are returned to scratch. DFSMSHsm calculates the total number of tapes returned by subtracting the number of output tapes used from the number of input tapes freed.

The SELECT parameter adds increased granularity to the RECYCLE eligibility tests. The *include_range_list* and *exclude_range_list* values help determine which tape volumes are candidates to be recycled. These values may be useful whenever you are addressing tape media conversion issues.

When a large number of tapes need to be manually pulled from shelves, the optional `TAPELIST` parameter produces a list of those tapes. The list is composed of pull groups, which are subsets of tapes, with each group in volume-serial sequence. The pull groups are listed in the sequence in which recycle processing will most likely request them. This projected sequence may differ from the actual processing sequence due to other factors affecting recycle processing, including, but not limited to: running more than one recycle task, unrecoverable errors, a mixture of device types or connected sets containing more than five volumes.

An alternative `VERIFY` parameter specifies the same lists as `EXECUTE TAPELIST`, but does not move any data sets. An alternative `DISPLAY` parameter specifies a single list in volume-serial sequence, but does not move any data sets.

The `OUTDATASET` parameter allows you to send `RECYCLE` messages and, if specified, the `DISPLAY` output to an output DASD data set rather than to `SYSOUT`.

The `PREFIX` and `FULLDSNAME` parameters allow you to send the tape list output to a DASD data set, rather than to `SYSOUT`.

For the `RECYCLE VOLUME` form of the command, `DFSMSHsm` does not check the volume to see whether the amount of valid data is equal to or less than the percent of valid data or any other selection criteria specified in either the `RECYCLE` command or the `SETSYS` command; `DFSMSHsm` recycles the volume's connected set, no matter how much of the data on the volume is invalid.

For the forms of the command using recycle categories of volumes (or all `DFSMSHsm`-owned tape volumes), `DFSMSHsm` automatically checks each connected set to determine whether it is eligible to be recycled. A volume is eligible for recycling only if the following conditions are met:

- The tape volume is an ML2 tape or a backup tape, but not an unassigned backup tape.
- A backup tape volume is marked full, or an ML2 volume is either marked full or is a partial tape not associated with any migration or recycle task and not connected to a predecessor tape.
- The percentage of valid data on the volume's connected set is equal to or less than the applicable value. This percentage is specified with the `PERCENTVALID` parameter, the `SETSYS ML2RECYCLEPERCENT` parameter, or the `SETSYS RECYCLEPERCENT` parameter. When these parameters are not specified, the `DFSMSHsm` default is 20%.
- The `SELECT` criteria, if specified, must be met. That is, all tapes of a connected set must be within the `INCLUDE RANGE` and none in the `EXCLUDE RANGE`.
- The volume has not previously failed recycle processing, or if it has failed, now contains no valid data.
- The `TTOC` for the volume is complete.
- The volume does not have a disaster alternate volume that has been flagged as a disaster alternate volume.

Each volume in an eligible connected set containing data is mounted, and all valid data is transferred to other tape volumes. When the first volume has had all of its valid data transferred, the next volume processed from that category has all of its valid data transferred to the same target volume until the target volume becomes full.

`DFSMSHsm` recycles ML2 volumes to ML2 volumes. It recycles both daily backup volumes and spill backup volumes to spill backup volumes.

`DFSMSHsm` selects and maintains different target tape volumes for recycle, data set migration, volume backup, and volume migration processing. This allows you to run migration, backup, and recycle at the same time.

If you specify `PERCENTVALID(0)`, `DFSMSHsm` selects only those tapes that are already empty but have not yet been returned to scratch. You may want to use `PERCENTVALID(0)` at those times when tape operators are not available.

Note: Tape drives will not be allocated if you specify `PERCENTVALID(0)`.

PERCENTVALID calculation

The PERCENTVALID calculation determines the valid data that exists within an entire connected set; it doesn't determine valid data for an individual volume within that set (unless the connected set consists of a single volume).

In determining whether to recycle a connected set, DFSMSHsm by default requires the first tape within the connected set to meet the current PERCENTVALID specification before continuing with the calculation. To allow connected sets to be recycled sooner, and possibly free up more volumes to scratch or tape pools, you can bypass this first volume check by entering the RECYCLE command with the CHECKFIRST parameter specified as N (no). When you specify CHECKFIRST(N), DFSMSHsm calculates the percent valid of the entire connected set, regardless of how much valid data is on the first volume of the connect set.

Note that CHECKFIRST(N) causes recycle processing to always determine the average percentage valid for a connected set, which can slow recycle performance if your installation has many large connected sets. Also, if you enter the RECYCLE command with the VOLSER parameter, the system ignores the CHECKFIRST parameter.

Examples

Volser	% Valid	Predecessor	Successor
-----	-----	-----	-----
TAPEA	35	-NONE-	TAPEB
TAPEB	5	TAPEA	-NONE-

If RECYCLE ALL PERCENT(30) EX is issued, the TAPEA–TAPEB connected set will not be selected, since the first tape in the set does not meet the PERCENTVALID criterion and CHECKFIRST(N) was not specified.

Volser	% Valid	Predecessor	Successor
-----	-----	-----	-----
TAPEC	25	-NONE-	TAPED
TAPED	75	TAPEC	-NONE-

If RECYCLE ALL PERCENT(30) EX is issued, the TAPEC–TAPED connected set will not be selected. Even though the first tape in the set meets the PERCENTVALID criterion, the entire connected set does not meet the criterion.

Volser	% Valid	Predecessor	Successor
-----	-----	-----	-----
TAPEE	5	-NONE-	TAPEF
TAPEF	45	TAPEE	-NONE-

If RECYCLE ALL PERCENT(30) EX is issued, the TAPEE–TAPEF connected set will be selected: the first tape volume in the set meets the PERCENTVALID criterion, and the entire connected set meets the criterion.

Volser	% Valid	Predecessor	Successor
-----	-----	-----	-----
TAPEG	55	-NONE-	TAPEH
TAPEH	1	TAPEG	-NONE-

If RECYCLE ALL PERCENT(30) EX CHECKFIRST(N) is issued, the TAPEG–TAPEH connected set will be selected. Although the first tape, TAPEG, does not meet the PERCENTVALID criterion, CHECKFIRST(N) was specified, and the entire connected set meets the PERCENTVALID criterion.

Selecting tape volumes during recycling

When you recycle tape volumes, DFSMSHsm selects the target tape volume to receive the valid data sets. No other task can be using this tape volume. (See [Table 13 on page 138](#) for volume contention information.)

Making cartridge-type, single-file-format tapes from reel-type or multiple-file-format tapes

Many of DFSMSHsm's advanced tape operations can be performed only on cartridge-type, single-file-format tape volumes. If you have reel-type or cartridge-type, multiple-file-format, DFSMSHsm-owned volumes in your system, you can transfer the data to cartridge-type, single-file format tapes by using the recycle function. Ensure that your RECYCLEOUTPUT parameter specifies a unit type that uses the magnetic tape cartridge. Then you can issue RECYCLE commands either to recycle individual volumes or to recycle categories of volumes. If you recycle categories of volumes, you will want to use the PERCENTVALID parameter to specify a high percentage so that all of the volumes in the category are recycled.

Limiting data that is written to a cartridge-type device

When you want to direct data to a cartridge-type tape volume within a single-file-format tape environment, you can specify the percent full from 1 to 9999 on the SETSYS TAPEUTILIZATION(UNITTYPE(unit) PERCENTFULL(pct)) command. When the percent full is reached, DFSMSHsm forces end-of-volume function on that volume. This command limits the amount of data that is written on a cartridge-type tape volume, which ensures that the tape can be copied with the TAPECOPY command. This also ensures that the original and the alternate tapes are identical when they are duplexed.

If your installation is backing up or migrating many small data sets, you might not be making full use of newer high capacity tape volumes. Here, it is recommended that your installation use extended tape table of contents (TTOCs). Doing so allows DFSMSHsm to write as many as 1,060,000 data sets on a single migration tape or backup tape, if the data sets will fit. In previous releases of z/OS, this number was limited to 330,000 data sets.

To use extended TTOCs, you must:

- Define the offline control data set (OCDS) with a maximum record size of 6144 bytes
- Specify the use of extended TTOCs by entering the SETSYS command with the EXTENDEDTTTOC(Y) parameter specified.

For information about defining the OCDS, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Note:

1. When you specify SETSYS DUPLEX for either backup or migration, or both, and you specify the NOLIMIT parameter for your tapes, PERCENTFULL is automatically set to 97%. If you designate a specific percentage, such as 100, DFSMSHsm uses the percentage that you specify.
2. Other effects of specifying the NOLIMIT parameter with the TAPEUTILIZATION command are that only one buffer is used for recycle processing rather than three, and no action is taken to reduce data set tape volume spanning.
3. You can use the following table to determine the PERCENTFULL value on the SETSYS TAPEUTILIZATION(UNITTYPE(unit)PERCENTFULL(pct)) command.

Table 51. Cartridge Percentages, by Output Device and Length			
Output Device	IBM Standard Length Cartridge Percentages	IBM Enhanced Length Cartridge Percentages	Other Cartridge Length Percentages
3490E native attached drive	1–100	1–100	1–9999, when compared to the IBM 3490E cartridge
3590B native attached drive	1–100	1–100	1–9999, when compared to the IBM 3590 cartridge
3590B emulating 3490E	1–100	1–100	1–9999, when compared to the IBM 3590 cartridge

Table 51. Cartridge Percentages, by Output Device and Length (continued)			
Output Device	IBM Standard Length Cartridge Percentages	IBM Enhanced Length Cartridge Percentages	Other Cartridge Length Percentages
3590E emulating 3490E or 3590B	1–100	1–100	1–9999, when compared to the IBM 3590 cartridge
3590 Model H emulating a 3590 Model B or 3490E	1–100	1–100	1–9999, when compared to the IBM 3590 cartridge
3592 Model J emulating 3490E or 3590B	1–100	N/A	N/A
3592 Model E-05 emulating 3590B	1–100	N/A	N/A
3592 Model E-06 emulating 3590B	1–100	N/A	N/A

Note: If your environment consists of different technology 3590 tape devices configured as stand-alone (non-SMS tape), you must associate an esoteric name with each type of device. You also must request that DFSMSHsm use the esoteric names rather than the generic names. This also applies to a mixed 3490 environment.

4. If backup or migration is using a system-managed tape library, specify the LIBRARYBACKUP or LIBRARYMIGRATION parameters with the SETSYS TAPEUTILIZATION command.
5. Virtual tape systems should generally use a PERCENTFULL value of 97%, unless a bigger value is needed to account for virtual tapes larger than the nominal 400 MB standard capacity MEDIA1 or 800 MB enhanced capacity MEDIA2 tapes. In the case of the newer virtual tape systems (TS7700 Release 1.4 and above), where DFSMSHsm derives media capacity by checking the mounted virtual tape, DFSMSHsm allows a PERCENTFULL value up to 110%. Anything larger is reduced to 100%. For older virtual tape systems, where DFSMSHsm cannot dynamically determine virtual tape capacity, PERCENTFULL values larger than 110% are honored. For example, to use virtual tapes for backup and migration volumes, you could enter the following commands:

```
SETSYS TAPEUTILIZATION(LIBRARYBACKUP PERCENTFUL(97))
SETSYS TAPEUTILIZATION(LIBRARYMIGRATION PERCENTFUL(97))
```

Related reading

For more information about the LIBRARYMIGRATION parameters, see [“TAPEUTILIZATION: Specifying the desired tape utilization”](#) on page 1198.

Specifying capacity utilization

On a CAPACITYMODE-switchable 3590 or 3592 device that is emulating a 3490 device, you have an option when you specify tape capacity. You can specify either to fill tapes to their full capacity or to stop at a point compatible with non-CAPACITYMODE-switchable devices. You specify whether to fill tapes to their full capacity using the SETSYS TAPEUTILIZATION (UNITTYPE(unittype)...CAPACITYMODE(COMPATIBILITY|EXTENDED)) command.

Note: CDS backup, ABARS output, and DUMP tapes do not support this function because they do not have the same block limitation as the supported functions.

DFSMSHsm writes to 97% of the capacity of MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13 tapes unless you specify otherwise through the SETSYS

TAPEUTILIZATION command. DFSMSHsm uses the reported cartridge type on the physical device to determine the tape capacity.

Related reading

For more information about commands related to capacity utilization, see [Part 2, “DFSMSHsm Storage Administration Reference,”](#) on page 677.

Considerations for using the 3592 Model J tape drive

Here are some tips you might find helpful when using the 3592 Model J tape drive.

- If your installation has an excessive number of spanning data sets, consider specifying a larger value in the SETSYS TAPESPANSIZE command. A larger absolute value is needed to represent the same amount of unused capacity on a percentage basis when the tape has a larger total capacity. For example, if you allow 2% of unused tape to avoid tape spanning for a 3590-Hxx device using enhanced media, you specify a TAPESPANSIZE of 1200 MB. To allow 2% unused tape for a MEDIA5 tape on a 3592 Model J device (no performance scaling), you specify a TAPESPANSIZE of 6000 MB. All size calculations for scaled tapes are based upon the scaled size and not the unscaled size.
- If the speed of data access on MEDIA5 tape is more important than full use of capacity, consider using performance scaling. Performance scaling allows 60 GB of physical (and more logical) data on each tape and keeps all data sets closer together and closer to the initial tape load point. If you use performance scaling with the DFSMSHsm duplex function, ensure that the original tape and the alternate tape both use performance scaling. Similarly make sure that Tapecopy input tapes and output tapes have the same performance scaling attributes.

Note: Performance scaling is not available on MEDIA6, MEDIA7, or MEDIA8 tape cartridge media.

- If your installation is using MEDIA5 tapes with performance scaling, consider using MEDIA7 tapes for high performance functions. The available MEDIA5 tapes could then be used to their full capacity. DFSMSHsm recycle processing of 3592 tapes can take significantly longer than the recycle of smaller tapes because the amount of data moved at the same RECYCLEPERCENT can be much larger. Besides moving more data, the likelihood of a tape takeaway for recall processing increases with the number of data sets still remaining on the tape.

One option for controlling overall recycle run time is the LIMIT(nnnn) parameter of recycle. Recycle returns no more than the specified number of tapes to scratch during the current recycle run. Because recycle sorts the tapes based on the amount of valid data still on each volume, the tapes recycled require the least processing time.

Another option to consider is decreasing the ML2RECYCLEPERCENT parameter, the RECYCLEPERCENT parameter, or both. For example, if your installation uses performance scaling (60 GB) for ML2, and you would like to move no more than 6 GB of data when an ML2 tape is recycled, set ML2RECYCLEPERCENT(10). If your installation uses full capacity (300 GB) for backup tapes and you would like to move no more than 6 GB of data when a backup tape is recycled, set RECYCLEPERCENT(2). These examples assume that the ML2 and Backup tapes in the installation are filled to capacity because the calculations are based upon the average fullness of marked full tapes on your system (the reuse capacity.)

To determine how much data your current recycle threshold implies, use the reuse capacity associated with the tapes. The current recycle threshold percent multiplied by the reuse capacity gives the maximum amount of data on any of the tapes when they are recycled.

While lowering the recycle threshold reduces recycle processing time and decreases the number of times each tape must be recycled, it may also increase the overall number of tapes needed in your installation. Also, if you have a mix of ML2 or Backup tape capacities in need of recycle processing, you may want to recycle tapes with the RANGE parameter and use the appropriate recycle threshold for the tape capacities in the range.

In an SMS tape environment, and optionally in a non-SMS tape environment, the SMS data class construct can be used to select WORM tapes for ABACKUP processing. The output data set prefix

specified in the aggregate group definition can be used by the ACS routines to select a WORM data class. Set up the ACS routine and the output data set name to uniquely identify the ABARS output files that must go to WORM tape.

In a non-SMS tape environment, the default is to allow tape pooling to determine if ABARS data sets go to WORM or R/W media. Optionally, if the DEVSUP parameter ENFORCE_DC_MEDIA=ALLMEDIATY or ENFORCE_DC_MEDIA=MEDIA5PLUS is used, then the data class must request the appropriate media type for it to be successfully mounted.

- If your installation is running z/OS in a Sysplex environment, consider using the Common Recall Queue to optimize mounts of migration tapes.
- AUDIT MEDIACONTROLS for a FAILEDCREATE situation typically only needs to look at the last few files on a tape. If it is available for your system level, make sure Audit APAR OA04419 is applied.
- To fully use MEDIA5 tapes, unless you have a specific need to run in 22-bit FBID mode, operate 3592 Model J tape drives emulating device type 3490 in 32-bit FBID mode. MEDIA6, MEDIA7, and MEDIA8 tapes cannot be used on a 3592 Model J emulating device type 3490. The 3592 Model J tape drive always operates in 32-bit FBID mode when it is emulating device type 3590.
- If you decide to change the performance scaling attribute for a particular function and partially full MEDIA5 tapes already exist, mark full the partial MEDIA5 tapes for that function. This is necessary because OPEN processing carries over the performance scaling attribute from partial tapes to the next mounted empty tape as long as the tape is mounted under the same allocation. When the function runs the next time, it will need to mount an empty tape and the performance scaling change can take effect.

Considerations for using the 3592 Model E05 tape drive

Here are some tips you might find helpful when using DFSMSHsm with the 3592 Model E05 tape drive.

- In a non-SMS mixed tape hardware environment, where multiple types of tape hardware are used to emulate 3590 devices, it is recommended that you define unique esoterics for each type of tape hardware. This action is necessary to avoid mixing incompatible recording technologies. You can define an esoteric to DFSMSHsm through the SETSYS USERUNITTABLE command, for example:

```
SETSYS UUT(3592E05:3592E05 3590H:3590H)
```

With esoterics defined, you can then direct output to the desired set of drives through the SETSYS command, for example:

```
SETSYS BACKUP(TAPE(3592E05))
```

- If your installation has an excessive number of spanning data sets, consider specifying a larger value in the SETSYS TAPESPANSIZE command. A larger absolute value is needed to represent the same amount of unused capacity on a percentage basis when the tape has a larger total capacity. For example, if you allow 2% of unused tape to avoid tape spanning for a 3590-Hxx device using enhanced media, specify a TAPESPANSIZE of 1200 MB. To allow 2% unused tape for a MEDIA5 tape on a 3592 Model E05 tape drive (no performance scaling), specify a TAPESPANSIZE of 9999 MB. All size calculations for scaled tapes are based upon the scaled size and not the unscaled size.
- If the speed of data access on a MEDIA5 or MEDIA9 tape is more important than full use of capacity, consider using performance scaling. Performance scaling uses 20% of the physical capacity on each tape and keeps all data sets closer together and closer to the initial tape load point. If you use performance scaling with the DFSMSHsm duplex function, be sure the original tape and the alternate tape both use performance scaling. Similarly make sure that TAPECOPY input tapes and output tapes have the same performance scaling attributes.

Note: Performance scaling is not available on the following tape cartridge media: MEDIA6, MEDIA7, MEDIA8, or MEDIA10.

- If your installation is using MEDIA5 tapes with performance scaling, consider using MEDIA7 tapes for high performance functions. The available MEDIA5 tapes could then be used to their full capacity.

Consider performance segmentation as a compromise solution. Performance segmentation increases the performance of data sets in the first twenty percent of the tape's capacity, but also uses the remaining capacity as a slower access segment. Average performance for the tape is increased at the expense of losing a percentage of the MEDIA5 or MEDIA9 overall tape capacity. (You cannot determine which data sets reside in which segment.)

DFSMSHsm recycle processing of 3592 Model E05 tapes can take significantly longer than with smaller tapes because the amount of data moved at the same RECYCLEPERCENT can be much larger. Besides moving more data, the likelihood of a tape takeaway for recall processing increases with the number of data sets still remaining on the tape.

One option for controlling overall recycle run time is the LIMIT(*nnnn*) parameter of recycle. Recycle returns no more than the specified number of tapes to scratch during the current recycle run. Because recycle sorts the tapes based on the amount of valid data still on each volume, the tapes recycled require the least processing time.

Another option to consider is decreasing the ML2RECYCLEPERCENT parameter, the RECYCLEPERCENT parameter, or both. Assume, for example, that your installation uses MEDIA7 tape for ML2 and MEDIA5 tape for backup. If the EFMT1 format is used and you would like no more than 6 GB of data to be moved when an ML2 tape is recycled, set ML2RECYCLEPERCENT(10) because a MEDIA7 tape can hold 60 GB of data in EFMT1. If your installation uses full capacity for backup tapes and you want no more than 6 GB of data to be moved when a backup tape is recycled, set RECYCLEPERCENT(2) because a MEDIA5 tape can hold 300 GB of data in EFMT1. These examples assume that the ML2 and backup tapes in the installation are filled to capacity because the calculations are based upon the average fullness of marked full tapes on your system (the reuse capacity.)

To determine how much data your current recycle threshold implies, use the reuse capacity associated with the tapes. The current recycle threshold percent multiplied by the reuse capacity gives the maximum amount of data on any of the tapes when they are recycled.

While lowering the recycle threshold reduces recycle processing time and decreases the number of times each tape must be recycled, it may also increase the overall number of tapes needed in your installation. Also, if you have a mix of ML2 or backup tape capacities in need of recycle processing, you may want to recycle tapes with the RANGE parameter and use the appropriate recycle threshold for the tape capacities in the range.

In an SMS tape environment, and optionally in a non-SMS tape environment, the SMS data class construct can be used to select WORM tapes for ABACKUP processing. The output data set prefix specified in the aggregate group definition can be used by the ACS routines to select a WORM data class. Set up the ACS routine and the output data set name to uniquely identify the ABARS output files that must go to WORM tape.

In a non-SMS tape environment, the default is to allow tape pooling to determine if ABARS data sets go to WORM or R/W media. Optionally, if the DEVSUP parameter ENFORCE_DC_MEDIA=ALLMEDIATY or ENFORCE_DC_MEDIA=MEDIA5PLUS is used, the data class must request the appropriate media type for it to be successfully mounted.

- Consider using the Fast Subsequent Migration function to reduce the need to RECYCLE these high-capacity tapes.
- For a Sysplex environment, consider using the Common Recall Queue to optimize mounts of migration tapes.
- AUDIT MEDIACONTROLS for a FAILEDCREATE situation typically only needs to look at the last few files on a tape. If it is available for your system level, ensure that Audit APAR OA04419 is applied.
- The 3592 Model E05 tape drive is used in 3590 emulation mode only; never 3490.

Considerations for using the 3592 Model E06 tape drive

Here are some tips you might find helpful when using DFSMSHsm with the 3592 Model E06 tape drive.

- In a non-SMS mixed tape hardware environment, where multiple types of tape hardware are used to emulate 3590 devices, it is recommended that you define unique esoterics for each type of tape

hardware. This action is necessary to avoid mixing incompatible recording technologies. You can define an esoteric to DFSMSHsm through the SETSYS USERUNITTABLE command, for example:

```
SETSYS UUT(3592E06:3592E06 3590H:3590H)
```

With esoterics defined, you can then direct output to the desired set of drives through the SETSYS command, for example:

```
SETSYS BACKUP(TAPE(3592E06))
```

- If your installation has an excessive number of spanning data sets, consider specifying a larger value in the SETSYS TAPESPANSIZE command. A larger absolute value is needed to represent the same amount of unused capacity on a percentage basis when the tape has a larger total capacity. For example, if you allow 2% of unused tape to avoid tape spanning for a 3590-Hxx device using enhanced media, specify a TAPESPANSIZE of 1200 MB. To allow 2% unused tape for a MEDIA5 tape on a 3592 Model E06 tape drive (no performance scaling), specify a TAPESPANSIZE of 9999 MB. All size calculations for scaled tapes are based upon the scaled size and not the unscaled size.
- If the speed of data access on a MEDIA5 or MEDIA9 tape is more important than full use of capacity, consider using performance scaling. Performance scaling uses 20% of the physical capacity on each tape and keeps all data sets closer together and closer to the initial tape load point. If you use performance scaling with the DFSMSHsm duplex function, be sure the original tape and the alternate tape both use performance scaling. Similarly make sure that TAPECOPY input tapes and output tapes have the same performance scaling attributes.

Note: Performance scaling is not available on the following tape cartridge media: MEDIA6, MEDIA7, MEDIA8, or MEDIA10.

- If your installation is using MEDIA5 tapes with performance scaling, consider using MEDIA7 tapes for high performance functions. The available MEDIA5 tapes could then be used to their full capacity.

Consider performance segmentation as a compromise solution. Performance segmentation increases the performance of data sets in the first twenty percent of the tape's capacity, but also uses the remaining capacity as a slower access segment. Average performance for the tape is increased at the expense of losing a percentage of the MEDIA5 or MEDIA9 overall tape capacity. (You cannot determine which data sets reside in which segment.)

One option for controlling overall recycle run time is the LIMIT(*nnnn*) parameter of recycle. Recycle returns no more than the specified number of tapes to scratch during the current recycle run. Because recycle sorts the tapes based on the amount of valid data still on each volume, the tapes recycled require the least processing time.

Another option to consider is decreasing the ML2RECYCLEPERCENT parameter, the RECYCLEPERCENT parameter, or both. Assume, for example, that your installation uses MEDIA7 tape for ML2 and MEDIA5 tape for backup. If the EFMT1 format is used and you would like no more than 6 GB of data to be moved when an ML2 tape is recycled, set ML2RECYCLEPERCENT(10) because a MEDIA7 tape can hold 60 GB of data in EFMT1. If your installation uses full capacity for backup tapes and you want no more than 6 GB of data to be moved when a backup tape is recycled, set RECYCLEPERCENT(2) because a MEDIA5 tape can hold 300 GB of data in EFMT1. These examples assume that the ML2 and backup tapes in the installation are filled to capacity because the calculations are based upon the average fullness of marked full tapes on your system (the reuse capacity.)

To determine how much data your current recycle threshold implies, use the reuse capacity associated with the tapes. The current recycle threshold percent multiplied by the reuse capacity gives the maximum amount of data on any of the tapes when they are recycled.

While lowering the recycle threshold reduces recycle processing time and decreases the number of times each tape must be recycled, it may also increase the overall number of tapes needed in your installation. Also, if you have a mix of ML2 or backup tape capacities in need of recycle processing, you may want to recycle tapes with the RANGE parameter and use the appropriate recycle threshold for the tape capacities in the range.

In a non-SMS tape environment, the default is to allow tape pooling to determine if ABARS data sets go to WORM or R/W media. Optionally, if the DEVSUP parameter ENFORCE_DC_MEDIA=ALLMEDIATY or ENFORCE_DC_MEDIA=MEDIA5PLUS is used, the data class must request the appropriate media type for it to be successfully mounted.

- DFSMSShsm recycle processing of 3592 Model E06 tapes can take significantly longer than with smaller tapes because the amount of data moved at the same RECYCLEPERCENT can be much larger. Besides moving more data, the likelihood of a tape takeaway for recall processing increases with the number of data sets still remaining on the tape.
- In an SMS tape environment, and optionally in a non-SMS tape environment, the SMS data class construct can be used to select WORM tapes for ABACKUP processing. The output data set prefix specified in the aggregate group definition can be used by the ACS routines to select a WORM data class. Set up the ACS routine and the output data set name to uniquely identify the ABARS output files that must go to WORM tape.
- Consider using the Fast Subsequent Migration function to reduce the need to RECYCLE these high-capacity tapes.
- For a Sysplex environment, consider using the Common Recall Queue to optimize mounts of migration tapes.
- The 3592 Model E06 tape drive is used in 3590 emulation mode only; never 3490.

Considerations for using the 3592 Model E07 tape drive

Here are some tips you might find helpful when using DFSMSShsm with the 3592 Model E07 tape drive.

- In a non-SMS mixed tape hardware environment, where multiple types of tape hardware are used to emulate 3590 devices, it is recommended that you define unique esoterics for each type of tape hardware. This action is necessary to avoid mixing incompatible recording technologies. You can define an esoteric to DFSMSShsm through the SETSYS USERUNITTABLE command, for example:

```
SETSYS UUT(3592E07:3592E07 3590H:3590H)
```

With esoterics defined, you can then direct output to the desired set of drives through the SETSYS command, for example:

```
SETSYS BACKUP(TAPE(3592E07))
```

- If your installation has an excessive number of spanning data sets, consider specifying a larger value in the SETSYS TAPESPANSIZE command. A larger absolute value is needed to represent the same amount of unused capacity on a percentage basis when the tape has a larger total capacity. For example, if you allow 2% of unused tape to avoid tape spanning for a 3590-Hxx device using enhanced media, specify a TAPESPANSIZE of 1200 MB. To allow 2% unused tape for a MEDIA11 tape on a 3592 Model E07 tape drive (no performance scaling), specify a TAPESPANSIZE of 9999 MB. All size calculations for scaled tapes are based upon the scaled size and not the unscaled size.
- If the speed of data access on a tape is more important than full use of capacity, consider using performance scaling, if the media you are using supports it. Performance scaling uses 20% of the physical capacity on each tape and keeps all data sets closer together and closer to the initial tape load point. If you use performance scaling with the DFSMSShsm duplex function, be sure the original tape and the alternate tape both use performance scaling. Similarly make sure that TAPECOPY input tapes and output tapes have the same performance scaling attributes.

Note: Performance scaling is not available on the following tape cartridge media: MEDIA6, MEDIA7, MEDIA8, MEDIA10, MEDIA12, and MEDIA13.

- If you are using MEDIA5, MEDIA9, or MEDIA11 tapes with performance scaling, consider using MEDIA7 or MEDIA13 tapes for high performance functions. The available MEDIA5, MEDIA9, or MEDIA11 tapes could then be used to their full capacity.

Consider performance segmentation as a compromise solution. Performance segmentation increases the performance of data sets in the first twenty percent of the tape's capacity, but also uses the

remaining capacity as a slower access segment. Average performance for the tape is increased at the expense of losing a percentage of the overall tape capacity. (You cannot determine which data sets reside in which segment.)

One option for controlling overall recycle run time is the `LIMIT(nnnn)` parameter of `recycle`. `Recycle` returns no more than the specified number of tapes to scratch during the current recycle run. Because `recycle` sorts the tapes based on the amount of valid data still on each volume, the tapes recycled require the least processing time.

Another option to consider is decreasing the `ML2RECYCLEPERCENT` parameter, the `RECYCLEPERCENT` parameter, or both. Assume, for example, that your installation uses lower capacity tapes for ML2 and higher capacity tapes for backup. If you want to move only 100 GB of data from each recycled tape, calculate the percent that 100 GB represents of each of the two tape media. Then set `RECYCLEPERCENT()` to the calculated percent 100 GB represents of the backup media and set `ML2RECYCLEPERCENT()` to the calculated percent 100 GB represents of the smaller migration tapes. These examples assume that the ML2 and backup tapes in the installation are filled to capacity because the calculations are based upon the average fullness of marked full tapes on your system (the reuse capacity.)

To determine how much data your current recycle threshold implies, use the reuse capacity associated with the tapes. The current recycle threshold percent multiplied by the reuse capacity gives the maximum amount of data on any of the tapes when they are recycled.

While lowering the recycle threshold reduces recycle processing time and decreases the number of times each tape must be recycled, it may also increase the overall number of tapes needed in your installation. Also, if you have a mix of ML2 or backup tape capacities in need of recycle processing, you may want to recycle tapes with the `RANGE` parameter and use the appropriate recycle threshold for the tape capacities in the range.

In a non-SMS tape environment, the default is to allow tape pooling to determine if ABARS data sets go to WORM or R/W media. Optionally, if the `DEVSUP` parameter `ENFORCE_DC_MEDIA=ALLMEDIATY` or `ENFORCE_DC_MEDIA=MEDIA5PLUS` is used, the data class must request the appropriate media type for it to be successfully mounted.

- DFSMSShm recycle processing of 3592 Model E07 tapes can take significantly longer than with smaller tapes because the amount of data moved at the same `RECYCLEPERCENT` can be much larger. Besides moving more data, the likelihood of a tape takeaway for recall processing increases with the number of data sets still remaining on the tape.
- In an SMS tape environment, and optionally in a non-SMS tape environment, the SMS data class construct can be used to select WORM tapes for `ABACKUP` processing. The output data set prefix specified in the aggregate group definition can be used by the ACS routines to select a WORM data class. Set up the ACS routine and the output data set name to uniquely identify the ABARS output files that must go to WORM tape.
- Consider using the Fast Subsequent Migration function to reduce the need to `RECYCLE` these high-capacity tapes.
- For a Sysplex environment, consider using the Common Recall Queue to optimize mounts of migration tapes.
- The 3592 Model E07 tape drive is used in 3590 emulation mode only; never 3490.

Limiting the number of ML2 and backup data sets that span tape volumes

The `TAPESPANSIZE` optional parameter of the `SETSYS` command can be used to reduce the occurrences of data sets spanning tape volumes. This is of benefit when running certain functions such as recall, recover, and recycle. When a data set spans tape volumes, a recall or recovery of that data set requires two tape mounts rather than one.

If you are recycling a connected set of tape volumes, recycle processing determines whether the average of all the tapes in the connected set meets the selection criteria and optionally whether the first volume

in the connected set also meets the selection criteria. If not, no tapes in the connected set are recycled, potentially leaving some tapes with very little data on them. You can modify your installation's use of recycling for connected sets through the CHECKFIRST parameter.

With the SETSYS TAPESPANSIZE command, you can specify a value between 0 and 4000. This value represents the maximum number of megabytes of tape that DFSMSHsm can leave unused while trying to eliminate spanning data sets. To state this differently, this value is the minimum size of a data set that is allowed to span tape volumes. Data sets whose size is less than the value do not normally span volumes. Only those data sets whose size is greater than or equal to the specified value are allowed to span volumes. The data written to tape results from compression by the tape subsystem, so predictions of that resultant size are not always correct. Therefore, spanning can occur for some data sets that do not compress as expected.

When data is being written to tape, DFSMSHsm estimates whether the space remaining on a tape can contain the entire data set, based on the following criteria:

- The limit that is specified with the SETSYS TAPEUTILIZATION command
- Whether the data set is already compressed on DASD
- The compression ratio that is assumed for your data sets
- Whether (for a logical 3490 device) the new data set would exceed the architected block count

If TAPEUTILIZATION specifies the NOLIMIT parameter, no action is taken to reduce the spanning of the data set tape volume.

Note: The use of the NOLIMIT parameter is not recommended.

Related reading

- For more information about tuning, see *z/OS DFSMSHsm Implementation and Customization Guide*.
- For more information about the spanning of the data set tape volume, see [“TAPESPANSIZE: Reducing tape volume spanning” on page 1197](#).

How to handle inconsistencies between tape media contents and OCDS TTOC records

When DFSMSHsm detects an inconsistency between the TTOC record and the content of the tape media, an ARC0378I message is issued during the DFSMSHsm startup to notify the system programmer that the missing TTOC record data set entries need to be resolved.

To get a list of all the volumes whose TTOC records are missing data set entries:

1. Issue a LIST TAPETABLEOFCONTENTS command with the SELECT(FAILEDCREATE) parameter. Inconsistent TTOC records for a failing DFSMSHsm host will not be listed until DFSMSHsm is restarted in that z/OS image because the records appear to still be in use.
2. Use the extended AUDIT MEDIACONTROLS function to reconstruct the missing TTOC data set entries for the indicated tape volume.

If data checks occur or invalid file block identifiers are encountered during AUDIT MEDIACONTROLS processing, a RECYCLE FORCE command can be issued to recycle the known valid data set entries on a tape volume. A warning message (ARC0378I) is issued, and the recycled tape volume is not deleted.

In general, you should *not* use the RECYCLE FORCE command to recycle a tape volume before using the AUDIT MEDIACONTROLS command to resolve the missing TTOC data set entries. If you do use it, you must *immediately* perform the following steps to avoid losing the valid data sets on the just-recycled tape volume:

- a. Issue a LIST DSN MCDS SELECT(VOLUME(volume serial number of recycled migration tape)) command, or a LIST DSN BCDS SELECT(VOLUME(volume serial number of recycled backup tape)) command to list the data set names of the data sets that DFSMSHsm indicates still reside on the tape volume.

- b. Recall or recover the indicated data sets that do not have TTOC data set entries associated with valid data sets residing on the tape volume.
- c. Issue a DELVOL command for the recycled tape volume.

When an ML2 tape volume cannot be deleted because an ARC0260I message indicates that valid data may exist on a volume, it is because the OCDS TTOC record indicates that the volume may contain valid migration copies. This condition occurs when an inconsistency exists between the TTOC record and the content of the tape media. Inconsistencies are created whenever TTOC creation or TTOC invalidation information is lost because of a DFSMSHsm cancel, a system failure due to a power outage, or an I/O error on the OCDS.

A TTOC creation inconsistency is present when the ARC0260I message is preceded by an ARC0378I message that relates to the same volume. TTOC creation inconsistencies are resolved by running the extended AUDIT MEDIACONTROLS function to reconstruct missing data set entries on the OCDS TTOC record that relates to the tape volume. After the extended AUDIT MEDIACONTROLS function is completed for the inconsistent tape volume, the normal generic RECYCLE function established by the installation will DELVOL the tape volume after all valid data is moved to another ML2 tape volume. If an immediate DELVOL of the tape volume is necessary, the single volume RECYCLE function will DELVOL the tape volume after all valid data is moved to another ML2 tape volume.

A TTOC invalidation inconsistency is present when the ARC0260I message is not preceded by an ARC0378I message that relates to the same volume. TTOC invalidation inconsistencies are resolved by the normal generic RECYCLE function established by the installation. If an immediate DELVOL of the tape volume is necessary, the single volume RECYCLE function can be used. Either way, the RECYCLE function will DELVOL the tape volume after all valid data is moved to another ML2 tape volume.

Recovering a damaged DFSMSHsm single-file migration level 2 volume

For information on recovering the undamaged remains of a single-file tape after it has been partially overwritten, see Chapter 60, “RECOVER command: Recovering a backup version or dump copy of a data set, or UNIX file, or a volume,” on page 997.

Making migration level 2 key range volumes available

When the ML2 volume for a specific key range becomes full during migration, DFSMSHsm tries to assign a new volume to the key range. If no ML2 volumes are available, you can do one of the following to make volumes available:

- Use the ADDVOL command to add more ML2 volumes to DFSMSHsm.
- Use the UNASSIGN parameter of the DELVOL command for each ML2 volume you want to be made available for reuse. When you use this parameter, specify DASD ML2 volumes that have been used before and that you want to reuse now. Ensure that space is available on these volumes because of subsequent recalls and deletions.
- Use a DEFINE command to assign a volume to a key range. To associate a specific volume with a key range, use the DEFINE command. Issue a QUERY ML2 command to determine the current association of volumes to key ranges. Then enter a DEFINE command specifying the same set of volumes but with the new volume replacing the old volume associated with the key range.

These three commands (DELVOL, ADDVOL, and DEFINE) allow you to reuse the current level 2 volumes.

Recovering a migrated data set

A restriction was noted in the discussion on recovering backup versions of data sets: DFSMSHsm cannot recover a data set to a migration volume.

Note: DFSMSHsm can recover a non-VSAM data set that is currently marked as migrated in the computing system catalog if a recover data set name is issued with NEWNAME specified, and the NEWNAME data set is not a migrated data set.

To recover a backup version of a migrated data set, use one of the following procedures:

- Recall the migrated data set and recover a backup version of that data set. You can use the REPLACE parameter if necessary.
- Delete the migrated data set and recover a backup version of the data set you just deleted.
- Issue a recover data set name with NEWNAME(*newdsname*) specified. If the NEWNAME(*newdsname*) is not a migrated data set, it will successfully recover.

Recovering a migration volume

You can use the RECOVER command to recover a data set to a level 0 volume. The RECOVER command recovers a data set in its original format. Because the migrated copy of a data set is stored in a different format from the original user's data set, you cannot use the RECOVER command to recover a data set to a migration volume. Depending on whether you use DFSMSHsm commands to periodically dump your DASD migration volumes, you can use one of the following procedures to recover an entire migration volume.

Note: These procedures assume that you have been using DFSMSHsm to back up data sets because the procedures recover the latest backup version of each data set that was on the migration volume.

Be aware that the latest backup version of each data set might not be identical to the data set that DFSMSHsm migrated. For example, the data set might have needed backup when DFSMSHsm migrated it, but DFSMSHsm did not have a chance to back up the migrated data set before the migration volume became inaccessible.

If you use DFSMSHsm's BACKVOL command to periodically dump your DASD ML2 volumes and use the automatic dump facility to periodically dump your ML1 volumes, perform [“Procedure 1” on page 552](#) to recover the data, mostly to a DASD migration volume.

If you do not dump your DASD migration volumes or need to recover the data sets that were on a tape migration volume and have no alternate tape, perform [“Procedure 2” on page 553](#) to recover all the data to primary volumes. For a discussion of alternate volumes, see [Chapter 9, “Making disaster backup copies of DFSMSHsm-owned tape volumes,” on page 435](#).

Procedure 1

1. To ensure that DFSMSHsm does not select the volume being recovered as a target volume for migration, issue the DFSMSHsm ADDVOL command with the DRAIN parameter. Issue the command in all DFSMSHsm hosts.
2. Restore the latest dump copy of the volume. This restores the DASD migration volume to its level when the dump copy was created. Issue the following DFSMSHsm command to request a full volume restore of the volume.

```
RECOVER * TOVOLUME(valid) UNIT(unittype) FROMDUMP
```

3. If the migration volume contains SDSP data sets and you have your own procedures to back up these data sets more frequently than you dump the volume, recover the most recent backup of the SDSP data set to the migration volume being recovered.
4. After the volume has been successfully restored, issue the FREEVOL command. This leaves data on the migration volume that is no longer valid, migrated data. It also enables you to easily obtain a list of the data sets that still need to be recovered (those that the MCDS indicates are still on the migration volume). The FREEVOL command process is driven off of the volume table of contents (VTOC) and includes all data sets in an SDSP data set on the volume that are currently migrated. Use the AGE(0) parameter to indicate that all data sets must be migrated. You can specify either ML1 or ML2 as the target migration level when you want to process a ML1 volume. Use the following command as an example:


```
FREEVOL MIGRATIONVOLUME(volser) TARGETLEVEL(MIGRATIONLEVEL1) AGE(0)
```

- Request from DFSMSHsm a list of all the migrated data sets that are indicated as being on the volume being recovered. These data sets represent the data sets that were migrated to the volume after the most recent dump copy was created. The migration copies of the data sets have been lost, so the data sets must be recovered from backup versions. Use the following LIST command to request this list of data sets:

```
LIST DATASETNAME MIGRATIONCONTROLDATASET -  
SELECT(VOLUME(volser)) OUTDATASET(dsname)
```

The OUTDATASET parameter is highly recommended so that the output from the LIST command is written to the data set whose name is given by *dsname*. This allows you to use a text editor to modify the output to easily perform the remaining steps:

- Make one more copy of the output data set, *dsname*.
- For each data set in the output from “5” on page 553, perform steps “8” on page 553 and “9” on page 553. You cannot perform these steps concurrently for the same data set. By using two copies of the list of data sets from step “5” on page 553, you can modify each of the two lists to perform steps “8” on page 553 and “9” on page 553 for all of the data sets at one time. Use a text editor and its global change command to add and remove the necessary text around each data set name.
- Delete the migrated data set. This uncatalogs the data set and deletes the MCDS data set record and any associated MCL, MCA, and MCO records. Do this with the following DFSMSHsm command:

```
DELETE dsname
```

- Recover the latest backup version for each data set. Do this with the following DFSMSHsm command:

```
RECOVER dsname REPLACE
```

Note: The backup copy that is recovered may not be the most current version of a data set if the data set was migrated to the ML1 volume before it was backed up.

- If the recovered volume is an ML1 volume, there might be backup versions that were restored to the migration volume. Some of these may represent valid backup versions. These can exist if data set backup requests were processed between the time automatic backup last ran on the primary DFSMSHsm host before the volume was dumped and the time when the volume was dumped. You can use the following procedure to assist you in recovering the backup versions, see “[Recovering backup versions from a damaged migration level 1 volume](#)” on page 554.
- After all valid, DFSMSHsm-owned data sets have been removed from the volume, you can reinitialize the volume. DFSMSHsm no longer requires access to it. You can also remove the volume from DFSMSHsm’s control with the DELVOL command.

The data sets will probably migrate the next time automatic primary space management occurs on the DFSMSHsm-managed volumes. This happens because when DFSMSHsm recovers a data set, the data set VTOC entry is restored exactly as it was when the backup version being recovered was created. DFSMSHsm does not reset the date-last-referenced field to the current date when it recovers the data set unless you rename the data set. Therefore, the recovered data sets will probably have a high inactive age.

Procedure 2

Rule: Do not start any step until the previous steps have completed.

- Determine which data sets were on the migration volume. To do this, use the following command:

```
LIST DSN MCDS SELECT(VOLUME(volser)) -  
OUTDATASET(dsname)
```

For *volser*, specify the volume serial number of the migration volume. DFSMSHsm writes the list in the data set whose name is *dsname*. You can issue this command from a console, terminal, or batch job.

2. Do a LISTCAT command if you want to save information about the data sets that you are going to delete in the next step. You may want to know, for example, whether the status of a generation data set (GDS) is DEFERRED or ACTIVE before it is deleted, because after being deleted it is always recovered with a status of DEFERRED.
3. Delete the MCDS data set record. This uncatalogs the data set and deletes the MCDS data set record and any associated MCL, MCA, and MCO records. To do this, use the following command for each data set on the migration volume:

```
DELETE dsname
```

When you use a text editor such as ISPF, you can edit the data set produced by the LIST command (step 1) so it contains one line for each data set. Each line contains the data set name. Replicate this data set once because you will need a copy for the next step. Then, use a global change command to precede each data set name with the characters HSEND CMD DELETE. A DFSMSHsm-authorized user can then run the data set as a command procedure (CLIST). However, if you have many data sets, include the necessary JCL in the data set and submit the data set as a batch job.

4. Recover the latest backup version for each data set that migrated to the migration volume. To do this, issue the following DFSMSHsm command:

```
RECOVER dsname
```

When you use a text editor such as ISPF, you can edit the copy of the data set containing the name of each data set on a separate line. Then, use a global change command to precede each data set name with the characters HSEND CMD RECOVER. A DFSMSHsm-authorized user can then run the data set as a command procedure (CLIST). If you have many data sets, include the necessary JCL in the data set and submit the data set as a batch job.

5. Issue the IDCAMS ALTER ROLLIN command if you want to change the status of a recovered GDS from DEFERRED to ACTIVE.

The catalog status of a recovered GDS is dependent upon its catalog status before being recovered. If the GDS is not cataloged before recovery, its catalog status after recovery is DEFERRED. If the GDS is cataloged before recovery, its catalog status after recovery remains the same, with the following exception: when DFSMSHsm is used as a data mover to recover a rolled-off GDS, the cataloged status of the GDS, after it is recovered, is DEFERRED.

6. Follow the procedure titled [“Recovering backup versions from a damaged migration level 1 volume”](#) on [page 554](#) if the volume being recovered is an ML1 volume and there are backup versions that must be recovered or replaced.

The data sets will probably migrate the next time automatic space management occurs on the DFSMSHsm-managed volumes. This happens because when DFSMSHsm recovers a data set, the data set VTOC entry is restored exactly as it was when the backup version being recovered was created. DFSMSHsm does not reset the date-last-referenced field to the current date when it recovers the data set. Therefore, the recovered data sets will probably have a high inactive age.

Related reading

For more information about the LISTCAT command, see [z/OS DFSMS Access Method Services Commands](#).

Recovering backup versions from a damaged migration level 1 volume

Use the following steps to recover backup versions from a damaged migration level 1 volume.

1. Issue the following Access Method Services command to print a list of MCM records. Use this list to determine if any of the backup versions on a damaged or restored ML1 volume are still valid.


```
PRINT IDS(bcds-dsname) FROMKEY(X'26') -
TOKEY(X'26') OUTFILE(ddname)
```

For *bcds-dsname*, substitute the fully qualified data set name of the DFSMSHsm backup control data set. For *ddname*, substitute the DDname of a preallocated data set that contains the output from the PRINT command.

2. Examine each MCM record that was printed to see if it represents a backup version that resides on the migration volume being recovered. To do this, look at offset 152 (X'98') in the record for the ML1 volume serial number. For all MCM records that have the volume serial number of the volume being recovered, you should perform the following. For *dsname* in the following commands, use the original data set name found at offset 64 (X'40') of the MCM record.
 - a. Use the following DFSMSHsm command to find out the version number of the backup version indicated as being on the volume being recovered:

```
LIST DATASETNAME(dsname) BACKUPCONTROLDATASET
```

- b. Using the version number found in the previous step, issue the following DFSMSHsm command to delete that backup version:

```
BDELETE dsname VERSIONS(bvn)
```

- c. Issue the following DFSMSHsm command to create a replacement backup version of the data set:

```
BACKDS dsname
```

An optional first step might be appropriate if the number of backup versions on the migration volume is large or if the original data sets no longer exist from which to create new backup versions. The movement of the backup versions phase of automatic backup, run in the primary DFSMSHsm host, moves all of the valid backup versions to backup volumes. Allowing automatic backup to run first moves all backup versions off all ML1 volumes so that only the MCM records for the damaged volume must be examined. The remainder of the procedure must still be followed for lost backup versions. If the migration volume was dumped, backup versions might have been lost if placed on the migration volume after the dump copy was created. If the migration volume was not restored from a dump copy, backup versions could still be lost if they resided on the parts of the volume that were damaged.

Returning dump tapes to use

Dump tapes in dump classes with the NOAUTOREUSE parameter specified are not automatically returned to usability when their generation is rolled off or expires. To return these tapes to use:

1. Physically return the tape volumes to the computer site.
2. Issue the following command for each tape volume:

```
DELVOL volser DUMP(REASSIGN)
```

3. Determine if the volumes are ready for reuse. If you have specified the TAPEDELETION(HSMTAPE) parameter of the SETSYS command, the volumes are ready for reuse. If you have specified the TAPEDELETION(SCRATCH) parameter, you might be required to reinitialize the volumes before they are returned to the scratch pool.

Managing SDSP data sets

Three procedures must be considered in managing SDSP data sets:

- Reorganizing the data sets
- Backing up and recovering the data sets
- Removing the data sets

Reorganizing SDSPs

Migration of user data sets into an SDSP causes VSAM CI and CA splits. Erasing the VSAM records of recalled and deleted data sets in an SDSP makes space reusable. However, these dynamics cause a redistribution of the free space in the SDSP. The SDSP will eventually appear to be full even though some free space may exist within the SDSP. When the high-used relative byte address equals the high-allocated relative byte address, the SDSP can appear to be full to a migration request if the SDSP cannot be extended. If DFSMSHsm fails in migrating to an SDSP, it issues a message to the migration activity log, and stops directing migrations to that SDSP.

For discussions of selecting SDSPs and the causes of out-of-space conditions, see [“SDSP selection”](#) on page 77 and [“Out-of-space condition during migration”](#) on page 77.

In this case, and after migration cleanup has run, you should consider reorganizing the SDSP. The frequency of reorganization depends on the size and number of SDSPs and on the number of data sets that regularly migrate to SDSPs in your environment. You should have enough SDSPs so that when one becomes full, the migrations occur to other SDSPs while you schedule and perform the SDSP reorganization. Use the TSO LISTC command once per week to determine the status of SDSPs as reported by the high-used and high-allocated relative byte addresses. After observing the SDSP status for several weeks, you can establish an appropriate frequency for SDSP reorganization in your environment. You may also choose to expand the SDSP, should you encounter an out-of-space condition. Alternatively, a reorganization of the SDSP can be automatically triggered when DFSMSHsm issues message ARC0542I to the console.

The format of the SDSP data set name is `uid.SMALLDS.Vvolser`. Where

- `uid` is a high-level qualifier that is the DFSMSHsm-authorized user ID from the startup procedure.
- `SMALLDS` and `V` are constants.
- `volser` is the volume serial number of the volume where the SDSP resides.

Note: An SDSP data set name is essentially a fixed name relating to a specific volume serial number in your environment and, once established, should not be changed.

You can reorganize the SDSP data set by using the Access Method Services EXPORT and IMPORT commands. Then, use the DFSMSHsm ADDVOL command to add the ML1 volume to DFSMSHsm again and specify the SMALLDATASET PACKING parameter to make the volume eligible for SDSP data set selection again.

Related reading

- For more information about the SDSP data set name is generated, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about the EXPORT and IMPORT commands and how to use them, see [z/OS DFSMS Access Method Services Commands](#).

SDSP backup and recovery

You should back up your SDSP data sets daily, such as at the end of automatic space management. The frequent backup is desirable because of the large number of user data sets whose data is stored in an SDSP data set. An SDSP data set is an active data set and is dynamically changing. The more frequently you back it up, the more current it will be if it must be recovered.

Example: If there are 50,000 data sets in an SDSP data set and it is lost, DFSMSHsm has to do 50,000 data set recovers. If you have backed up the SDSP, you can recover the SDSP from your latest backup and then recover only a few user data sets that have migrated to it since your latest backup of the SDSP.

DFSMSHsm does not automatically back up an SDSP data set as a single entity, but it can back up the individual data sets that the SDSP data set contains. However, you can use DFSMSHsm commands to back up and recover an SDSP data set. To back up an SDSP data set, issue the BACKDS command and include only the data set name of the SDSP.

To recover an SDSP that has been backed up by DFSMSHsm, do the following:

1. Issue a DELVOL MIGRATION command for the ML1 volume on which the SDSP resides. DFSMSHsm's recovery function does not recover backup versions to migration volumes, so the ML1 volume must be temporarily removed from DFSMSHsm's inventory of migration volumes.
2. Issue a RECOVER command for the SDSP data set.
3. Issue an ADDVOL command for the ML1 volume after the RECOVER command has been processed successfully. This adds the volume back to DFSMSHsm's inventory of migration volumes.

Removing SDSPs

There are two different methods of removing SDSPs: removing an entire ML1 volume that contains the SDSP or removing the SDSP from the ML1 volume that it resides on.

Removing a migration level 1 volume that contains an SDSP

The following steps remove all data sets from an ML1 volume that includes an SDSP:

1. Remove all migrated data sets from the ML1 volume using the FREEVOL command. The data sets will select a target volume using the same selection process DFSMSHsm uses during migration. Small data sets that move to another ML1 volume may go into an SDSP data set. The FREEVOL process cleans up data sets in the SDSP in need of scratching.
2. Verify that all valid data sets are out of the SDSP by using the following command:

```
LIST dsname SEL(VOL(ml1volser) SDSP) TERM
```

If there are data sets in the SDSP that do not move and are no longer wanted, try to delete them. It is important that you do not leave the control records for a data set in DFSMSHsm's system when you delete the SDSP.

3. Delete the SDSP data set using IDCAMS.
4. Check the ML1 volume to verify the SDSP and all user data sets have been moved off the volume. Use ISPF 3.4 or the LIST VTOC command to display the data set list for the ML1 volume. If the ML1 volume contains BACKDS data sets, automatic backup should be run on the primary DFSMSHsm host to move them to a backup tape.
5. DELVOL the ML1 volume from each DFSMSHsm host and remove the ADDVOL for the ML1 volume from each startup parmlib. The DELVOL command is issued as follows:

```
HSEND DELVOL ml1volser MIGRATION
```

The former ML1 volume is no longer associated with DFSMSHsm.

Removing an SDSP from a migration level 1 volume

The steps for removing only the SDSP from an ML1 volume are:

1. Code the ARCMEXT user exit to select only the data sets in the SDSP that will be migrated during the FREEVOL. Use the FREEVOL command to move all the valid data sets in the SDSP to another volume. The data sets in the SDSP will select a target volume using the same selection process DFSMSHsm uses during migration. Small data sets that move to another ML1 volume may go into an SDSP data set.
2. Verify that all valid data sets are out of the SDSP by using the following command:

```
LIST dsname SEL(VOL(ml1volser) SDSP) TERM
```

If there are data sets in the SDSP that do not move and are no longer wanted, try to delete them. It is important that you do not leave the control records for a data set in DFSMSHsm's system when you delete the SDSP.

3. Delete the SDSP data set using IDCAMS.

4. Use ISPF 3.4 or the LIST VTOC command to display the data set list for the ML1 volume. Check this list to verify the SDSP has been moved off the volume.
5. Issue the following ADDVOL NOSDSP command for the source ML1 volume. Update all startup parmlibs with an ADDVOL command for the ML1 volume to reflect the NOSDSP status.

```
ADDVOL ml1volser UNIT(unitttype) MIGRATION(ML1 NOSDSP)
```

6. Update the SETSYS SDSP parameter to reflect a NOSDSP status if you are removing all SDSPs from the DFSMSHsm system.

Protecting DFSMSHsm-owned tape volumes

You can protect DFSMSHsm-owned tape volumes with password protection, RACF protection, expiration date protection, or a combination of the three.

Related reading

For more information about protecting DFSMSHsm-owned tape volumes, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Converting to a new tape technology

If you are converting from an old tape technology to a new tape technology, identify whether the old tapes have the same logical device type as the new ones. Then mark the old tapes full so that they are no longer available for output consideration. With DFSMSHsm support of the 3590E device, you do not have to perform this step, because DFSMSHsm can distinguish 3592, 3590E, 3590B, 3590H, and VTS tapes from other type cartridges. However, DFSMSHsm does not differentiate between other hardware that has a common logical device type.

Use the following command to identify the old backup tapes:

```
LIST TTOC SELECT (BACKUP NOTFULL)
```

Use the following command to mark the old tapes as full:

```
DELVOL volser BACKUP(MARKFULL)
```

Requesting DFSMSHsm service from your applications

DFSMSHsm has macros available that allow you to request DFSMSHsm service from your application programs. You provide information to the macro and execute it from your application program. The macro builds the required DFSMSHsm control information and issues the request for DFSMSHsm service.

Related reading

For more information about the macros, see [z/OS DFSMSHsm Managing Your Own Data](#).

Chapter 14. Obtaining information from DFSMSHsm

In using DFSMSHsm, you may have occasion to require information about what is happening in the processing. DFSMSHsm provides extensive facilities for determining what the conditions are in the product. The DFSMSHsm log and the activity logs have already been mentioned. In addition to the logs, you can obtain the following from DFSMSHsm:

- Information from the control data sets
- Information about the present operating conditions in the DFSMSHsm address space
- Information about availability of output tape volumes
- Statistical information about DFSMSHsm operations
- A trace of DFSMSHsm's processing
- Information about the conditions in the system if a specific error occurs
- Determining the results of volume space management data set selection without actually moving data sets

Related reading

For more information about using DCOLLECT for reporting, see [*z/OS DFSMSdfp Storage Administration*](#) and [“Running new reports with report generator”](#) on page 575 for running new reports.

Obtaining information from the control data sets

Use the LIST command to obtain information from the control data sets. Information can be about:

- Backup volumes
- Copy pools
- Data sets
- UNIX files
- Dump classes
- Dump volumes
- DFSMSHsm host serialization
- DFSMSHsm-managed volumes
- DFSMSHsm-owned volumes
- Tape volumes
- User authorization

In obtaining the information, you can receive it in any of three ways: as a printed listing, as a DASD data set, or as terminal output. For small amounts of data, such as DFSMSHsm host serialization information or user-authorization information, terminal output is satisfactory. However, the LIST command is capable of producing very large amounts of information, for example, when you request information about all migration volumes. For such requests, you want to be sure to direct the output to either a printed listing or a DASD data set that you can edit or print later.

The parameters that control the destination for the LIST command output are the same for any request. In showing the forms of the command for each type of output, all other parameters (such as request) are being identified.

The form of the command for obtaining terminal output is:

```
LIST (request) TERMINAL
```


The form of the command for obtaining a printed listing is:

```
LIST (request) SYSOUT
LIST (request) SYSOUT(class)
```

The first form of the command directs the printed output to the SYSOUT class specified with the SETSYS command. The second form of the command specifies a SYSOUT class to which the output is directed.

The form of the command to direct the output to a DASD data set is:

```
LIST (request) OUTDATASET(dsname)
```

dsname is the fully qualified name of the data set to which you want the LIST command output sent. If the data set exists, DFSMSHsm places the report at the end of the existing data set. If the data set does not exist, DFSMSHsm dynamically allocates and catalogs an output data set.

Related reading

For more information about the LIST command, see [Chapter 53, “LIST command: Listing information from the MCDS, BCDS, and OCDS,” on page 919](#).

Obtaining backup volume information

The forms of the requests for obtaining backup volume information are:

```
LIST BACKUPVOLUME
LIST BACKUPVOLUME SELECT(EMPTY)
LIST BACKUPVOLUME(volser)
```

The first request lists the information about all the backup volumes in the system. The second request lists the information about all the empty backup volumes in the system. The second request can be used to determine which DFSMSHsm tapes can be placed in the cartridge loader. The third request lists the information about a specific backup volume.

Obtaining data set information

You can obtain information about a single data set or about every data set for which DFSMSHsm has any record. You can also select groups of data sets. Optional parameters allow you to select:

- Information from the backup control data set
- Information from the migration control data set
- Information from both the backup and the migration control data sets
- Information about data sets that have been recalled
- Information about data sets with a certain minimum age, maximum age, or between a minimum and a maximum age
- Information about data sets on ML1 volumes
- Information about data sets on ML2 volumes
- Information about the designated data sets on a specific migration volume
- A list of all of the migrated data sets that were empty at the time of migration
- Information about the data sets in SDSP data sets
- Information about the data sets not in SDSP data sets
- Information about VSAM data sets
- A summary of the information about the data sets

Many permutations exist for requesting information about data sets and levels. Some of them are shown here.

Sample requests for information about data sets are:

This request obtains information about all data sets that are recorded in the backup control data set.

```
LIST DATASETNAME BACKUPCONTROLDATASET
```

This request obtains information about the data set identified by *dsname* from the migration control data set.

```
LIST DATASETNAME(dsname) MIGRATIONCONTROLDATASET
```

This request obtains information about all data sets recorded in both the backup control data set and the migration control data set. In addition, it obtains information about any data sets that have migrated and been recalled.

```
LIST DATASETNAME BOTH INCLUDEPRIMARY
```

This request obtains information about data sets with the qualifier *levelid* recorded in the migration control data set, but only for the data sets on ML1 volumes.

```
LIST LEVEL(levelid) MIGRATIONCONTROLDATASET -  
SELECT(MIGRATIONLEVEL1)
```

This request obtains information about all data sets recorded in the migration control data set as being migrated to SDSP data sets.

```
LIST DATASETNAME MIGRATIONCONTROLDATASET -  
SELECT(SMALLDATASETPACKING)
```

This request obtains a list of all migrated data sets (excluding PDS and PDSE data sets) that were empty at the time of migration.

```
LIST DATASETNAME MIGRATIONCONTROLDATASET -  
SELECT(EMPTY)
```

Related reading

For more information about the LIST command, see [Chapter 53, “LIST command: Listing information from the MCDS, BCDS, and OCDS,” on page 919](#).

Obtaining UNIX file information

You can obtain information about a single UNIX file or about every UNIX file for which DFSMSHsm has any record. You can select subdirectories of UNIX files. Optional parameters allow you to select:

- Information from the backup control data set
- Information about UNIX files with a certain minimum age, maximum age, or between a minimum and a maximum age
- Information about the designated UNIX files on a specific volume

Many permutations exist for requesting information about UNIX files and directories. Some of them are shown here.

Sample requests for information about UNIX files are:

This request obtains information about all UNIX files that are recorded in the backup control data set.


```
LIST FILELEVEL('/ ') RECURSE BACKUPCONTROLDATASET
```

This request obtains information about the UNIX file identified by *pathname* from the backup control data set.

```
LIST FILENAME(pathname) BACKUPCONTROLDATASET
```

This request obtains information about UNIX files sets contained in directory */dir1/dir2* recorded in the backup control data set, but only for the backups on VOLSER MIG102.

```
LIST FILELEVEL('/dir1/dir2/') BACKUPCONTROLDATASET -  
SELECT(VOLUME(MIG102))
```

Related reading

For more information about the LIST command, see [Chapter 53, “LIST command: Listing information from the MCDS, BCDS, and OCDS,”](#) on page 919.

Obtaining copy pool information

You can obtain information about all copy pool fast replication backup versions by using the following command:

```
LIST COPYPPOOL
```

This request provides version information for each copy pool that has at least one attempted backup. The list of source and target volume pairs is not included as part of this list request. The list output consists of the following information:

- Copy pool name
- Version number
- Valid or invalid
- VTOCENQ
- Token
- Backup date and time
- Dump, DASD copy, or both

You can obtain information about a specific copy pool by using the following command:

```
LIST COPYPPOOL(cpname)
```

This request provides detailed version information for each valid or invalid backup version for the specified copy pool. The date and time fields are zero when the PREPARE subparameter is used with the FRBACKUP command. The list output consists of the following information:

- Copy pool name
- Version number
- Valid or invalid
- VTOCENQ
- Token
- Backup date and time (for Prepare versions these fields display a value of zero)
- Source and target volume pairs
- File sequence number
- Dump data set name

- Dump class availability
- Tape device type

You can use keywords FRVOLS, NOVOLS, DUMPVOLS, or ALLVOLS to specify what type of volume information is to be displayed.

- The FASTREPLICATIONVOLS keyword, the default value, indicates that the source/target pairs should be listed.
- The NOVOLS keyword doesn't list any volume pairs data, but lists the dump class information if there is a dump copy version.
- The DUMPVOLS keyword indicates that source and dump volser data for all dump versions should be listed.
- The ALLVOLS keyword indicates all volume pairs should be listed.

The GENERATION(*gennum*) keyword only applies to a situation when the ALLVOLS keyword is specified. The default (GENERATION(*gen0*)) indicates that the most recent DASD and dump versions are to be listed. The *gennum* parameter is required, it means only the indicated generation should be listed. ALLVERS indicates that all DASD and dump versions are to be listed. TOKEN(*token*) refers to a version.

The SELECT keyword provides more flexibility in terms of which records for the copy pools should be listed, FASTREPLICATIONSTATE or DUMPSTATE.

FRSTATE indicates information that is wanted based on the FRBACKUP DASD copy.

- RECOVERABLE indicates that the copy pool volumes can be recovered from the DASD version.
- NONRECOVERABLE indicates that the copy pool volumes cannot be recovered from the DASD version.
- FAILED indicates that the DASD version is NONRECOVERABLE due to a failure during the FRBACKUP or WITHDRAW was done before all volumes in the copy pool were successfully copied
- NONE indicates that a DASD copy does not exist.

DUMPSTATE indicates information that is wanted based on the FRBACKUP DUMP.

- ALLCOMPLETE indicates that all volumes have been successfully dumped to all dump classes specified.
- REQUIREDCOMPLETE indicates that all volumes have been successfully dumped to at least all REQUIRED dump classes.
- PARTIAL indicates that some volumes of a copy pool have been successfully dumped.
- NONE indicates that no dumps are associated with the copy pool.

You can obtain information of volume pairs of all copy pool volumes that are recoverable from the DASD version by using the following command:

```
LIST CP(CPA) ALLVOLS(ALLVERS) SELECT(FRSTATE(REC)) ODS(LIST.CPA)
```

You can obtain information about the volumes contained in a specific copy pool backup storage group by using the following command:

```
LIST COPYPOOLBACKUPSTORAGEGROUP(cpbsgname)
```

This request provides a list of each volume in the copy pool backup storage group that DFSMSHsm has selected as a target volume. Volumes that are defined to the storage group that are not listed have not been selected as target volumes. In addition, the name of the source volume that each target volume is a target for will be listed.

You can obtain information about which copy pool a volume is associated with by using the following command:

```
LIST PRIMARYVOLUME BCDS
```


This request provides information about the copy pools that a primary volume is associated with. Specify either the BCDS or BOTH parameter for this request to process correctly. Use the LIST COPYPOOL(*cpname*) command to determine which target volumes are associated with a particular source volume.

You can obtain information about which copy pools the volume is associated with by using the following command:

```
LIST PVOL BCDS
```

When you use the PVOL parameter, with either the BCDS or BOTH subparameters, DFSMSHsm returns information about which copy pools the volume is associated with. With ALLDUMPS or BCDS subparameters, should also consider copy pool dumps. Copy pool dumps versions should include the name of the source copy pool. If the volume is in a copy pool, associated information needs to be displayed. Target volumes of a copy pool source/target pair do not display information. To determine which target volumes are associated with a particular source volume, you can use the LIST COPYPOOL(*cpname*) command.

You can obtain information about whether the volume resides in a copy pool by using the following command:

```
LIST DVOL
```

You can obtain information about whether the dump class associated with the copy pool is required or optional by using the following command:

```
LIST DUMPCLASS
```

You can obtain information about the host holding a copy pool resource by using the following command:

```
LIST HOST
```

When you use the HOST parameter, DFSMSHsm lists the host holding a copy pool resource. If subparameter RESET is specified, DFSMSHsm resets associated serialization fields.

Obtaining dump class information

You can obtain information about either all dump classes or an individual dump class. The forms of the request for dump class information are:

```
LIST DUMPCLASS
```

```
LIST DUMPCLASS(class)
```

Obtaining dump volume information

You can request information about all dump volumes or specific dump volumes. In addition, you can select the volumes with specific characteristics. The forms of the requests for information about dump volumes are:


```

LIST DUMPVOLUME

LIST DUMPVOLUME -
  SELECT(AVAILABLE)

LIST DUMPVOLUME -
  SELECT(UNAVAILABLE)

LIST DUMPVOLUME -
  SELECT(EXPIRED)

LIST DUMPVOLUME -
  SELECT(UNEXPIRED)

LIST DUMPVOLUME -
  SELECT(NORETENTIONLIMIT)

LIST DUMPVOLUME -
  SELECT(DUMPCCLASS(class))

LIST DUMPVOLUME(volser)

```

The first form of the request obtains information about all the dump volumes in the system. The next six forms of the request obtain information about all dump volumes in the system that meet certain conditions. The eighth form obtains information about all dump volumes in a group of volumes made from the same DASD volume from one dump operation. The variable *volser* can be any of the dump volumes in the group.

You can determine the data sets that were on a particular volume when the volume was dumped. The commands to do this are:

```

LIST ML1(volser) BCDS ALLDUMPS

LIST PRIMARY(volser) BCDS ALLDUMPS

LIST DUMPVOLUME(volser) -
  CONTENTS

```

The first two commands give lists of the volumes to which the source volume has been dumped and the date of the dump. The third command lists the contents of the source volume when it has been dumped. You can specify any one of the dump volumes in the set of dump volumes for *volser* for a particular date.

If a volume is processed by both full-volume dump and incremental backup, DFSMSHsm maintains VTOC copy data sets for both operations. To determine the most up-to-date copy, issue the following command:

```

LIST PRIMARYVOLUME(volser) BCDS

```

If the full-volume dump is more recent than the backup, use the three preceding dump-related commands to obtain the latest VTOC copy. If the backup was more recent than the dump, use the LIST command for BACKUPCONTENTS as indicated in [“Obtaining DFSMSHsm-managed volume information” on page 568](#).

Obtaining DFSMSHsm host serialization information

As DFSMSHsm processes volumes and data sets, it obtains control of certain control data set records. While DFSMSHsm in one z/OS image has control of a control data set record, other hosts in the same or different z/OS image cannot use that control data set record. To determine which control data set records that a particular DFSMSHsm host has control of, issue the following command:

```

LIST HOST(hostid)

```

If one DFSMSHsm host in a multiple DFSMSHsm-host environment fails and you want it to relinquish control of all CDS records, issue the following command:

```

LIST HOST(hostid) RESET

```


LIST HOST and LIST HOST RESET also list the host holding a copy pool resource. When RESET is specified, associated serialization fields are reset.

Obtaining lists of backup and dump parameters and backup volume serial numbers

Note: In the following cases, full backup tapes are not listed after a FIXCDS REFRESH function has been performed.

To obtain a list of all backup and dump parameters and all non-full tape backup volumes and DASD backup volumes, issue the following command:

```
QUERY BACKUP(ALL)
```

To obtain a list of all backup parameters and all non-full daily tape backup volumes and DASD daily backup volumes, issue the following command:

```
QUERY BACKUP(DAILY)
```

To obtain a list of all backup parameters, all non-full daily tape backup volumes, and DASD daily backup volumes assigned to a particular day in the backup cycle, issue the following command:

```
QUERY BACKUP(DAILY(day))
```

To obtain a list of all backup parameters and all non-full spill tape backup volumes and DASD spill backup volumes, issue the following command:

```
QUERY BACKUP(SPILL)
```

To obtain a list of all backup parameters and all non-full tape unassigned backup volumes and DASD unassigned backup volumes, issue the following command:

```
QUERY BACKUP(UNASSIGNED)
```

Running new reports with the DFSMSrmm Report Generator

DFSMSHsm reports can be run with the DFSMSrmm Report Generator. The reports are based on FSR and WWFSR (SMF) data and DFSMSHsm control data set data obtained by DCOLLECT.

For HSM FSR records

```
For HSM FSR Records
  . ARCGAB01 ABARS ABACKUP Statistics
  . ARCGAR01 ABARS ARECOVER Statistics
  . ARCGS00A Statistics for Automatic Space Management
  . ARCGS001 Statistics for DFSMSHsm
  . ARCGS002 Statistics for Backup
  . ARCGS003 Statistics for Migration
  . ARCGS004 Statistics for Recall
  . ARCGS005 Statistics for Recovery
  . ARCGS006 Statistics for Volume Dump
  . ARCGS007 Statistics for Restore from Dump Copy
  . ARCGS008 Statistics for FRBACKUP
  . ARCGS009 Statistics for FRRECOVER
  . ARCGS010 DFSMSHsm Thrashing Report
  . ARCGS011 Statistics for Class Transition
```

```
...
...
```


Panel help

Panel Help

DFSMSrmm Report Definitions

Row 1 to 17 of 44

Command ==>

Scroll ==> PAGE

The following line commands are valid: A,D,G,H,J,L,M,N,S, and T

S Name	Report title	Report type	User id
ARCGAB01	ABARS ABACKUP Statistics	DFSMSHsm ABARS Report	HSM
ARCGAR01	ABARS ARECOVER Statistics	DFSMSHsm ABARS Report	HSM
ARCGDB01	DCOLLECT BACKUP DATA	DFSMSHsm DCOLLECT BACKUP	HSM
ARCGDD01	DCOLLECT DASD CAPACITY PLANNIN	DFSMSHsm DCOLLECT DASD CAP	HSM
ARCGDM01	DCOLLECT MIGRATION DATA	DFSMSHsm DCOLLECT MIGRATION	HSM
ARCGDT01	DCOLLECT TAPE CAPACITY PLANNIN	DFSMSHsm DCOLLECT TAPE CAP	HSM
ARCGS00A	Statistics for Automatic Space	DFSMSHsm FSR-SMF Records	HSM
ARCGS001	Statistics for DFSMSHsm	DFSMSHsm FSR-SMF Records	HSM
ARCGS002	Statistics for Backup	DFSMSHsm FSR-SMF Records	HSM
ARCGS003	Statistics for Migration	DFSMSHsm FSR-SMF Records	HSM
ARCGS004	Statistics for Recall	DFSMSHsm FSR-SMF Records	HSM
ARCGS005	Statistics for Recovery	DFSMSHsm FSR-SMF Records	HSM
ARCGS006	Statistics for Volume Dump	DFSMSHsm FSR-SMF Records	HSM
ARCGS007	Statistics for Restore from Du	DFSMSHsm FSR-SMF Records	HSM
ARCGS008	Statistics for FRBACKUP	DFSMSHsm FSR-SMF Records	HSM
ARCGS009	Statistics for FRRecover	DFSMSHsm FSR-SMF Records	HSM
ARCGS010	DFSMSHsm Thrashing Report	DFSMSHsm FSR-SMF Records	HSM

Obtaining information about migration and DFSMSHsm-managed volumes

To obtain information about migration and DFSMSHsm-managed volumes, you can select the following types of information:

- All migration volumes
- ML1 volumes
- All ML2 volumes
- ML2 DASD volumes
- ML2 tape volumes
- All migration and DFSMSHsm-managed volumes
- A particular migration or DFSMSHsm-managed volume

You also can select whether the information is to be taken from the backup control data set, the migration control data set, or both, and whether to select only empty volumes.

Again, because of the many possible forms for this request, only a sample is presented.

To request information about all migration volumes from both the migration control data set and the backup control data set, issue the following command:

```
LIST MIGRATIONVOLUME BOTH
```

To request information about all ML1 volumes from the backup control data set, issue the following command:

```
LIST MIGRATIONLEVEL1 BACKUPCONTROLDATASET
```

To request information about empty ML2 volumes from the migration control data set, issue the following command. You can use this command to obtain a list of tapes that can be inserted in the cartridge loader:

```
LIST MIGRATIONLEVEL2(TAPE) -  
MIGRATIONCONTROLDATASET SELECT(EMPTY)
```


To request information about ML2 DASD volumes from both control data sets, issue the following command:

```
LIST MIGRATIONLEVEL2(DASD) BOTH
```

To obtain information about ML2 tape volumes from the migration control data set, issue the following command:

```
LIST MIGRATIONLEVEL2(TAPE) MIGRATIONCONTROLDATASET
```

To obtain information about all DFSMSHsm-managed and migration volumes from both control data sets, issue the following command:

```
LIST VOLUME BOTH
```

To obtain information about a particular volume from the backup control data set, issue the following command:

```
LIST VOLUME(volser) BACKUPCONTROLDATASET
```

Obtaining DFSMSHsm-managed volume information

You can also obtain information about the contents of a particular level 0 volume when the latest or next-to-latest incremental backup was performed. The information can be obtained from the backup control data set, the migration control data set, or both. For information about the incremental backup, you can select information about multiple-volume or VSAM data sets.

To obtain information about the contents of a level 0 volume when the latest incremental backup was performed, issue the following command:

```
LIST PRIMARYVOLUME(volser) BCDS BACKUPCONTENTS(0)
```

If a volume is processed by both full-volume dump and incremental backup, DFSMSHsm maintains VTOC copy data sets for both operations. To determine the most up-to-date copy, issue the following command:

```
LIST PRIMARYVOLUME(volser) BCDS
```

If the backup was more recent than the full-volume dump, use the preceding backup-related command to obtain the latest VTOC copy. If the dump was more recent than the backup, issue the following command:

```
LIST DUMPVOLUME(volser) CONTENTS
```

To obtain information about a DFSMSHsm-managed volume that is contained in the migration control data set, issue the following command:

```
LIST PRIMARYVOLUME(volser) MIGRATIONCONTROLDATASET
```

To obtain information about the non-VSAM multiple-volume data sets that were on a level 0 volume at the next-to-latest incremental backup, issue the following command:

```
LIST PRIMARYVOLUME(volser) BCDS BACKUPCONTENTS(1) -  
  SELECT(MULTIPLEVOLUME)
```

To obtain information about the VSAM data sets that were on a level 0 volume at the latest incremental backup, issue the following command:


```
LIST PRIMARYVOLUME(volser) BCDS BACKUPCONTENTS(0) -  
SELECT(VSAM)
```

Obtaining tape volume information

You can obtain information about all tape volumes or specific tape volumes. In addition, you can select the volumes with specific characteristics.

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and where you *do not* want a list of data set information for the volume, issue the following command:

```
LIST TAPETABLEOFCONTENTS
```

To obtain a list of tape volume information from an OCDS, where you specify the volume serial number and you *do not* want a list of data set information for the volume, issue the following command:

```
LIST TAPETABLEOFCONTENTS(volser) NODATASETINFORMATION
```

To obtain a list of tape volume information from an OCDS, where you specify the volume serial number and you *do* want a list of data set information for the volume, issue the following command:

```
LIST TAPETABLEOFCONTENTS(volser) DATASETINFORMATION
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of volume entries for all ML1 volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(MIGRATIONLEVEL1)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of volume entries for all ML2 volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(MIGRATIONLEVEL2)
```

To obtain a list of ML2 partial tape volumes that are currently associated for output by specific migration or recycle tasks, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(ASSOCIATED)
```

To obtain a list of ML2 partial tape volumes that are available for an initial allocation, and are not currently associated for output by any migration or recycle task, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(NOTASSOCIATED)
```

To obtain a list of all ML2 volumes taken away from another function because RECALL needed a data set on the volume, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(RECALLTAKEAWAY)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of backup volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(BACKUP)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of both migration and backup volumes, issue the following command:


```
LIST TAPETABLEOFCONTENTS SELECT(BOTH)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of tape volumes marked full that do not have alternate tape volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(NOALTERNATEVOLUME)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of tape volumes that have failed recycle, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(FAILEDRECYCLE)
```

To obtain a list of tape volume information from an OCDS, where you *do not* specify a volume serial number and you want a list of tape volumes that have a mismatch between the OCDS TTOC record and the actual tape media, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(FAILEDCREATE)
```

To obtain a list of tape volume information showing volumes that are marked as disaster alternate volumes, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(DISASTERALTERNATEVOLUMES)
```

To obtain a list of tape volume information showing volumes that are capable of being resumed after Audit Mediacontrols was interrupted, issue the following command:

```
LIST TAPETABLEOFCONTENTS SELECT(RESUMEAUDIT)
```

Obtaining user authorization information

To determine the level of DFSMSHsm authorization of a particular user ID, issue the following command:

```
LIST USER(userid)
```

Determining which tape dump volumes are available for output during dump processing

To determine which tape volumes are available for volume dump processing, issue the following command:

```
LIST DUMPVOLUME SELECT(AVAILABLE)
```

This command lists each dump volume that is available for use for dump output.

To find out what tape volumes are available for a particular dump class for volume dump processing, issue the following command:

```
LIST DUMPVOLUME SELECT(AVAILABLE DUMPCCLASS(class))
```

Obtaining information about the DFSMSHsm address space

You can use the DFSMSHsm QUERY command to display the status of DFSMSHsm parameters, statistics, and pending requests. The output of the QUERY command is messages returned to your terminal and written in the DFSMSHsm log.

To obtain a list of parameters that apply to aggregate backup and recovery, issue the following command:

```
QUERY ABARS
```

To obtain a list about what functions are running in DFSMSHsm, issue the following command:

```
QUERY ACTIVE
```

This list includes the following types of information:

- The status of the data set backup function
- The status of variables that affect the placement and selection of requests on the common recall queue
- The processing status of FRBACKUP and FRRECOV COPYPOOL commands

To obtain the Task Control Block addresses of active DFSMSHsm data movement tasks, issue the following command:

```
QUERY ACTIVE(TCBADDRESS)
```

TCBADDRESS is an optional subparameter of ACTIVE requesting only messages directly related to data movement activities be listed, along with extra information allowing cancellation of the active task. For ABARS the Secondary Address Space Index is returned when QUERY ACTIVE(TCBADDRESS) is issued.

To obtain a list of volumes that are defined for aggregate recovery, issue the following command:

```
QUERY ARPPOOL
```

To display the progress of DFSMSHsm automatic functions, issue the following command:

```
QUERY AUTOPROGRESS
```

To obtain a list of backup and dump parameters, including new data set backup parameters, issue the following command:

```
QUERY BACKUP
```

To obtain a list of all backup and dump parameters and all backup volumes, issue the following command:

```
QUERY BACKUP(ALL)
```

To obtain a list of all backup parameters and all daily backup volumes, issue the following command:

```
QUERY BACKUP(DAILY)
```

To obtain a list of all backup parameters and daily backup volumes assigned to a particular day in the backup cycle, issue the following command:

```
QUERY BACKUP(DAILY(day))
```

To obtain a list of all spill parameters and all spill backup volumes, issue the following command:

```
QUERY BACKUP(SPILL)
```

To obtain a list of backup parameters and all unassigned backup volumes, issue the following command:

```
QUERY BACKUP(UNASSIGNED)
```


To obtain a list of parameters from the SETSYS command that are governing the backup of the control data sets, issue the following command:

```
QUERY CDSVERSIONBACKUP
```

To determine how full the DFSMSshm coupling facility structures are, issue the following command:

```
QUERY COMMONQUEUE
```

To determine all outstanding recall requests in the CRQ, issue the following command:

```
QUERY COMMONQUEUE(RECALL)
```

To display space-allocated-and-used statistics for the control data sets, issue the following command:

```
QUERY CONTROLDATASETS
```

To determine if any of the volumes defined to a copy pool are in an active FlashCopy relationship, issue the following command:

```
QUERY COPYPOOL(cpname)
```

For FlashCopy version 2, the source volume may be in more than one relationship. This command can also be used to determine if another backup version can be created when using FlashCopy version 1, or if a recover can be performed. This command works only on FlashCopy relationships.

To display the common service area limits that were specified with the SETSYS command, issue the following command:

```
QUERY CSALIMITS
```

To display a list of requests associated with a specific data set, issue the following command:

```
QUERY DATASETNAME(dsname)
```

Note: This command returns information only for those requests that were initiated by the host on which the QUERY DATASETNAME command is issued.

To display information for active DFSMSshm address spaces in a single z/OS image, issue the following command:

```
QUERY IMAGE
```

To display the key ranges and the associated ML2 volumes, issue the following command:

```
QUERY MIGRATIONLEVEL2
```

Note: This command also displays the volume serial numbers of the tape ML2 volumes that are currently selected as the target volumes for data set and volume migration and secondary space management.

To display a list of the recall pools, issue the following command:

```
QUERY POOL
```

To display a list of requests associated with a specific request number, issue the following command:

```
QUERY REQUEST(reqnum)
```


Note: This command returns information only for those requests that were initiated by the host on which the QUERY REQUEST command is issued.

To display a list of data set levels that are prevented from migrating by the SETMIG command, issue the following command:

```
QUERY RETAIN
```

To display the parameters that you set with the SETSYS command, issue the following command.

```
QUERY SETSYS
```

To display the space statistics on specified primary and ML1 volumes, issue the following command:

```
QUERY SPACE(volser1,volser2,...,volsern)
```

Note: The QUERY SPACE command is not valid for SMS-managed volumes.

To display the parameters you specified in the PROC statement of the DFSMSHsm startup procedure, issue the following command:

```
QUERY STARTUP
```

To display selected DFSMSHsm statistics that have accumulated during the day, issue the following command:

```
QUERY STATISTICS
```

To display the TRAP requests you have established, issue the following command:

```
QUERY TRAPS
```

To display a list of requests associated with a specific user ID, issue the following command:

```
QUERY USER(userid)
```

Note: This command returns information only for those requests that were initiated by the host on which the QUERY USER command is issued.

To display the current volume pools, issue the following command:

```
QUERY VOLUMEPOOL
```

To display requests for DFSMSHsm services that are waiting to be processed, including information on the number of requests waiting in the common recall queue, issue the following command:

```
QUERY WAITING
```

Note: This command returns information only for those requests that were initiated by the host on which the QUERY WAITING command is issued.

Determining which tape volumes are available for output

The QUERY command is also used to determine tape volumes that are to be requested for backup and migration if volumes are available for selection.

Determining which tape backup volumes are available for output

To find out which tape backup volumes are available for volume backup processing, follow this procedure:

1. Issue a QUERY BACKUP command to determine the current day in the backup cycle.
2. Issue a QUERY BACKUP(DAILY(x)) to find out what tapes are assigned to day x in the backup cycle. Replace x with the current day in the backup cycle.
3. Look at the volume serial numbers of the daily backup volumes assigned to the current day in the backup cycle. The first tape daily backup volume shown with an 'A' for available in the ARC0164I message is the first volume that DFSMSHsm considers to allocate when it selects a tape backup volume.
4. To determine which tape backup volumes are empty and available for subsequent volume selection (if you specified SELECTVOLUME(SPECIFIC) on the SETSYS command), issue the following command:

```
LIST BACKUPVOLUME SELECT(EMPTY)
```

You should be aware that additional tapes can be made available by performing the RECYCLE, EXPIREBV, or DELVOL commands, and by creating new backup tapes, which causes existing backup tapes to be invalidated.

Determining which tape migration level 2 volumes are available for output

To find out which tape ML2 volumes are available for migration processing, follow this procedure:

1. Issue a QUERY MIGRATIONLEVEL2 command.
2. Look at the volume serial numbers of the ML2 volumes. The ARC0227I message shows the volser of the tapes currently associated with the respective data set, volume, and secondary space management tasks. These are the tapes DFSMSHsm attempts to allocate when the tasks start. If you want to know what other ML2 tapes exist and might be used in place of these, issue LIST TTOC SELECT(NOTASSOCIATED).
3. Issue one of the following commands to determine which tape ML2 volumes are empty and available for subsequent volume selection (if you specified SELECTVOLUME(SPECIFIC) on the SETSYS command):

```
LIST MIGRATIONLEVEL2(TAPE) SELECT(EMPTY)
```

```
LIST TTOC SELECT(ML2 EMPTY)
```

Obtaining statistical information about DFSMSHsm operations

You can use the REPORT command to obtain statistics about DFSMSHsm's operations. You can report the statistics on either a daily basis for all volumes or a volume basis for specific volumes. You can restrict the scope and format of the statistics reported to the backup, deletion of data sets or UNIX files, migration, recall, recovery, recycle, and spill functions. You can restrict the beginning date of the statistics or the dates between which the statistics are reported. You can also restrict the report to a summary of the statistics.

To report daily statistics about data sets and UNIX files that are backed up and to send the report to a DASD data set, issue the following command:

```
REPORT DAILY FUNCTION(BACKUP) OUTDATASET(dsname)
```

To obtain daily statistics about all functions, issue the following command:

```
REPORT DAILY FUNCTION
```

The report is sent to the SYSOUT class specified with the SETSYS command.

To obtain volume statistics for all volumes and to send the report to SYSOUT class A, issue the following command. This report also includes the total number of times that a fast replication backup was

successfully created for a volume and the number of failed attempts for the volume. Recover statistics that are associated with the fast replication operation are also reported.

```
REPORT VOLUME FUNCTION SYSOUT(A)
```

To obtain volume statistics for one volume about migration between two dates and delete the statistics records after the report is printed, issue the following command:

```
REPORT VOLUME(volser FUNCTION(MIGRATION) -  
FROMDATE(date) TODATE(date) DELETE
```

To obtain statistics for UNIX files, the VOLSER, %UNIX%, can be specified on the VOLUME parameter. Because UNIX files are not allocated on a DASD volume, DFSMSHsm uses %UNIX% as the volume indicator when backing up a UNIX file. Only one %UNIX% may be specified on the command.

You can issue the following command:

```
REPORT VOLUME(%UNIX% FUNCTION(BACKUP))
```

After you have formatted a report, you can free the space allocated for the statistics that made up the report. To clean up unneeded report statistics, do one of the following:

- Issue the DELETE parameter of the REPORT command. DFSMSHsm deletes the statistics records (used as input for the report) after the REPORT command finishes its processing. For example, if you allocate one additional cylinder for the accumulation of statistics and control records for one week, issue the DELETE parameter of the REPORT command once a week.
- Specify how often DFSMSHsm deletes MCDS data set records and statistics records with the MIGRATIONCLEANUPDAYS(*recalldays statdays*) parameter of the SETSYS command. The cleanup of these records occurs as a part of secondary space management.

Related reading

- For an alternate method of gathering information about DFSMSHsm activities from SMF records, see [“Running new reports with report generator” on page 575](#) and [z/OS DFSMSrmm Reporting](#).
- For complete syntax of the report command, see Chapter 63, [“REPORT Command: Requesting reports based on DFSMSHsm statistics records,” on page 1043](#).

Running new reports with report generator

DFSMSHsm reports can be run with the DFSMSrmm Report Generator. The reports are based on FSR and WWFSR (SMF) data and DFSMSHsm control data set data that is obtained by DCOLLECT.

For HSM FSR Records

- ARCGAB01 ABARS ABACKUP Statistics
- ARCGAR01 ABARS ARECOVER Statistics
- ARCGS00A Statistics for Automatic Space Management
- ARCGS001 Statistics for DFSMSHsm
- ARCGS002 Statistics for Backup
- ARCGS003 Statistics for Migration
- ARCGS004 Statistics for Recall
- ARCGS005 Statistics for Recovery
- ARCGS006 Statistics for Volume Dump
- ARCGS007 Statistics for Restore from Dump Copy
- ARCGS008 Statistics for FRBACKUP

- ARCGS009 Statistics for FRRECOVER
- ARCGS010 DFSMSHsm Thrashing Report
- ARCGS011 Statistics for Class Transition

For DCOLLECT Records

- ARCGDB01 DCOLLECT BACKUP DATA
- ARCGDD01 DCOLLECT DASD CAPACITY PLANNING
- ARCGDM01 DCOLLECT MIGRATION DATA
- ARCGDT01 DCOLLECT TAPE CAPACITY PLANNING

The reporting records are:

ARCFSR2 (SYS1.MACLIB)

Reformatted from ARCFSR for reporting records with DFSMSrmm Report Generator. Maps FSR DS records for DFSMSHsm in SMF form. Skeleton is ARCGFSRC.

ARCWFSR2 (SYS1.MACLIB)

Reformatted from ARCWWFSR for reporting records with DFSMSrmm Report Generator. Maps WWFSR for DFSMSHsm in SMF form. Skeleton is ARCGWFSC.

ARCGFSRC/ARCGWFSC

Supplied skeletons used by the generator to convert FSR and WWFSR records to FSR2 and WWFSR2 records respectively. Shipped in SYS1.DGTSLIB.

Note: Report generator cannot process the ARCFSR and ARCWWFSR records. ARCFSR and ARCWWFSR records are converted to ARCFSR2 and ARCWWFSR2, respectively, using ARCGFSRC and ARCGWFSC while running DFSMSrmm report generator.

ARCUTILP (SYS1.MACLIB)

It is much easier to create and manage reports, within the DFSMSrmm Report Generator, when the TYPE keyword is used to specify a single record type such as TYPE=M. Otherwise, all records are mapped. ARCUTILP provides keywords to map each structure. TYPE=ALL|M|B|C|T is as follows:

- ALL = For all structures
- M = Migrate
- B = Backup
- C = DASD Capacity Planning
- T = Tape Capacity Planning

No conversion is needed for ARCUTILP records as each type is uniquely written.

Prior to creating DFSMSHsm reports by using the DFSMSrmm Report Generator, DFSMSHsm FSR and WWFSR records need to be collected. They are written to SMF data sets by DFSMSHsm. Following is a JCL sample that collects this data from SMF. In this case SYS1.MANX and SYS1.MANY are the output ds from SMF, but this can vary depending upon setup.

The DFSMSrmm Report Generator includes sample report types for DFSMSHsm FSR and DCOLLECT reporting. When DFSMSrmm Report Generator searches for sample report definitions in the installation (SAMPLIB) library it looks for DFSMSHsm sample members with the prefixes ARCG and EDGG.

This sample JCL describes collecting SMF records containing FSR and WWFSR Records for use in running DFSMSrmm Report Generator.

For more information about EFASMFDP and IFASMFDL see [*z/OS MVS System Management Facilities \(SMF\)*](#).

SMF dump sample JCL:


```

//SMFDUMP JOB ,
//* USE D SMF TO SEE DS STATUS
//* USE D SMF,0 TO SEE SMF SETTINGS
//* USE SWITCH SMF TO SWAP THE ACTIVE AND INACTIVE DS
//* NOTE OPTIONS = DUMP | CLEAR | ALL
//* ONLY INACTIVE DS CAN BE CLEARED
//* DEFAULT ALL = DUMP AND CLEAR
//DUMPSMF EXEC PGM=IFASMFDP,REGION=512K
//INDD0 DD DSN=SYS1.MANX,DISP=SHR
//INDD1 DD DSN=SYS1.MANY,DISP=SHR
//DUMPOUT DD DSN=DFHSM.EXTRACT,UNIT=SYSALLDA,
// DISP=(NEW,CATLG),SPACE=(CYL,(50,5),RLSE)
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
INDD(INDD0,OPTIONS(DUMP))
INDD(INDD1,OPTIONS(DUMP))
OUTDD(DUMPOUT,TYPE(xxx))
/*

```

Where xxx is the DFSMSHsm SMF type you used. Refer to the startup procedure for DFSMSHsm or issue QUERY SETSYS.

For more information about IFASMF DL and IFASMF DP see Dumping SMF log streams - IFASMF DL and Dumping SMF log streams - IFASMF DP in Using the SMF Dump Programs in [z/OS MVS System Management Facilities \(SMF\)](#).

This is sample JCL for collecting DCOLLECT records for possible use when running DFSMSrmm Report Generator.

The DCOLLECT job step can be generated within the DFSMSHsm Report Generator by selecting **Create report data...Y (Y/N)** in the JCL generation panel (EDGPG022).

DCOLLECT sample JCL:

```

//DCOLLECT JOB ,
//STEP1 EXEC PGM=IDCAMS
//OUTDS DD DSN=DFHSM.DCOLLECT.DATA,DISP=(NEW,CATLG),
// UNIT=SYSALLDA,
// DCB=(LRECL=644,BLKSIZE=0,RECFM=VB),SPACE=(CYL,(50,2)),
// AVGREC=K
//SYSPRINT DD SYSOUT=A
//MCDS DD DSN=DFHSM.MCDS,DISP=SHR
//BCDS DD DSN=DFHSM.BCDS,DISP=SHR
//SYSIN DD *
DCOLLECT -
OUTFILE(OUTDS) -
MIGRATEDATA -
BACKUPDATA -
CAPD
/*

```

To start the DFSMSrmm report generator use option, G - Report Generation, on the ISMF Storage Administrator Primary Panel.

Set up need only be done once for a given user.

Select 0-Options. This only needs to be done once.

Panel Help

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
X  EXIT             - Exit DFSMSrmm dialog
Enter selected option or END command. For more info., enter HELP or PF1.

```

The options menu has three selections:

- USER - Sets up basics and must be done by all users at least one time.

- SORT - Specifies sort information for DFSMSrmm report users.
- REPORT - Specifies Report options.

See Figure 8. Selecting the Options Option on the DFSMSrmm Dialog Options Menu Panel in Chapter 2, Using the DFSMSrmm Report Generator in [z/OS DFSMSrmm Reporting](#) for specifying libraries for the report generator.

At a minimum the JCL job card should be added. This will be used by all jobs submitted by Report Generator. If the job card is not supplied, the JCL generation uses the ISPF job card if it exists. If not, it generates a default job card.

```

Panel  Help

                                DFSMSrmm Dialog User Options
Command ==>

Date format . . . . . JULIAN      ( American, European, Iso or Julian )
Time zone . . . . . LOCAL        ( zone offsetHH:MM:SS )

Confirm deletes . . . YES        ( Yes or No )

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

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

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

Job statement information:-

==> //HSMREPT JOB ,CLASS=A,MSGCLASS=H.MSGLEVEL(1,1)
==> //*
==> //*
==> //*

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

Add the name of the target data sets for this user.

```

Panel  Help

                                DFSMSrmm Report Options
Command ==>

Report definition libraries:
  User . . . . . 'USER001.REPORT.LIB'
  Installation . . . . . 'TEAM001.REPORT.LIB'
  Product . . . . . 'SYS1.SAMPLIB'

User report JCL library . . 'USER001.REPORT.JCL'
```

Before running reports, select Reporting Tool using Option 3.

```

Panel  Help

                                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
X  EXIT            - Exit DFSMSrmm dialog
Enter selected option or END command. For more info., enter HELP or PF1.
```

Notice the number of columns between each heading is set on the Reporting Tool Panel.

Report Types are provided by DFSMSShsm as the outline for creating reports.

Table 52. Report Types Supplied by DFSMSHsm

Name	Description
ARCGWFSR	DFSMSHsm ABARS Report
ARCGDBCK	DFSMSHsm DCOLLECT BACKUP
ARCGDDSD	DFSMSHsm DCOLLECT DASD CAP
ARCGDMIG	DFSMSHsm DCOLLECT MIGRATION
ARCGDTAP	DFSMSHsm DCOLLECT TAPE CAP
ARCGFSR2	DFSMSHsm FSR-SMF Records

Report Type has predefined types but the user may modify and add their own, however, it is not necessary since sample reports are also provided.

Panel Help

DFSMSImm 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
X  EXIT             - Exit DFSMSImm dialog
Enter selected option or END command. For more info., enter HELP or PF1.
```

Report Types is sufficient for all reports but users may modify or add their own. DFSMSHsm has already supplied types for all currently provided records.

Panel Help

DFSMSImm Report Types

Row 2 to 7 of 32
Scroll ==> CSR

Command ==>

The following line commands are valid: A,C,D,L,M,R, and S

```

S Name      Description
-----
ARCGWFSR DFSMSHsm ABARS Report
  Macro library . . : 'SYS1.MACLIB'
  Applicable macros : ARCGWFSR2
  Input data set. . : 'DFHSM.EXTRACT.ABARS.REFORMAT'
ARCGDBCK DFSMSHsm DCOLLECT BACKUP
  Macro library . . : 'SYS1.MACLIB'
  Applicable macros : ARCUTILP
  Input data set. . : 'DFHSM.DCOLLECT.DATA'
ARCGDDSD DFSMSHsm DCOLLECT DASD CAP
  Macro library . . : 'SYS1.MACLIB'
  Applicable macros : ARCUTILP
  Input data set. . : 'DFHSM.DCOLLECT.DATA'
ARCGDMIG DFSMSHsm DCOLLECT MIGRATION
  Macro library . . : 'SYS1.MACLIB'
  Applicable macros : ARCUTILP
  Input data set. . : 'DFHSM.DCOLLECT.DATA'
ARCGDTAP DFSMSHsm DCOLLECT TAPE CAP
  Macro library . . : 'SYS1.MACLIB'
  Applicable macros : ARCUTILP
  Input data set. . : 'DFHSM.DCOLLECT.DATA'
ARCGFSR2 DFSMSHsm FSR-SMF Records
  Macro library . . : 'SYS1.MACLIB'
  Applicable macros : ARCGFSR2
  Input data set. . : 'DFHSM.EXTRACT.REFORMAT'
```

Line commands M and L allow viewing the source macros and the results of the compiles. There are other commands that may be seen in the Help screen. These allow the user to modify and create new report types.

Panel Help

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
X EXIT - Exit DFSMSrmm dialog
Enter selected option or END command. For more info., enter HELP or PF1.

Sample reports definitions supplied by DFSMSShsm include two WWFSR reports, four DCOLLECT reports, and nine FSR reports. These reports are provided to be used as is or can be customized to create new reports.

Panel Help

DFSMSrmm Report Definitions

Row 1 to 31 of 40

Command ==>

Scroll ==> CSR

The following line commands are valid: A,D,G,J,L,M,N,S, and T

S Name	Report title	Report type	User id
ARCGAB01	ABARS ABACKUP Statistics	DFSMSHsm ABARS Report	HSM
ARCGAR01	ABARS ARECOVER Statistics	DFSMSHsm ABARS Report	HSM
ARCGDB01	DCOLLECT BACKUP DATA	DFSMSHsm DCOLLECT BACKUP	HSM
ARCGDD01	DCOLLECT DASD CAPACITY PLANNIN	DFSMSHsm DCOLLECT DASD CAP	HSM
ARCGDM01	DCOLLECT MIGRATION DATA	DFSMSHsm DCOLLECT MIGRATION	HSM
ARCGDT01	DCOLLECT TAPE CAPACITY PLANNIN	DFSMSHsm DCOLLECT TAPE CAP	HSM
ARCGS001	Statistics for DFSMSHsm	DFSMSHsm FSR-SMF Records	HSM
ARCGS002	Statistics for Backup	DFSMSHsm FSR-SMF Records	HSM
ARCGS003	Statistics for Migration	DFSMSHsm FSR-SMF Records	HSM
ARCGS004	Statistics for Recall	DFSMSHsm FSR-SMF Records	HSM
ARCGS005	Statistics for Recovery	DFSMSHsm FSR-SMF Records	HSM
ARCGS006	Statistics for Volume Dump	DFSMSHsm FSR-SMF Records	HSM
ARCGS007	Statistics for Restore from Du	DFSMSHsm FSR-SMF Records	HSM
ARCGS008	Statistics for FRBACKUP	DFSMSHsm FSR-SMF Records	HSM
ARCGS009	Statistics for FRRecover	DFSMSHsm FSR-SMF Records	HSM
ARCGS010	DFSMSHsm Thrashing Report	DFSMSHsm FSR-SMF Records	HSM
ARCGS011	Statistics for Class Transiti	DFSMSHsm FSR-SMF Records	HSM

The DFSMSrmm report generator selection option is where sample reports may be viewed, modified, and created.

The DFSMSrmm Report Definition Panel shows the select columns for general DFSMSHsm reports. It shows the selected fields (CO) having a grouping (G), search criteria (S), and the sorted columns (SO). These fields may be changed to create custom reports.

Panel Help

```
DFSMSrmm Report Definition - ARCGS001      Row 1 to 26 of 80
Command ===>                               Scroll ===> CSR

Report title . . . Statistics for DFSMSHsm
Report footer . .
Reporting tool . : ICETOOL                  Report width: 209

Use END to save changes or CANCEL
Select a field name with S to specify a field selection criterion
```

S	CO	SO	Field name	Column header text	CW	Len	Typ
	G	1A	FSR2HOST	HOSTID	6	2	C
	1	2A	FSR2TYPE	FUNC (hex)	10	1	B
*	2	3A	FSR2DATR	DATE	8	8	C
	3	4A	FSR2DSN	DSN OR CP NAME	44	44	C
	4		FSR2FVOL	SOURCE	6	6	C
	5		FSR2TVOL	TARGET	6	6	C
	6		FSR2NENT1	# TAPE	6	2	N
	7		FSR2NENT2	# RCY OUT	9	2	N
	8		FSR2BYTR	KB RD	5	4	N
	9		FSR2BYTW	KB WR	5	4	N
	10		FSR2JBN	JOB NAME	8	8	C
	11		FSR2UID	USERID	8	8	C
	12		FSR2MCNAM	MC NAME	8	8	C
	13		FSR2SCNAM	SC NAME	8	8	C
	14		FSR2RC	RC (hex)	8	4	B
*			FSR2RTY	SMF RECORD TYPE	15	1	B

The search criteria must be met in each record selected for inclusion in the report.

Panel Help

```
DFSMSrmm Report Criteria - ARCGS001      Row 1 to 2 of 2
Command ===>                               Scroll ===> CSR

Report title : Statistics for DFSMSHsm

Use END to save changes or CANCEL
The following line commands are valid: B,D,N,P,R,T, and I (for details)
Comparison operators: EQ =, NE <>, GT >, GE >=, LT <, LE <=, IN, BW, SE, SN
Conjunction: AND, OR, AND(, )AND
```

S	Field name	Op	Compare value(s)	Conj	Len	Typ
	FSR2DATR	BW	&TODAY; - 7 Days,&TODAY;		8	C
	FSR2RTY	EQ	X'FF'		1	B

This panel defines the search criteria which must be met in each record selected for inclusion in the report.

Panel Help

```
DFSMSrmm Report Definitions      Row 1 to 3
Command ===>                               Scroll ==

The following line commands are valid: A,D,G,J,L,M,N,S, and T
```

S	Name	Report title	Report type
	ARCGAB01	ABARS ABACKUP Statistics	DFSMSHsm ABARS Report
	ARCGAR01	ABARS ARECOVER Statistics	DFSMSHsm ABARS Report
	ARCGDB01	DCOLLECT BACKUP DATA	DFSMSHsm DCOLLECT BACKUP
	ARCGDD01	DCOLLECT DASD CAPACITY PLANNIN	DFSMSHsm DCOLLECT DASD CAP
	ARCGDM01	DCOLLECT MIGRATION DATA	DFSMSHsm DCOLLECT MIGRATION
	ARCGDT01	DCOLLECT TAPE CAPACITY PLANNIN	DFSMSHsm DCOLLECT TAPE CAP
G	ARCGS001	Statistics for DFSMSHsm	DFSMSHsm FSR-SMF Records
	ARCGS002	Statistics for Backup	DFSMSHsm FSR-SMF Records
	ARCGS003	Statistics for Migration	DFSMSHsm FSR-SMF Records
	ARCGS004	Statistics for Recall	DFSMSHsm FSR-SMF Records

After selecting generate report the following panel variables should be entered:

input data set

The set of records for the report that must be made up of ARCF SR2, ARCWF SR2, or ARCUTILP (DCOLLECT). These records are previously converted except ARCUTILP records which do not need to be converted.

date format

YYYYDDD is specific to DFSMSHsm FSR records.

create report data

When Y tells Report Generator that Skeleton_Variable_2 has source data ARCF SR or ARCWWFSR that must be converted to ARCF SR2 or ARCWF SR2. If this is the case, the Skeleton_Variable_1 must provide the FSR record type as defined in DFSMSHsm's command SETSYS SMF(xx). This step will also create the temporary data set named in **input data set**. This may be changed to a non-temporary data set by modifying the generated JCL. Using a non-temporary data set allows preservation of the input data set for other reports.

Panel Help

DFSMSrmm Report Generation - ARCGS001

Command ==>

Enter or change the skeleton variables for the generated JCL:

Input data set 'DFHSM.EXTRACT.REFORMAT'

Date format YYYYDDD

(American, European, Iso, Julian, or free form)

Required if you use variable dates (&TODAY;) in your selection criteria.

Create report data . . N (Y/N)

Choose Y if you want an extract step included into your generated JCL.

Additional skeleton variables, if an extract step is included:

Skeleton Variable_1 . . X'FF'

Skeleton Variable_2 . . DFHSM.EXTRACT

Skeleton Variable_3 . .

The skeleton selection depends on the reporting macro . . . : **ARCF SR2**
and macro keyword . . :

Enter END command to start the report generation or CANCEL

The J line command tell Report Generator to display the JCL for editing and submission.

Panel Help

DFSMSrmm Report Definitions

Row 1 to
Scroll =

Command ==>

The following line commands are valid: A,D,G,J,L,M,N,S, and T

S	Name	Report title	Report type
-	ARCGAB01	ABARS ABACKUP Statistics	DFSMSHsm ABARS Report
	ARCGAR01	ABARS ARECOVER Statistics	DFSMSHsm ABARS Report
	ARCGDB01	DCOLLECT BACKUP DATA	DFSMSHsm DCOLLECT BACKUP
	ARCGDD01	DCOLLECT DASD CAPACITY PLANNIN	DFSMSHsm DCOLLECT DASD CAP
	ARCGDM01	DCOLLECT MIGRATION DATA	DFSMSHsm DCOLLECT MIGRATION
	ARCGDT01	DCOLLECT TAPE CAPACITY PLANNIN	DFSMSHsm DCOLLECT TAPE CAP
J	ARCGS001	Statistics for DFSMSHsm	DFSMSHsm FSR-SMF Records
	ARCGS002	Statistics for Backup	DFSMSHsm FSR-SMF Records
	ARCGS003	Statistics for Migration	DFSMSHsm FSR-SMF Records
	ARCGS004	Statistics for Recall	DFSMSHsm FSR-SMF Records

Report JCL has the job card defined previously. The conversion JCL will also be visible if selected.


```

File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          USER001.REPORT.JCL(ARCGS001) - 01.01          Columns 00001 00080
Command ==>                                         Scroll ==> CSR
***** Top of Data *****
000001 //HSMREPT JOB ,CLASS=A,MSGCLASS=H,MSGLEVEL(1,1)
000002 /**
000003 //*****
000004 /*** SKELETON MEMBER EDGSGICE
000005 /*** TAILORED BY THE RMM REPORT GENERATOR
000006 //*****
000007 //ARCGFSRC EXEC PGM=SORT
000008 //*****
000009 /** z/OS DFSMSHsm V1R10
000010 /**
000011 /** PROPRIETARY V3 STATEMENT
000012 /** LICENSED MATERIALS - PROPERTY OF IBM
000013 /** "RESTRICTED MATERIALS OF IBM"
000014 /** 5694-A01
000015 /** COPYRIGHT 2008 IBM CORP.
000016 /** END PROPRIETARY V3 STATEMENT
000017 //*****
000018 /**
000019 /** FUNCTION - THIS JCL CONVERTS ARCF SR SMF RECORDS TO ARCF SR2
000020 /** RECORDS TO BE USED AS INPUT FOR REPORT GENERATOR
000021 /**
000022 /** METHOD OF ACCESS
000023 /** - INCLUDED AS AN EXTRACT STEP WHEN GENERATING REPORTS
000024 /**
000025 /** Change History:
000026 /**
000027 /** $P0=HSMRP,Z1A,072307,THMB: NEW CONVERSION SKELETON @PRA*

```

Data sets previously named **input data set** and **Skeleton_Variable_2** are source and target data sets of the conversion part of the JCL.

Note: DISP=(NEW,PASS) may be changed to create a non-temporary data set if desired.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help

EDIT          USER001.REPORT.JCL(ARCGS001) - 01.01          Columns 00001 00080
Command ==>                                         Scroll ==> CSR
000032 //SYSPRINT DD SYSOUT=*
000033 //SORTIN DD DISP=SHR,
000034 // DSN=DFHSM.EXTRACT
000035 //SORTOUT DD DISP=(NEW,PASS),
000036 // UNIT=SYSALLDA,
000037 // LIKE=DFHSM.EXTRACT,
000038 // DSN=DFHSM.EXTRACT.REFORMAT
000039 //MSGOUT1 DD SYSOUT=*
000040 //SYSOUT DD SYSOUT=*
000041 //SYSIN DD *
000042 OPTION VLSHRT,VLSCMP
000043 INCLUDE COND=((6,1,BI,EQ,X'FF'), DFSMSHSM FSR
000044 AND,(43,1,CH,NE,X'0F'),
000045 AND,(43,1,CH,NE,X'10'))
000046 RECORD TYPE=V
000047 INREC IFTHEN=(WHEN=INIT,
000048 BUILD=(1,18, 1 FM 1 x 18 - CH FSRHDR
000049 19,8, 19 FM 19 x 8 - CH FSRJBN
000050 27,4, 27 FM 27 x 4 - BI FSRRST
000051 31,4,4X'00', 31x4+4 FSRRSR
000052 35,8, 39 FM 35 x 8 - CH FSRUID
000053 43,1, 47 FM 43 x 1 - BI FSRTYPE
000054 44,1, 48 FM 44 x 1 - BI FSRFLAGS
000055 45,44, 49 FM 45 x 44 - CH FSRDSN
000056 89,6, 93 FM 89 x 6 - CH FSRTVOL
000057 95,4, 99 FM 95 x 4 - CH FSRDEVT
000058 99,6, 103 FM 99 x 6 - CH FSRFVOL

```

The date form in the sample report and the selection criteria are part of the generated JCL.


```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      USER001.REPORT.JCL(ARCGS001) - 01.01      Columns 00001 00080
Command ==>      Scroll ==> CSR
000179 //*****
000180 /** DATE CALCULATION STEP
000181 /** COMPARE VALUES CONTAINING ARE CALCULATED BASED ON RUN DATE
000182 //*****
000183 //DATECONV EXEC PGM=IKJEFT01,PARM='%EDGRGDAT'
000184 //SYSPROC DD DISP=SHR,DSN=SYS1.SEDGEXE1
000185 //SYSTSPRT DD SYSOUT=*
000186 //DATEFMT DD *
000187 DATE PATTERN:YYYYDDD
000188 //INCLIN DD *
000189 OPTION VLSHRT,VLSCMP
000190 INCLUDE COND=((161,8,CH,GE,C'&TODAY-007D;'),
000191             AND,
000192             (161,8,CH,LE,C'&TODAY-000D;')),
000193             AND,
000194             (6,1,CH,EQ,X'FF'))
000195 INREC FIELDS=(1,4,
000196             369,2,C' ',

```

The input data set, whether created as a temporary data set by the conversion JCL, or pre-existing, is used by the DFSMSrmm Report Generator when extracting data for the report.

The OUTDD card may be changed to a DSN.

```

File Edit Edit_Settings Menu Utilities Compilers Test Help
EDIT      USER001.REPORT.JCL(ARCGS001) - 01.01      Columns 00001 00080
Command ==>      Scroll ==> CSR
000240 //WRITE1 EXEC PGM=ICETOOL,REGION=0M
000241 //SYSPRINT DD SYSOUT=*
000242 //TOOLMSG DD SYSOUT=*
000243 //DFSMSG DD SYSOUT=*
000244 //INDD DD DSN=DFHSM.EXTRACT.REFORMAT,
000245 // DISP=SHR
000246 //OUTDD DD SYSOUT=*
000247 //TEMP DD UNIT=SYSALLDA,SPACE=(TRK,(5,25))
000248 //TOOLIN DD *
000249 SORT FROM(INDD) TO(TEMP) USING(INCL)
000250 DISPLAY FROM(TEMP) LIST(OUTDD) -
000251 TITLE('Statistics for DFSMSHsm') -
000252 PAGE DATE(4MD/) TIME -
000253 HEADER('FUNC (hex)') ON(8,1,HEX) -
000254 HEADER('DATE') ON(10,8,CH) -
000255 HEADER('DSN OR CP NAME') ON(19,44,CH) -
000256 HEADER('SOURCE') ON(64,6,CH) -
000257 HEADER('TARGET') ON(71,6,CH) -

```

The report may be run by issuing SUB after the line command J displays the JCL in edit or from the Report Generator JCL library as defined in the set up panels. The report may be viewed in the output of the JCL or in an output data set if one was defined.

A thrashing report (ARCGS010) is created from the DFSMSrmm Report Definition (ARCGS004) panel to demonstrate the simplicity of creating new reports from the supplied reports.

Add new selection criteria by entering the S line command by FSR2AGE.

Panel Help

DFSMSrmm Report Definition - ARCGS004 Row 1 to 26 of 80
Command ===> Scroll ===> CSR

Report title . . . Statistics for Recall

Report footer . .

Reporting tool . : ICET00L

Report width: 200

Use END to save changes or CANCEL

Select a field name with S to specify a field selection criterion

S	C0	S0	Field name	Column header text	CW	Len	Typ
*	1	1A	FSR2DATR	DATE	8	8	C
	2		FSR2TIMR	TIME REQ	8	4	B
	3	3A	FSR2TIME	TIME COMP	9	4	B
	4	2A	FSR2DSN	DSN	44	44	C
s	5		FSR2AGE	AGE	3	2	B
	6		FSR2TVOL	TARGET	6	6	C
	7		FSR2BYTR	KB READ	7	4	N
	8		FSR2BYTW	KB WRITE	8	4	N
	9		FSR2SCNAM	SC NAME	8	8	C
	10		FSR2MCNAM	MC NAME	8	8	C
	11		FSR2RECALL_RETRIES	RETRIES	7	1	B
	12		FSR2HOST	HOST	4	2	C
	13		FSR2ORGNL_HID	REQHOST	7	2	C
	14		FSR2RC	RC (hex)	8	4	B
*			FSR2TYPE	FUNCTION TYPE - Use browse (M) macro	36	1	B

In this example we are adding thrashing criteria, which in this case is two days between migrate and recall. The report job will then include any records indicating that recall occurred within two days of migration. Each installation should define the number of days between migration and recall which constitute thrashing.

Panel Help

DFSMSrmm Report Criteria - ARCGS004 Row 1 to 4 of 4
Command ===> Scroll ===> CSR

Report title : Statistics for Recall

Use END to save changes or CANCEL

The following line commands are valid: B,D,N,P,R,T, and I (for details)

Comparison operators: EQ =, NE <>, GT >, GE >=, LT <, LE <=, IN, BW, SE, SN

Conjunction: AND, OR, AND(,)AND

S	Field name	Op	Compare value(s)	Conj	Len	Typ
	FSR2DATR	BW	&TODAY; - 7 Days,&TODAY;	AND(8	C
	FSR2TYPE	EQ	X'04'	OR	1	B
	FSR2TYPE	EQ	X'05')AND	1	B
	FSR2AGE	LE	X'0002'		2	B

Note:

1. Date criteria needs adjustment

Sample reports all have a relative date specified for reporting. This date needs to be changed prior to generating reports.

2. Reports are provided as the foundation for creating user defined reports.

Sample report definitions and report types are provided as the foundation for creating your own installation and user-defined reports.

3. DCOLLECT reports require IDCAMS DCOLLECT be run first to generate source data

The DCOLLECT job step is generated as part of the J option to generate JCL - if the user specifies

Create report data . . Y.

Reporting on DFSMSHsm FSR and WWFSR records requires that you use the ARCFSR2 and ARCWFSR2 macros in the report generator with converted SMF data before use or select **Create report data . . Y** and use the generator to create the JCL to convert the records inline.

For more details about the DFSMSrmm Report Generator see Chapter 2, Using the DFSMSrmm Report Generator in *z/OS DFSMSrmm Reporting*.

Using the problem determination aid to trace DFSMSHsm processing

To help you gather information about DFSMSHsm processing, DFSMSHsm provides the problem determination aid (PDA). This aid can be used for both primary and secondary address spaces. It gathers diagnostic information at specific internal module points, records the data in a wraparound table in storage, and writes the data to data sets on DASD. You use the diagnostic information to provide data to your service representative should a problem develop with DFSMSHsm. The purpose of the PDA is to gather sufficient information during DFSMSHsm execution to enable subsequent analysis to pinpoint module flow and resource usage related to any given problem.

Note: If the DFSMSHsm PDOX data set is on a non-SMS volume that will be backed up, use the `ALTERDS datasetname VERSIONS(0)` command to prevent DFSMSHsm from backing it up.

If the PDOX data set is on an SMS volume that will be backed up, assign it to an SMS management class that does not allow backup copies.

Specifying to DFSMSHsm to use the PDA

In order for the PDA to trace DFSMSHsm processing, it must not be disabled in the ARCCMDxx member referred to in the startup member. If you disable it at that time, no space is set aside for the circular file in storage, and the DASD data sets are not opened. The PDA is controlled by the following command:

```
SETSYS PDA
```

The default (if you do not specify the SETSYS command with the PDA parameter in the ARCCMDxx member) is PDA(ON).

If you do not want to use the PDA, add the following command to the ARCCMDxx member:

```
SETSYS PDA(NONE)
```

If you specify PDA(NONE), no space is set aside for the circular file in storage, and the DASD data sets are not opened.

After the PDA is started, you can turn it off and on during DFSMSHsm operation. To turn the PDA off, issue the following command:

```
SETSYS PDA(OFF)
```

The PDA stops recording the diagnostic data, but the circular file space remains in storage, and the DASD data sets remain open.

To resume use of the PDA, issue the following command:

```
SETSYS PDA(ON)
```

Using the TRAP command to obtain information about conditions in the system when an error occurs

The TRAP command allows you to determine conditions in the system if an error occurs. You can test for all error codes in all modules, for a specific error code in all modules, for all error codes in a specific module, or for a specific error code in a specific module. When an error occurs, you can have DFSMSHsm abnormally end, log the error, or take a SNAP dump. You can control whether the abnormal end or SNAP dump occurs always, never, or once.

Note: TRAP is not supported in the secondary address space.

To trap all error codes in all modules and cause an abnormal end for every error, issue the following command:

```
TRAP ALL ABEND(ALWAYS)
```

To trap all error codes in all modules and cause an abnormal end once, issue the following command:

```
TRAP ALL ABEND(ONCE)
```

To trap a specific error code in a specific module and log the error, issue the following command:

```
TRAP module errorcode LOG
```

When the error indicated by *errorcode* occurs in the module indicated by *module*, an entry is written in the DFSMSHsm log.

To cause a SNAP dump every time a specific error code occurs in a specific module, issue the following command:

```
TRAP module errorcode SNAP(ALWAYS)
```

Related reading

For more information about the TRAP command, see [z/OS DFSMSHsm Diagnosis](#).

Using DEBUG to determine the results of DFSMSHsm processing without moving data sets

When you first set up DFSMSHsm or when you make major changes to the data set selection parameters you have established for it, you may want to know which data sets are selected during volume space management, backup and backup expiration before you allow it to move any data sets. To allow this freedom to test, DFSMSHsm provides a DEBUG mode of operation. To run in DEBUG mode, put the following command in the ARCCMDxx member for the DFSMSHsm host on which you want to perform the testing:

```
SETSYS DEBUG
```

The DEBUG parameter of the SETSYS command permits DFSMSHsm to exercise all of its volume backup, dump, and migration functions; but instead of moving or deleting data sets, it merely writes messages to the activity logs to indicate what data set movement or deletion would have occurred. For example, if you are already using DFSMSHsm and want to test the effects of the EXPIREBV command, you could issue the SETSYS DEBUG command followed by the EXPIREBV command. DFSMSHsm would then write messages to the activity logs, indicating which of your backup versions would have been deleted by the EXPIREBV command.

Issuing the ABACKUP and ARECOVER commands while SETSYS DEBUG is in effect causes the commands to be processed as if the VERIFY parameter were in effect.

If DFSMSHsm is stopped when it is running in DEBUG mode, any command requests that have not been processed are lost and are not processed when DFSMSHsm is restarted. See [“Stopping DFSMSHsm” on page 595](#) for a discussion of restarting processing when DFSMSHsm is restarted.

The FREQUENCY parameter of the DEFINE DUMPCCLASS command is calculated from the last date that the level 0 volume was dumped while DFSMSHsm was not in DEBUG mode. This date is not updated when DFSMSHsm is running in DEBUG mode because it would prevent volume dumps from processing on schedule when NODEBUG is specified. In DEBUG mode, if the date last dumped is zeros, the FREQUENCY value is not calculated, the level 0 volume is eligible to be dumped, and the date last dumped is not

updated. To cancel the effect of the DEBUG parameter of the SETSYS command, you can issue the following command:

```
SETSYS NODEBUG
```

However, if the DEBUG parameter is specified in the ARCCMDxx member, you may find it more convenient just to remove the SETSYS DEBUG command from the ARCCMDxx member.

Chapter 15. DFSMSHsm tape libraries

A system-managed tape library is a named collection of storage groups, and tape subsystems and their associated tape cartridges.

Automated tape library

An automated tape library (ATL) allows you to use system-managed storage for mountable tape cartridges. Mount requests for these cartridges are handled by a robotic tape cartridge handler. An ATL provides management facilities for tape volumes, but does not manage the data sets contained on those tape volumes.

Related reading

For more information about the automated tape library, see [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Manual tape library

A manual tape library (MTL) also allows you to use system-managed storage for mountable tape cartridges. Like an ATL, it provides management facilities for tape volumes, but does not manage the data sets contained on those tape volumes. Unlike an ATL, mount requests are directed to a human operator.

Related reading

For more information about a manual tape library, see [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Chapter 16. Using DFSMSHsm with DFSMSrmm

The z/OS component, DFSMSrmm provides enhanced management for tape volumes that DFSMSHsm uses for each of its tape functions. Precisely how the two products work together depends on how you use each of them in your installation.

DFSMSHsm can tell DFSMSrmm when it no longer requires a tape volume or when a tape volume changes status. When DFSMSHsm is using DFSMSrmm, it cannot mistakenly overwrite one of its own tape volumes if an operator mounts a tape volume in response to a request for a non-specific tape volume.

DFSMSrmm supports any of the naming conventions DFSMSHsm uses for its tape data sets, and all DFSMSHsm options—with the exception of password security on tape volumes—are supported.

DFSMSrmm lets you use a system scratch tape pool or a DFSMSHsm-managed tape pool. DFSMSrmm manages the movement of disaster recovery and vital record data for both pool types. It also provides a way to retain DFSMSHsm tapes until the contents are invalid.

Recommendation: Use the system scratch tape pool.

You must let DFSMSHsm decide whether a tape volume contains valid data and whether it should return to the scratch pool or DFSMSHsm-managed tape pool.

You can manage the retention of valid tape volumes in two ways:

- Use DFSMSrmm vital record specifications to retain tape volumes until DFSMSHsm finishes with them.
- Set up DFSMSHsm to use expiration date protected tape volumes. Use the date 99365 to prevent DFSMSrmm from considering the volumes for release at any time.

You should use vital record specifications because you can avoid expiration dates, operator message handling, and the need to reinitialize tape volumes before you can reuse them. However, if you are using a DFSMSHsm-managed tape pool, you can use expiration date protection without this overhead; DFSMSHsm can override expiration protection for its own tape volumes.

DFSMSHsm uses the DFSMSrmm EDGTVEXT interface to maintain the correct volume status. DFSMSrmm provides a programming interface so you can use the DFSMSHsm exit.

Related reading

For more information about the DFSMSrmm EDGTVEXT interface, see [*z/OS DFSMSrmm Implementation and Customization Guide*](#).

Retaining DFSMSHsm tape volumes by expiration date

If you use the system scratch tape pool for DFSMSHsm tape volumes, you need a way to manage tape volumes protected with expiration dates. In this situation, DFSMSrmm lets you automate the responses to expiration date protection messages for system scratch pool tape volumes.

Related reading

For more information about using the system scratch tape pool for DFSMSHsm tape volumes, see [*z/OS DFSMSrmm Implementation and Customization Guide*](#).

Retaining DFSMSHsm tape volumes by DFSMSrmm vital record processing

DFSMSrmm vital record processing uses either data set names or volume serial numbers to determine how long to keep tapes. This retention period overrides any expiration date or retention period previously

specified for the volume. In addition, you can use vital record specifications to identify volumes that should be moved out of the library for safe keeping.

Related reading

For more information about DFSMSrmm vital record processing, see [z/OS DFSMSrmm Implementation and Customization Guide](#).

Tape security when using DFSMSHsm and DFSMSrmm

The way in which you use the DFSMSHsm option DFSMSHsm TAPESECURITY may conflict with the way DFSMSrmm protects volumes.

Related reading

For more information about protecting volumes, see [z/OS DFSMSrmm Implementation and Customization Guide](#).

Recommendations for using DFSMSHsm with DFSMSrmm

You should perform the following steps when running DFSMSHsm with DFSMSrmm:

- Run DFSMSHsm with a system scratch pool that DFSMSrmm manages.
- Protect HSM tape data either using RACF TAPEVOL profiles and SETSYS TAPESECURITY(RACF), or using RACF DATASET profiles by using the RACF TAPEDSN option or the DEVSUPxx keyword TAPEAUTHDSN=YES.
- Use retention period and AUTOREUSE for automatic dumps.
- Set the following DFSMSHsm parameters. For the system options, specify:

```
SETSYS SELECTVOLUME(SCRATCH) -  
TAPERELETION(SCRATCH) -  
TAPESECURITY(RACF) -  
PARTIALTAPE(REUSE)
```

For automatic dump definitions, specify:

```
DEFINE DUMPCLASS(class -  
AUTOREUSE -  
RETENTIONPERIOD(days))
```


Chapter 17. Operating procedures for DFSMSHsm

The computing system operator is expected to perform certain operations for DFSMSHsm. This topic describes those operations. You might want to copy this topic to provide your system operators with directions for performing the operations.

The following operations are described here:

- Partitioning DFSMSHsm functions
- Starting DFSMSHsm
- Stopping DFSMSHsm
- Preventing DFSMSHsm functions from running
- Allowing previously prevented DFSMSHsm functions to run
- Canceling DFSMSHsm queued requests
- Altering the Priority of DFSMSHsm Queued Requests
- Printing DFSMSHsm log information
- Restarting DFSMSHsm after an abnormal end
- Running automatic functions concurrently
- Updating the DFSMSHsm control data sets

Partitioning DFSMSHsm functions

This topic describes the capabilities and functions of DFSMSHsm and the uses for DFSMSHsm control data sets. When you map these functions against the storage management needs of your installation, you might want to allocate certain subsets of functions to specific DFSMSHsm hosts. In this context, a host is a DFSMSHsm address space that performs automatic functions, such as migration. Secondary address spaces that perform aggregate backups and recovery (ABARS) belong to the main host and are not themselves considered hosts.

One z/OS image can have more than one DFSMSHsm host. In a sysplex, multiple z/OS images can each have one or more hosts, all sharing the same set of control data sets (an HSMplex).

There are two possible modes of host activity that you can specify at startup time for any host: HOSTMODE=MAIN (the default) and HOSTMODE=AUX.

HOSTMODE=MAIN

This (main) DFSMSHsm host

- Processes implicit requests, such as recalls, or deletes migrated data sets, from user address spaces
- Processes explicit commands from TSO, such as HSEND CMD and HBACKDS, as well as requests from batch jobs, such as ARCINBAK
- Manages ABARS secondary address spaces
- Allows MODIFY commands from a console or from an HSM Monitor/Tuner (HMT)
- Can run automatic backup, dump, and space management

HOSTMODE=AUX

This (auxiliary) DFSMSHsm host

- Allows MODIFY commands from a console or from an HSM Monitor/Tuner (HMT)
- Can run automatic backup, dump, or space management

In addition, another keyword PRIMARY identifies the primary host in an HSMplex.

Rule: Within one z/OS image, all started hosts must be running DFSMSHsm V2R10, or a later release.

Note:

1. Up to 39 DFSMSHsm hosts can be started concurrently in an HSMplex.
2. Within one z/OS image, any started host running on V2R10 or subsequent releases can share data with any host running on prior releases in the same HSMplex, but from a different z/OS image.
3. An HSMplex can consist of both single and multiple DFSMSHsm-host environments.

There are several advantages to starting more than one DFSMSHsm host in an z/OS image:

- Multiple DFSMSHsm address spaces mean less work per address space and less contention between functions, because each SYSZTIOT resource serializes only functions in its address space.
- Multiple DFSMSHsm address spaces mean that each address space that is doing some part of DFSMSHsm's work can have an appropriate MVS dispatching priority for that type of work.
- Multiple DFSMSHsm address spaces provide a larger number of tasks that perform any given DFSMSHsm function; for example, migration.
- DFSMSHsm functions that operate in more than one address space allow more MIPs that are allocated to DFSMSHsm functions.

Within a single z/OS image, you can have one of the following combinations, all sharing the same set of control data sets:

- One main host
- One main host and one or more auxiliary hosts

Note: Because you can start up or shut down hosts in any order, another permitted (but normally temporary) combination is one or more auxiliary hosts without a main host.

Related reading

For more information about the HOSTMODE and PRIMARY keywords, see the topic about DFSMSHsm multiple host environments in the *z/OS DFSMSHsm Implementation and Customization Guide*.

Starting DFSMSHsm

Before you can start DFSMSHsm, you must have a DFSMSHsm start procedure.

DFSMSHSM is used as the start procedure name. You can replace DFSMSHSM with any identification you designate for your computing system.

You can start DFSMSHsm in either of two ways:

- The first time you start DFSMSHsm after it is installed, you can load your system control program at initial program load. In response to the request:

IEA101A SPECIFY SYSTEM PARAMETERS

you reply with:

R 0,CLPA,CMD=nn

where CMD=nn identifies the startup member in the PARMLIB data set for COMMNDnn, which contains a START DFSMSHsm command. You add the START command for DFSMSHsm to your parameter library member, COMMNDnn, in the following manner:

COM='S DFSMSHSM,parameters'

- You can issue the START DFSMSHsm command from the system console after the system control program is loaded at initial program load. You do not have to load the system control program at initial program load to restart DFSMSHsm. The command to start DFSMSHsm is:

S DFSMSHSM,parameters

DFSMSHsm only runs as a started task; you cannot start DFSMSHsm as a batch job. If you plan to run more than one DFSMSHsm host in an z/OS image, you must issue the START command once for each host. At that time, you must specify a unique host identifier in the startup procedure, on the START command itself, or in an ARCSTRxx PARMLIB member that is referred to by the startup procedure or by the command.

Rule: To enable a multiple address space environment for DFSMSHsm, you must specify a new keyword HOSTMODE in the DFSMSHsm start procedure.

If you start more than one DFSMSHsm host in an image, the following conditions must exist:

- All the hosts must share the same set of control data sets; that is, they must be in the same HSMplex.
- All the hosts, therefore, share the same journal.
- Each host has its own log, activity logs, and PDA data sets.

To assist in problem analysis, you may want to incorporate the host-identifier characters into the names of the procedures that are used to start those hosts.

When DFSMSHsm gains control after a START command has been issued, a message (ARC0001I) is issued to notify the operator that DFSMSHsm is starting up. After the control data sets are opened and all major subtasks are attached, another message (ARC0008I) is issued to notify the operator that initialization is successful and that DFSMSHsm is ready to accept commands.

Messages appear at the system console for each DFSMSHsm command processed from the parameter library member if the monitoring of startup messages is in effect when the command is processed. These messages are only informational but could be useful if any problems occur. You start monitoring startup messages by specifying one of the following:

- COM='S DFSMSHSM,STARTUP=YES'
- SETSYS MONITOR(STARTUP)

You stop monitoring startup messages by specifying one of the following:

- COM='S DFSMSHSM,STARTUP=NO'
- SETSYS MONITOR(NOSTARTUP)

You can include the SETSYS command in the parameter library member as many times as you want. This allows you to control which commands issue messages to the system console when DFSMSHsm is starting.

You can install DFSMSHsm on multiple z/OS images of a sysplex or in multiple hosts in one z/OS image (or both). However, be sure that the HOST parameter of the startup procedure or the HOST parameter that is specified with the START command is unique for each instance of DFSMSHsm that shares the same set of control data sets.

Note: You cannot start the DFSMSHsm ABARS secondary address space from the operator's console by using an MVS START command. If you attempt this, DFSMSHsm fails the command.

Related reading

For more information about creating a start procedure, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Stopping DFSMSHsm

To stop DFSMSHsm, issue the following MODIFY command:

F DFSMSHSM,STOP

F DFSMSHSM,STOP without additional keywords causes no promotion of a secondary host.

Note: DFSMSHSM is the started procedure name. You can replace it with any identification you designate for your computing system.

You can also issue the MVS STOP DFSMSHSM command. When DFSMSHsm is stopped, the normal job-log information is spooled to the printer along with the DFSMSHsm log and activity logs that DFSMSHsm maintains. Either one of the STOP commands causes an orderly shutdown of the DFSMSHsm address space and an orderly closing of the DFSMSHsm control data sets.

F DFSMSHSM,STOP PROMOTE makes the host eligible to have its unique functions taken over by a promotable host. You can issue a STOP PROMOTE command on a primary host, or a host that runs secondary space management, in sysplex environment with secondary host promotion enabled. This causes an orderly shutdown, and allows another host to take over the unique functions of the stopped primary host, or the secondary space management functions of the stopped host running secondary space management.

F DFSMSHSM,STOP DUMP issues a dump and makes the host eligible to have its unique functions taken over by a promotable host.

In a sysplex, a manual shutdown of the primary or secondary space management host, without using the PROMOTE or DUMP parameters, does not automatically cause promotion of another host, because it is assumed that there is a reason for the shutdown, such as for routine maintenance or because all systems in the sysplex are being shut down.

An orderly shutdown means that each attached subtask ends and all DFSMSHsm secondary address spaces must complete processing. Because many DFSMSHsm commands and automatic functions are potentially long-running, DFSMSHsm periodically checks whether shutdown has been requested, to prevent a delay in the shutdown process. If DEFERMOUNT is used, or the correct tape was not mounted during end volume processing, and the timer was set by the MOUNTWAITTIME parameter of the SETSYS command, the timer expires and checks for shutdown. Volume functions check for shutdown before processing the next data set. After all subtasks have been detached and all secondary address spaces have completed processing, DFSMSHsm issues a message (ARC0002I) to notify the operator that shutdown is complete and returns control to the system.

When DFSMSHsm is shut down and restarted but MVS has not been reloaded at initial program load, user and HSEND CMD commands that did not complete are restarted as part of the DFSMSHsm initialization process. Commands entered from the system console that have not been processed are lost, and any request that exceeds the extended common service area (ECSA) limit per address space is also lost.

Related reading

For more information about the secondary host promotion function, see [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Shutdown processing on the common recall queue

When a host shuts down in a common recall queue (CRQ) environment, WAIT recall requests on the CRQ fail with RC=65, but new NOWAIT requests replace the failed requests. The new requests have the same priority as the original and can be processed by any eligible host that is connected to the CRQ. NOWAIT requests remain available in the CRQ.

Preventing DFSMSHsm functions from running

You can prevent all or part of DFSMSHsm's functions from running without stopping DFSMSHsm. You can also interrupt all or part of the functions after they have begun running. For those functions that process on a volume basis (backup, dump, migration, and recover), you can choose whether to interrupt the function at the end of the data set currently being processed or at the end of the volume currently being processed. You use the HOLD command to prevent or interrupt DFSMSHsm processing.

To prevent or interrupt all DFSMSHsm processing, issue the following command:

```
HOLD ALL
```


Note: HOLD ALL does not control the hold level of the common recall queue (CRQ) placement and selection functions. To place a hold on the CRQ, specify HOLD COMMONQUEUE, with its optional subparameters.

To queue all aggregate backup processing and prevent any new commands from being processed, issue the following command:

```
HOLD ABACKUP
```

To queue all aggregate recovery processing and prevent any new commands from being processed, issue the following command:

```
HOLD ARECOVER
```

To interrupt AUDIT command processing after it finishes the current data set, issue the following command:

```
HOLD AUDIT
```

To interrupt or prevent all backup processing, issue the following command:

```
HOLD BACKUP
```

Backup is held by this command at the end of the data set currently being processed. To hold automatic backup processing but allow command backup processing, issue the following command:

```
HOLD BACKUP(AUTO)
```

Automatic backup is stopped by this command at the end of the data set currently being processed, but command backup processing continues and is allowed to start new data sets or volumes.

To interrupt or prevent backup processing at the end of the volume currently being processed, issue the following command:

```
HOLD BACKUP ENDOFVOLUME
```

To specify that a host should neither place requests on nor select requests from the common recall queue, issue the following command. After you issue HOLD COMMONQUEUE, any requests already in the CRQ remain available for processing by hosts that are connected to the CRQ. If no CRQ hosts are available, and the request is a WAIT, it fails. The originating host processes all recall requests that are received after you issue HOLD COMMONQUEUE.

```
HOLD COMMONQUEUE or HOLD COMMONQUEUE(RECALL)
```

To specify that the host should not place any additional requests onto the CRQ, issue the following command. After you issue this command, the host places all new recall requests in the local queue and continues to select requests to process from the CRQ. The host selects the highest priority request available in the local and common recall queues. When the highest priority request in the local queue and the highest priority request in the CRQ match, the host selects the request from the local queue. If you specify this command on all hosts, then all hosts place new recall requests in the local queue, but continue to select requests to process from the CRQ.

`HOLD COMMONQUEUE(RECALL(PLACEMENT))`

To specify that a host should continue to place but not select recall requests in the CRQ, issue the following command. After you issue this command, if a host places a request in the local queue because the CRQ is unavailable, the host processes that request. If you specify this command on all hosts, all WAIT requests in the CRQ fail because no host is available to process them.

`HOLD COMMONQUEUE(RECALL(SELECTION))`

To prevent or interrupt dump processing, issue the following command. Dump processing stops at the end of the volume currently being processed.

`HOLD DUMP`

To prevent or interrupt automatic dump processing, issue the following command. Automatic dump processing stops at the end of the volume currently being processed. Processing initiated by DUMP commands is allowed to start.

`HOLD DUMP(AUTO)`

To specify no data movement, issue the following command.

Note: The SETSYS EMERGENCY command does not affect the placement of recall requests onto the common recall queue by the host with the command in effect, as other active hosts can process those requests.

`SETSYS EMERGENCY`

To interrupt dump processing at the end of the current unit of work being processed (typically a number of tracks or a cylinder), issue the following command:

`HOLD DUMP ENDOFDATASET`

To interrupt expiration of backup data sets by the EXPIREBV command, issue the following command. Expiration of backup versions ends when the data set that is being processed expires.

`HOLD EXPIREBV`

To prevent fast replication from processing any new FRBACKUP commands, issue the following command. No new fast replication backup commands are selected for processing.

`HOLD FRBACKUP`

To prevent fast replication from processing any new FRRECOV commands, issue the following command. No new fast replication recover commands are selected for processing.

`HOLD FRRECOV`

To interrupt or prevent processing by the LIST command, issue the following command. LIST command processing stops immediately. This command can be useful if a list command is returning a large amount of data to your terminal and you want to interrupt the process. In such a case, the operator may be required to issue the HOLD command.

`HOLD LIST`

You can prevent DFSMSHsm from making entries in the DFSMSHsm log. Even though the log entries are prevented, DFSMSHsm continues to make entries in the activity logs. To prevent entries into the DFSMSHsm log, issue the following command:

```
HOLD LOG
```

To interrupt or prevent all migration processing, issue the following command. All migration stops at the end of the data set currently being processed.

```
HOLD MIGRATION
```

To interrupt or prevent only automatic volume space management and automatic secondary space management, issue the following command. Automatic volume space management and automatic secondary space management stops at the end of the data set currently being processed, but migration caused by commands is allowed to start.

```
HOLD MIGRATION(AUTO)
```

To interrupt migration at the end of the volume currently being processed, issue the following command:

```
HOLD MIGRATION ENDOFVOLUME
```

To prevent all recall processing, including no more processing from an already mounted migration tape, issue the following command.

Note: In a common recall queue (CRQ) environment, with the HOLD RECALL command in effect, the host is not eligible to select requests from the CRQ.

```
HOLD RECALL
```

To prevent all recall processing from tape volumes (except those from already mounted migration tapes) but allow recall processing from DASD volumes, issue the following command.

Note: In a CRQ environment, with the HOLD RECALL(TAPE) command in effect, the host is not eligible to select requests requiring tape from the CRQ.

```
HOLD RECALL(TAPE)
```

To prevent recall processing from tape volumes when the RECALL command is issued by a TSO user but to allow all other recall processing, issue the following command.

Note: In a CRQ environment, with the HOLD RECALL (TAPE(TSO)) command in effect, the host is not eligible to select requests requiring tape and TSO from the CRQ.

```
HOLD RECALL(TAPE(TSO))
```

To prevent all recover processing, including no more processing from an already mounted backup tape, issue the following command:

```
HOLD RECOVER
```

To prevent recover processing for data sets on tape backup (except those from already mounted backup tapes) and dump volumes but to allow recover processing for data sets on DASD backup volumes, issue the following command:


```
HOLD RECOVER(TAPEDATASET)
```

To interrupt or prevent recycle processing, issue the following command. Recycle processing stops at the end of the data set being processed.

```
HOLD RECYCLE
```

To interrupt or prevent REPORT command processing, issue the following command:

```
HOLD REPORT
```

Allowing previously prevented DFSMSHsm functions to run

You can use the RELEASE command to allow DFSMSHsm to run functions that you previously prevented from running, and to allow activity logs to be closed and produced as output.

To allow all held DFSMSHsm functions except logging to run, issue the following command. This command does not restart writing in the DFSMSHsm log and does not cause the activity logs to be closed and processed.

Note: RELEASE ALL does not control the releasing of holds on the common recall queue. You must instead specify RELEASE COMMONQUEUE, with its optional subparameters.

```
RELEASE ALL
```

To allow the ABACKUP command to be processed, issue the following command. Any queued ABACKUP commands are allowed to process.

```
RELEASE ABACKUP
```

To allow the ARECOVER command to be processed, issue the following command. Any queued ARECOVER commands are allowed to process.

```
RELEASE ARECOVER
```

To allow the AUDIT command to be processed, issue the following command. Any queued AUDIT commands are allowed to process. Any AUDIT command that was interrupted by a HOLD command does not restart.

```
RELEASE AUDIT
```

To release all held backup functions, issue the following command. All backup processing is allowed to start. If the time is still within the backup start window, automatic backup starts processing any functions that have not already been processed. Any volumes that were not already fully processed are candidates for processing if less than 24 hours has passed since automatic backup was last started from the beginning and if the time is still within the backup window.

```
RELEASE BACKUP
```

To allow automatic backup processing that was prevented by a HOLD BACKUP(AUTO) command, issue the following command. This command does not release automatic backup processing if automatic backup is prevented by a HOLD BACKUP command that prevents any backup processing. If the time is still within the backup start window, automatic backup starts processing any functions that had not been processed when backup was held or interrupted. Any volumes that were not already fully processed are candidates for processing if less than 24 hours has passed since automatic backup was last started from the beginning, and if the time is still within the backup window.

Note: If automatic backup is not scheduled to run on a particular day, you can use the BACKUP parameter of the RELEASE command to move backup versions (and backup migrated data sets). Automatic backup is not scheduled to run if you have specified SETSYS AUTOBACKUPSTART(0) or if you have specified N for this day in the backup cycle.

```
RELEASE BACKUP(AUTO)
```

To allow processing of data set or UNIX file backups to both tape and ML1 DASD that were prevented by a HOLD BACKUP(DSCOMMAND) command, issue the following command.

Note: Use of the BACKUP(DSCOMMAND) parameter allows data set or UNIX file backup by command processing. This command will not affect automatic backup processing.

```
RELEASE BACKUP(DSCOMMAND)
```

To allow processing of data set or UNIX file backups that are directed to ML1 DASD that were prevented by a HOLD BACKUP(DSCOMMAND(DASD)) command, issue the following command:

```
RELEASE BACKUP(DSCOMMAND(DASD))
```

To allow processing of data set or UNIX file backups that were directed to tape that were prevented by a HOLD BACKUP(DSCOMMAND(TAPE)) command, issue the following command:

```
RELEASE BACKUP(DSCOMMAND(TAPE))
```

To allow processing of data set or UNIX file backups that were directed to DASD that were prevented by a HOLD BACKUP(AUTO DSCOMMAND(DASD)) command, issue the following command:

```
RELEASE BACKUP(AUTO DSCOMMAND(DASD))
```

To allow processing of data set backups that were directed to tape if they were prevented by a HOLD BACKUP(AUTO DSCOMMAND(TAPE)) command, issue the following command.

Note: If you release a particular category, you will release the hold on that category's subcategory, as well. However, you cannot release a subcategory if the initial HOLD function occurred at a higher level. In contrast, you can release the higher level and then hold the complementary subcategory.

```
RELEASE BACKUP(AUTO DSCOMMAND(TAPE))
```

To release a common recall queue hold at the COMMONQUEUE level, holds at the RECALL level, and holds at the PLACEMENT and SELECTION level, issue the following command.

Note: When RELEASE COMMONQUEUE is issued, all unselected requests on the local queue move to the CRQ.

```
RELEASE COMMONQUEUE
```

If there is not a CRQ hold at the COMMONQUEUE level, to release a hold at the COMMONQUEUE(RECALL) level and a hold at the PLACEMENT and SELECTION level, issue the following command.

Note: When RELEASE COMMONQUEUE(RECALL) is issued, all unselected requests on the local queue move to the CRQ.

```
RELEASE COMMONQUEUE(RECALL)
```

To release a CRQ hold at the PLACEMENT level only if a hold is not at a higher level also, issue the following command.

Note: When RELEASE COMMONQUEUE(RECALL(PLACEMENT)) is issued, all unselected requests on the local queue move to the CRQ.

```
RELEASE COMMONQUEUE (RECALL (PLACEMENT))
```

To release a CRQ hold at the SELECTION level only if a hold is not at a higher level also, issue the following command:

```
RELEASE COMMONQUEUE (RECALL (SELECTION))
```

To allow dump processing to run, issue the following command. Any queued dump commands are allowed to process. If the time is still within the dump window, automatic dump starts processing any functions that had not been completed when the dump function was held.

```
RELEASE DUMP
```

To allow automatic dump processing to run if automatic dump was held by a HOLD DUMP(AUTO) command, issue the following command. If the time is still within the dump start window, automatic dump starts processing any functions that had not been completed when the dump function was held.

```
RELEASE DUMP (AUTO)
```

To allow expiration of backup versions if EXPIREBV was held by a HOLD EXPIREBV command, issue the following command. New EXPIREBV commands are processed. Any EXPIREBV processing that was interrupted by the HOLD EXPIREBV command is not restarted.

```
RELEASE EXPIREBV
```

To release the fast replication process by the FRBACKUP command, issue the following command.

```
RELEASE FRBACKUP
```

To release the fast replication process by the FRRECOV command, issue the following command.

```
RELEASE FRRECOV
```

To close the activity log data sets and process them, issue the following command. This command causes one of two actions.

- If the activity logs are directed to a SYSOUT data set, the activity logs are closed and printed.
- If the activity logs are directed to DASD data sets, the current DASD data sets are closed and new activity log data sets are allocated.

```
RELEASE HARDCOPY
```

To allow LIST command processing, issue the following command. Any queued LIST commands are allowed to process. Any list operation that was in process when list was held is not restarted.

```
RELEASE LIST
```

To allow entries to the DFSMSHsm log after the entries have been prevented, issue the following command:

```
RELEASE LOG
```


To allow all migration processing, issue the following command. Any queued MIGRATION commands are allowed to begin processing. If the time is still within the automatic primary space management or automatic secondary space management processing windows, any automatic primary space management or automatic secondary space management functions that were not completed when migration was held begin processing.

```
RELEASE MIGRATION
```

To allow automatic primary space management or automatic secondary space management processing if they were prevented by a HOLD MIGRATION(AUTO) command, issue the following command. If the time is still within the automatic primary space management or automatic secondary space management processing windows, any automatic primary space management or automatic secondary space management functions that were not completed when automatic primary space management or automatic secondary space management was held are started.

```
RELEASE MIGRATION(AUTO)
```

When DFSMSHsm is migrating data sets to both DASD and tape devices, an error condition can end migration to either of the device types without ending migration to the other. After the error is corrected, you can restart migration to the device that had the error by issuing one of the following commands:

```
RELEASE MIGRATION(AUTO)
RELEASE MIGRATION
```

To allow all recall processing, issue the following command. Any queued RECALL commands are allowed to process. In a common recall queue (CRQ) environment, RELEASE RECALL makes the host eligible to select requests from the CRQ.

```
RELEASE RECALL
```

To allow recall from tape data sets if recall was held by a HOLD RECALL(TAPE) command, issue the following command:

```
RELEASE RECALL(TAPE)
```

To allow recall from tape data sets that were requested by TSO users (if the recall was held by a HOLD RECALL(TAPE(TSO)) command, issue the following command:

```
RELEASE RECALL(TAPE(TSO))
```

To release the hold on recalls from DASD volumes, issue the following command:

```
RELEASE RECALL(DASD)
```

Note that there is no accompanying HOLD RECALL(DASD) command; instead, the RELEASE RECALL(DASD) command essentially converts a HOLD RECALL state to a HOLD RECALL(TAPE) state.

To allow all recover processing, issue the following command. Any queued RECOVER commands are allowed to process.

```
RELEASE RECOVER
```

To allow data sets or UNIX files to be recovered from tape volumes if recover processing was held by a HOLD RECOVER(TAPEDATASET) command, issue the following command:


```
RELEASE RECOVER(TAPEDATASET)
```

To allow recycle processing, issue the following command. Any queued recycle requests begin processing. Any volumes that were being recycled when recycle was held are not restarted.

```
RELEASE RECYCLE
```

To allow REPORT commands to be processed, issue the following command. Any queued report requests begin processing. Any report that was in process when reporting was held is not restarted.

Note: If DFSMSHsm functions are being held because of a journal-inhibited condition and that condition cannot be resolved at that time, the following DFSMSHsm commands can be issued to bypass journaling.

- SETSYS NOJOURNAL
- RELEASE XXX (for whatever function you want to release)
- SETSYS JOURNAL (after a successful CONTROLDATASET BACKUP and the journal reset is completed)

```
RELEASE REPORT
```



Attention: If journaling is bypassed, you cannot perform a CONTROLDATASET RECOVERY for that period of time the journal was inhibited. The journal updates are lost for that period of time.

Canceling DFSMSHsm requests

DFSMSHsm lets you cancel queued requests for:

- An individual command
- A user ID
- A data set name
- A UNIX file

Rule: If you are using the common recall queue, you must submit the cancel request on the same host that originated the recall request that you want to cancel.

Use the CANCEL command to identify the commands for cancellation. The only commands that can be canceled are those commands that have not started. Identification of the commands to cancel can be specified by the DFSMSHsm request number, user ID, or data set name or UNIX file. Use the QUERY command to find the request numbers, user IDs, data set names, or UNIX files. Use the QUERY ACTIVE command to determine which DFSMSHsm requests are currently being processed. Use the QUERY ACTIVE(TCBADDRESS) to obtain the Task Control Block addresses of active DFSMSHsm data movement tasks.

Exceptions:

1. Depending on whether you choose to back up data sets that are in use (see “Specifying how DFSMSHsm should respond to a data set in use” on page 191), DFSMSHsm during a BACKVOL can schedule multiple requests to retry backing up (BACKDS) data sets found to be open for update when backup was initially attempted. All requests scheduled during one BACKVOL are assigned the same request number originally assigned to the BACKVOL request itself. Canceling the request number cancels all requests. To cancel only one request, use the DATASETNAME parameter.
2. Data set recoveries scheduled during a volume recovery (or during APPLYINCREMENTAL processing following a volume restore) are assigned the *same* request number originally assigned to the volume recovery or volume restore request. Canceling that request number cancels *all* those requests. To cancel only one request, use the DATASETNAME parameter.

A CANCEL command with the USERID, DATASETNAME, or FILENAME parameters cancels all existing nonexecuting requests having the specified user ID or data set name or UNIX file name.

The following command cancels a single request:

```
CANCEL REQUEST(num)
```

If you issue the CANCEL REQUEST(*num*) command for an active FRRECOV DSNAME request that is processing more than one single data set, then all queued data set requests for that command fail and no additional requests are generated. All data set recovery MWEs generated by a single FRRECOV DSNAME command have the same request number. Hence, the CANCEL REQUEST(*num*) command cancels all queued data set recovery MWEs that originate from the same FRRECOV command.

If you issue the CANCEL REQUEST(*num*) command with a request number of the Fast Replication dumps, all dumps in progress are canceled, any queued DUMP MWEs fail, and no additional requests are generated. When a Fast Replication DUMP command is processed, individual volume DUMP MWEs with the same request number are generated for all volumes in each storage group defined to the copy pool.

The following command cancels the requests of a particular user:

```
CANCEL USERID(userid)
```

The following command cancels the requests for a particular data set:

```
CANCEL DATASETNAME(dsn)
```

Note:

- Depending on whether you choose to back up data sets that are in use (see [“Specifying how DFSMSHsm should respond to a data set in use” on page 191](#)), DFSMSHsm during autobackup can schedule multiple requests to retry backing up (BACKDS) data sets found to be open for update when backup was initially attempted. Each request scheduled during autobackup has a request number of zero, which you cannot specify with the REQUEST parameter. You should use the DATASETNAME parameter to cancel one of these requests.
- You can cancel individual data set MWEs generated from the same FRRECOV command by using the CANCEL DATASETNAME(*dsn*) command. You cannot use the CANCEL DATASETNAME command to cancel the main FRRECOV DSNAME MWE.

The following command cancels the requests for a particular UNIX file:

```
CANCEL FILENAME('pathname')
```

Note: The path name must be enclosed in single quotes and be an absolute path starting with a forward slash (/) character.

The following command cancels a particular ABARS secondary address space:

```
CANCEL SASINDEX(sasindex)
```

sasindex is an integer value between 1 and 64 received from the QUERY ACTIVE(TCBADDRESS) command. The CANCEL command may be controlled by RACF FACILITY class.

The following command cancels a particular active task:

```
CANCEL TCBADDRESS(X'tcbaddress')
```

tcbaddress is an 8-digit value received from the QUERY ACTIVE(TCBADDRESS) command. Volume level activity generally goes to the next data set after a data set task is canceled, so to also end the volume task you should issue the appropriate HOLD command before canceling the data movement task.

'TCB' is the first *tcbaddress* to cancel. If the function does not end, then a 'TCB2' address is given, and the *tcbaddress* in 'TCB2' can then be used. If the function still does not end and a 'TCB3' address is given, then the *tcbaddress* in 'TCB3' can be used. If the TCB value is '****?***', then no TCB was found.

When canceling a hung journal back task, allow any active CDS backup tasks to run to completion. If a CDS backup task is also hung, cancel it before canceling the hung journal backup task. The CANCEL command may be controlled by RACF FACILITY class.

The CANCEL and QUERY commands cannot be canceled.

Note: A backup or recover request for a directory or wildcard can result in many files being processed. Each of these requests has the same request number, so if you wish to cancel a long running directory request, specifying REQNUM will cancel all the files that haven't been started yet.

Altering the priority of DFSMSHsm queued requests

You can use the DFSMSHsm ALTERPRI command to alter the priority of queued requests. You can alter the priority of the following request types: ABACKUP, ARECOVER, BACKDS, BACKVOL, DELETE, FRBACKUP, FREEVOL, FRRECOV, MIGRATE, RECALL, and RECOVER. You cannot alter the priority of BACKVOL CDS commands and requests selected for processing. When you need to alter the priority of queued requests, you can use the ALTERPRI command. The ALTERPRI command affects queued requests only; it does not affect new requests.

The HIGH parameter of the ALTERPRI command, which is the default, alters the specified request so that it has the highest priority on its respective queue. Conversely, the LOW parameter alters the request so that it has the lowest priority on its respective queue. The mutually exclusive REQUEST, USER, DATASETNAME, or FILENAME parameters indicate which requests DFSMSHsm should re-prioritize. (Use the QUERY REQUEST command to determine the request number to issue on the ALTERPRI command.) DFSMSHsm re-prioritizes all queued requests that match the REQUEST, USERID, DATASETNAME, or FILENAME criteria specified on the ALTERPRI command.

The BACKVOL STORAGEGROUP, BACKVOL VOLUMES, FRBACKUP DUMP, FRBACKUP DUMPONLY, and FRRECOV DSNAME commands, and requests for single volume recovery from incremental backups, each generate multiple requests, all with the same request number. An ALTERPRI REQUEST command issued to re-prioritize one of these requests alters the priority of all requests generated for the command that are queued at the time of ALTERPRI processing.

Note: To re-prioritize a recall request on the common recall queue, issue the ALTERPRI command on the same host from where the recall request originates.

Note: The priority of one or more CDQ or CVQ queued requests can be changed for requests that originate in the host that issued the ALTERPRI command, and match the REQUEST or USERID filter criteria.

The DATASETNAME keyword is not supported when you issue an ALTERPRI command against a FRRECOV DSNAME request. When an ALTERPRI DSNAME command is issued, and a FRRECOV DSNAME request for the same data set exists on the queue during ALTERPRI processing, the FRRECOV DSNAME request is skipped. To alter the priority of a FRRECOV DSNAME request, use the REQUEST keyword.

The following command alters all requests for a particular data set so that they have the highest priority on their respective queues:

```
ALTERPRI DATASETNAME(dsname)
```

The following command alters all requests for a particular UNIX file so that they have the highest priority on their respective queues:

```
ALTERPRI FILENAME('pathname')
```

The following command alters all requests with a particular request number so that they have the highest priority on their respective queue:

```
ALTERPRI REQUEST(reqnum) HIGH
```

The following command alters all requests issued by a particular user so that they have the lowest priority on their respective queues:

ALTERPRI USERID(userid) LOW

Printing DFSMSHsm log information

The information in the DFSMSHsm log can be used to determine the status of the data sets and volumes in DFSMSHsm. Figure 85 on page 607 shows the DFSMSHsm log printing environment.

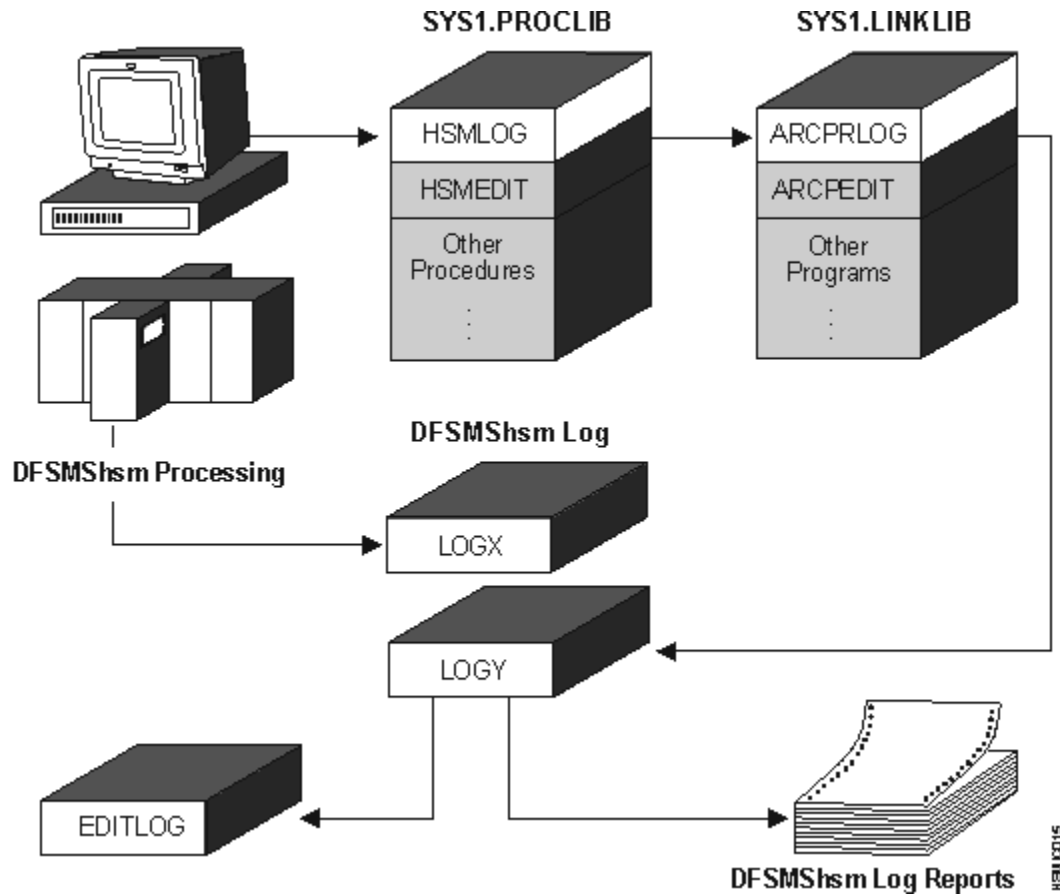


Figure 85. DFSMSHsm Log and Edit Log Overview

The DFSMSHsm log is actually two data sets, called LOGX and LOGY. LOGX is used to write the log information into, while LOGY is being processed for the information it contains. When LOGX becomes full, the data set receives a D37-04 ABEND indicating that it ran out of space. DFSMSHsm automatically switches the LOGX and LOGY data sets: LOGX becomes LOGY, and LOGY becomes LOGX. This switch allows access to the data recently recorded in the log.

The following messages are displayed on the operator's console:

```
IEC031I D37-04, IFG0554T, DFHSM, DFHSM, ARCLOGX,
        HSMLIB, DFHSM, HSMLGX
ARC0020I *****
ARC0024I DFSMSHsm LOGGING INHIBITED DUE TO EOV ERROR
ARC0024I (CONT.) ON LOG
ARC0020I *****
ARC0020I *****
ARC0027I DFSMSHsm LOG SWITCHED
ARC0020I *****
```

You can also force the logs to be switched by entering the SWAPLOG command. The following message are displayed:


```
ARC0020I *****
ARC0027I  DFSMSHsm LOG SWITCHED
ARC0020I *****
```

In either case, it is now possible to use the ARCPRLOG program and the ARCPEDIT programs to process LOGY. You use the ARCPRLOG program to do an initial formatting of the information in the log and to print the log.

If the LOGY data set or the edit-log data set is RACF-protected, you must take special steps before calling the HSMLOG procedure. The HSMLOG procedure processes the ARCPRLOG program to read, format, and print information in the LOGY data set and to insert selected log records into the edit-log data set. The HSMLOG procedure normally resides in SYS1.SAMPLIB in member ARCSTRST, and the ARCPRLOG program normally resides in the LINKLIB data set. After processing the ARCPRLOG program, the HSMLOG procedure empties the LOGY data set. If the HSMLOG procedure is called as a started task, the RACF started-procedures table (ICHRIN03) must be updated to include the procedure name (HSMLOG) and the RACF user ID to be associated with the procedure when it is started. If the HSMLOG procedure is called as part of a job, you must code `USER=userid` on the JCL JOB statement. In either case, the *userid* specified in the RACF started-procedures table or the *userid* specified on the JOB statement must have sufficient RACF access authority to read and update the LOGY data set.

If the edit-log data set is RACF-protected, you must take special steps before calling the HSMEDIT procedure. The HSMEDIT procedure processes the ARCPEDIT program to format and print the information in the edit-log data set. The HSMEDIT procedure normally resides in SYS1.SAMPLIB in member ARCSTRST, and the ARCPEDIT program normally resides in the LINKLIB data set. If the HSMEDIT procedure is called as a started task, the RACF started-procedures table must be updated to include the procedure name (HSMEDIT) and the RACF user ID to be associated with the procedure when it is started. If the HSMEDIT procedure is called as part of a job, you must code `USER=userid` on the OS/VS2 JCL JOB statement. In either case, the *userid* specified in the RACF started-procedures table or the *userid* specified on the JOB statement must have sufficient RACF access authority to read the edit-log data set.

The ARCPRLOG program selects the management work elements (MWEs) from all DFSMSHsm commands and the statistics records for each completed DFSMSHsm function for output to the ARCPEDIT program. You use the ARCPEDIT program to analyze, edit, and print the selected log records produced by the ARCPRLOG program.

The following example of JCL uses the ARCPRLOG and ARCPEDIT programs with the PROCLIB data set. Job HSMLOG initially analyzes, edits, and prints the information from the log and the LOGY data set.

```
//HSMLOG      JOB
//PRINTLOG    EXEC PGM=ARCPRLOG
//ARCPRINT    DD  SYSOUT=A
//ARCLOG      DD  DSN=DFHSM.HSMLOGY1,DISP=OLD
//ARCEdit     DD  DSN=DFHSM.EDITLOG,DISP=OLD
//*
//EMPTYLOG    EXEC PGM=IEBGENER
//SYSPRINT    DD  SYSOUT=A
//SYSIN       DD  DUMMY
//SYSUT2      DD  DSN=DFHSM.HSMLOGY1,DISP=OLD
//SYSUT1      DD  DUMMY,DCB=(DFHSM.HSMLOGY1)
//*
```

Note:

1. If the HSMLOGY1 data set is empty, do not use the ARCPRLOG program. If you do, unpredictable results can occur.
2. DSN=DFHSM.EDITLOG is only an example data set, created by the starter set. Specify your PS data set name, as appropriate.
3. Note that the log data set used as input to the ARCPRLOG program should not be compressed. This is because the log data set is created with RECFM=F but is opened by ARCPRLOG for update with RECFM=U. This is not allowed for compressed data sets. The input ARCLOG must be either the

HSMLOGY data set or an exact copy of HSMLOGY, with the same attributes: RECFM=F, LRECL=2048, BLKSIZE=2048.

4. The ARCPEDIT program must only be used for the selected records produced by the ARCPRLOG program.
5. Note that FSR records containing more than 144 tape volume entries are truncated when written to the LOGX/LOGY data sets. The formatted dump of each FSR in the ARCPRLOG output will include only the portion of the FSR that was written to the log. When the output volumes included in an FSR for RECYCLE have been truncated, the ARCPRLOG and ARCPEDIT output will display "TOVOL=*****".

Job EDITLOG formats and prints selected log records produced by the HSMLOG job.

```
//EDITLOG JOB
//EDITLOG PROC UID=DFHSM
//EDITLOG EXEC PGM=ARCPEDIT
//ARCPRINT DD SYSOUT=A
//ARCLOG DD DSN=&UID..EDITLOG,DISP=SHR
//
//*
```

Figure 86 on page 612, Figure 87 on page 613, and Figure 88 on page 614 show examples of the printed log output. The following is an alphabetical list of the information contained in the printed log output:

Field Name	Meaning
ABCC	System abnormal end completion code
ALLOC	Time of day allocated
APPLYINC	Indicates APPLYINCREMENTAL or not
ASCB	Pointer to the address space control block of the requester
AVGTIME	Average time for migration, per data set
BLKR	Number of 16K blocks read from a tape backup volume
BLKW	Number of 16K blocks written to a tape backup volume
BUFL	Length of buffer for the "control data set read" management work element
BUFU	Amount of buffer for the "control data set read" management work element
BYTR	Number of bytes read
BYTW	Number of bytes written
CCNT	Number of management work elements processed in the current request
CMDL	Length of the command
CMDO	Offset for TSO scan and parse
CMDT	Command text
COPY DSN	Data set name for the backup version
CPUSEC	Total job step time in milliseconds for the day
DARC	Dynamic allocation error return code
DATE	Date of the log record
DCOP FAIL	Number of requested dump copies that failed
DCOP REQ	Number of dump copies requested
DEVT	Device type as in the UCB
DFLGS	Dump restore flags
DLM	Date last migrated
DLU	Date last used

Field Name	Meaning
DSN	Data set name
DSORG	Data set organization
DUMP COPY REQUESTS	Number of dump copies requested
DUMPCLAS(ES)	DUMPCLASes listed or none
DUMPVOL	Dump volume serial
END	Time of day ended
FAILURES	Number of dump copy failures
FFLGS	Subsequent migration flags
FLGS	Flag bytes
FLG2	More DFSMSHsm flags
FLG3	More DFSMSHsm flags
FRVOL	Volume from which the backup version was copied
FRVSN	Volume from which the data set migrated
FUNC	Function description
GEN	Backup version generation number requested
GROUP	Group name for RACF
GROUP	Group name for RACF
HOST	Host identifier of an DFSMSHsm address space
ID	Log record code identification
JOBNM	Job name of the requesting job
KEY	DFSMSHsm record key
LEN	Length of the management work element
MCNT	Number of management work elements in the request
MIG	Migration
MSGID	Message identification
MWE FUNC	Function code of the management work element
NABEND	Number of DFSMSHsm subtask ABENDs for the day
NAME	Contents of the management work element
NDCOPFAIL	Number of dump copies that failed
NDCOPSSYS	Number of dump copies requested by the system
NDCOPSUSE	Number of dump copies requested by the user
NDSMIGFAIL	Number of failing migrate requests
NDSMIGL1	Number of migrates to ML1
NDSMIGL2	Number of migrates to ML2
NDSPW	New data set password for the recovered data set
NDSRECAFAIL	Recall failures
NDSRECAL1	Recalls from L1
NDSRECAL2	Recalls from L2
NDSRESTFAIL	Number of data set restores that failed
NDSRESTSREQ	Number of data set restores requested

Field Name	Meaning
NDUMPFAIL	Number of volume dumps that failed
NDUMPSSYS	Number of volume dumps requested by the system
NDUMPSUSE	Number of volume dumps requested by the user
NEWDSN	New data set name for the recovered data set
NGBRECA	Gigabytes read during recall
NGBYTMIG	Gigabytes read during migration
NI	Number of inserts in the message
NIPL	Number of DFSMSHsm startups for day
NKBRECA	Kilobytes read during recall
NKBYTMIG	Kilobytes read during migration
NMBRECA	Megabytes read during recall
NMBYTMIG	Megabytes read during migration
NMWE	Number of DFSMSHsm MWEs received
NSHUT	Number of DFSMSHsm shutdowns for day
NTBRECA	Terabytes read during recall
NTBYTMIG	Terabytes read during migration
NVOLRESTFAIL	Number of volume restores that failed
NVOLRESTSREQ	Number of volume restores requested
PW	Data set password
RACFLG	RACF flags
RC	Return code from the request
REAS	Reason code for the errors
RECFM	Record format
RETPD(S)	Retention period list
RQN	Request number
SEQN	Sequence number of this management work element in a multiple request
START	Time of day started
TO	Message directed to
TOD	Time of day
TOVOL	Volume to which the backup version was copied
TOVSN	Volume to which the data set migrated
TRKR	Number of tracks read
TRKW	Number of tracks written
TSCR	Create time stamp (always blank)
TSLU	Last update time stamp (always blank)
TYPE	DFSMSHsm record type
UID	User identification
VER	Backup version number requested
VSN	Volume serial number

Note: The TRKR and TRKW fields are approximate values for the number of tracks read and written and can vary, based on the blocking factors used.

```

LEN=0163 TOD=202630 DATE=95/06/07 ID=13 NAME=MWE COMPLETED
MWE FUNC=09 TOD=00:00:00 FLGS=000044 UID=B342236 RQN=0015 ASCB=F90100 RC=0000 REAS=0000
ABCC=00000000
CMDL=0111 CMD0=007 CMDT='SWAPLOG'
LEN=0182 TOD=202636 DATE=95/06/07 ID=04 NAME=ARCCCTL RCVE MWE
MWE FUNC=09 TOD=00:00:00 FLGS=200004 UID=B342236 RQN=0016 ASCB=F90100 RC=0000 REAS=0000
ABCC=00000000
CMDL=030 CMD0=000 CMDT='DEFINE DUMPCLASS(DCLASS01)'
LEN=0182 TOD=202636 DATE=95/06/07 ID=15 NAME=CATALOG CREATE OR UPDATE
CAT LEN=0170 TYPE=? TSLU= TSCR= KEY=|DCLASS01
00000000 00000000 00160000 F3F4F8F0 40404040 40404040 40404040 40404040
40404040 40404040 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000054
LEN=0084 TOD=202639 DATE=95/06/07 ID=03 NAME=OUTPUT MSG TO OPER/USER
MSGID=0216 TO=B342236 NI=3 ARC0216I DUMPCLASS DEFINITION MODIFIED, CLASS=DCLASS01 RC=0
LEN=0182 TOD=202639 DATE=95/06/07 ID=13 NAME=MWE COMPLETED
MWE FUNC=09 TOD=00:00:00 FLGS=000044 UID=B342236 RQN=0016 ASCB=F90100 RC=0000 REAS=0000
ABCC=00000000
CMDL=030 CMD0=026 CMDT='DEFINE DUMPCLASS(DCLASS01)'
LEN=0182 TOD=202644 DATE=95/06/07 ID=04 NAME=ARCCCTL RCVE MWE
MWE FUNC=09 TOD=00:00:00 FLGS=200004 UID=B342236 RQN=0017 ASCB=F90100 RC=0000 REAS=0000
ABCC=00000000
CMDL=030 CMD0=000 CMDT='DEFINE DUMPCLASS(DCLASS02)'
LEN=0182 TOD=202644 DATE=95/06/07 ID=15 NAME=CATALOG CREATE OR UPDATE
CAT LEN=0170 TYPE=? TSLU= TSCR= KEY=|DCLASS02
18000000 00000000 00000003 40404040 40404040 E3C8D9D6 E640C1E6 C1E84040
40404040 40404040 000A0000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000054
LEN=0084 TOD=202646 DATE=95/06/07 ID=03 NAME=OUTPUT MSG TO OPER/USER
MSGID=0216 TO=B342236 NI=3 ARC0216I DUMPCLASS DEFINITION MODIFIED, CLASS=DCLASS02 RC=0
LEN=0182 TOD=202646 DATE=95/06/07 ID=13 NAME=MWE COMPLETED
MWE FUNC=09 TOD=00:00:00 FLGS=000044 UID=B342236 RQN=0017 ASCB=F90100 RC=0000 REAS=0000
ABCC=00000000
CMDL=030 CMD0=026 CMDT='DEFINE DUMPCLASS(DCLASS02)'
LEN=0273 TOD=202650 DATE=95/06/07 ID=04 NAME=ARCCCTL RCVE MWE
MWE FUNC=09 TOD=00:00:00 FLGS=200004 UID=B342236 RQN=0018 ASCB=F90100 RC=0000 REAS=0000
ABCC=00000000
CMDL=030 CMD0=000 CMDT='BACKVOL VOL(MIG215) DUMP(DCLASS(BADDCL01,BADDCL02) RETPD(*,30)'
LEN=0273 TOD=202651 DATE=95/06/07 ID=13 NAME=MWE COMPLETED
MWE FUNC=09 TOD=00:00:00 FLGS=000044 UID=B342236 RQN=0018 ASCB=F90100 RC=0000 REAS=0000
ABCC=00000000
CMDL=121 CMD0=117 CMDT='BACKVOL VOL(MIG215) DUMP(DCLASS(BADDCL01,BADDCL02) RETPD(*,30)'
LEN=0110 TOD=202651 DATE=95/06/07 ID=03 NAME=OUTPUT MSG TO OPER/USER
MSGID=0184 TO=**OPER** NI=4 ARC0184I ERROR WHEN READING THE DFSMSHsm CONTROL DATA SET W RECORD
FOR BADDCL01, RC=0004
LEN=0091 TOD=202652 DATE=95/06/07 ID=03 NAME=OUTPUT MSG TO OPER/USER
MSGID=0643 TO=**OPER** NI=2 ARC0643I BACKVOL COMMAND REJECTED, DUMP CLASS BADDCL01 NOT DEFINED

```

Figure 86. Example of a List from the ARCPRLLOG Program


```

***** TOD=202630 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='SWAPLOG'

***** TOD=202639 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='DEFINE DUMPCLASS(DCLASS01)'

***** TOD=202646 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='DEFINE DUMPCLASS(DCLASS02)'

***** TOD=202651 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='BACKVOL VOL(MIG215)                DUMP(DCLASS(BADDCL01,BADDCL02) RETPD(*,30)'

***** TOD=202654 DATE=00095219 NAME=VOLUME DUMP ENDED *****
  FUNC=VOLUME DUMP          DSN=
  TOVOL=          FRVOL=MIG215 DEVT=00000000 DSORG=      RECFM=      START=202654 END=202654
  BATCH REQUEST      WAIT SPECIFIED
  USER REQUEST

***** TOD=202658 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='BACKVOL VOL(MIG215)                DUMP(DCLASS(DCLASS01,DCLASS02) RETPD(N,30)'

***** TOD=202946 DATE=00095219 NAME=VOLUME DUMP ENDED *****
  FUNC=VOLUME DUMP          DSN=
  TOVOL=          FRVOL=MIG215 DEVT=00000000 DSORG=      RECFM=      START=202659 END=202946
  BATCH REQUEST      WAIT SPECIFIED
  USER REQUEST

***** TOD=202954 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='RECOVER * TOVOL(MIG215) FROMDUMP(DCLAS(DCLASS01))          UNIT(3390)'

***** TOD=203000 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='RECOVER * TOVOL(MIG215) FROMDUMP(DVOL(DUTP1) AINC)          UNIT(3390)'

***** TOD=203002 DATE=00095219 NAME=VOLUME RESTORE ENDED *****
  FUNC=VOLUME OR DS RESTORE  DSN=
  TOVOL=MIG215 FRVOL=        DEVT=3010200E DSORG=      RECFM=      START=203001 END=203002
  BATCH REQUEST      WAIT SPECIFIED
  USER REQUEST

***** TOD=203004 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='RECOVER * TOVOL(MIG215) FROMDUMP(DGEN(1) AINC)          UNIT(3390)'

***** TOD=203006 DATE=00095219 NAME=VOLUME RESTORE ENDED *****
  FUNC=VOLUME OR DS RESTORE  DSN=
  TOVOL=MIG215 FRVOL=        DEVT=3010200E DSORG=      RECFM=      START=203004 END=203006
  BATCH REQUEST      WAIT SPECIFIED
  USER REQUEST

***** TOD=203007 DATE=00095219 NAME=MWE COMPLETED *****
MWE FUNC=09 FLGS=000044 UID=B342236
  CMDT='RECOVER * TOVOL(MIG215) FROMDUMP(AINC) DATE(95/01/01)          UNIT(3390)'

***** TOD=203009 DATE=00095219 NAME=VOLUME RESTORE ENDED *****
  FUNC=VOLUME OR DS RESTORE  DSN=
  TOVOL=MIG215 FRVOL=        DEVT=3010200E DSORG=      RECFM=      START=203008 END=203009
  BATCH REQUEST      WAIT SPECIFIED
  USER REQUEST

```

Figure 87. Example of a List from the ARCPEDIT Program

FUNCTION	DSNAME	WAIT	OPERANDS NOWAIT	THAT WERE VOLSER	SPECIFIED BY FUNCTION UNITNAME	MLEVEL2	REPLACE	FROMVOL	DCOP REQ	DCOP FAIL
MIGRATE L0->L1	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
MIGRATE L1->L2	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
MIGRATE L0->L2	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
RECALL L1->L0	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
RECALL L2->L0	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
DELETE MDS	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
BACKUP DATA SET	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
SPILL BACKUP COPY	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
RECOVER BACKUP COPY	00003	00000	00003	00000	00000	00000	00000	00000	00000	00000
RECYCLE DATA SET	00003	00000	00003	00000	00000	00000	00000	00000	00000	00000
DELETE BY AGE	00003	00000	00003	00000	00000	00000	00000	00000	00000	00000
RECYCLE MIG VOLUME	00003	00000	00003	00000	00000	00000	00000	00000	00000	00000
DUMP OF VOLUME	00000	00000	00003	00000	00000	00000	00000	00000	00009	00006
RESTORE VOLUME	00000	00000	00000	00000	00000	00000	00000	00000	00000	00000
RESTORE DATA SET	00006	00000	00006	00000	00000	00000	00000	00000	00000	00000

Figure 88. Example of a Summary List from the ARCPEDIT Program

Restarting DFSMSHsm after an abnormal end

DFSMSHsm automatically recovers after most abnormal terminations. However, if DFSMSHsm could not recover from the abnormal end and the RESTART keyword was used in the PROC statement of the startup procedure, DFSMSHsm will restart itself. The RESTART keyword allows you to specify a startup procedure—which can be the same or different from the original startup procedure—and to pass additional parameters to DFSMSHsm. RESTART is only valid in an ESA environment.

Should an abnormal end occur that interrupts MVS processing, DFSMSHsm can process any waiting requests as long as the extended common service area (ECSA) is not disturbed. When DFSMSHsm is restarted, it will process any requests that are waiting in the ECSA.

Recovery from an abnormal end could possibly extend long enough to cause checking for the need to restart to occur outside the automatic processing window for migration, backup, or dump. Should such an incident occur and should you want the automatic function to restart, issue one of the following commands (as appropriate for the function that you want to start):

```
SETSYS PRIMARYSPMGMTSTART (starttime,endingtime)
SETSYS SECONDARYSPMGMTSTART (starttime,endingtime)
SETSYS AUTOBACKUPSTART (starttime,lateststarttime,quiescetime)
SETSYS AUTODUMPSTART (starttime,lateststarttime,quiescetime)
```

Leave *starttime* as it is for normal operations. Change *endingtime*, *lateststarttime* and *quiescetime* to be later than the time that DFSMSHsm is restarted. *starttime* specifies the planned start time. *Endingtime* or *quiescetime* determine when DFSMSHsm stops processing new volumes for the automatic function being processed. Because you do not change the *starttime* specification, DFSMSHsm recognizes that you are not defining a new start window for the function. After the function has completed, you can issue a new SETSYS command to reset *endingtime*, *lateststarttime*, and *quiescetime* to their former values.

If a DFSMSHsm host fails in a sysplex environment where secondary host promotion is enabled, the unique functions of a primary host, or the secondary space management processing of any host are eligible to be taken over by a promoted host, when certain conditions are met.

If you are using the common recall queue (CRQ) when a host fails, all of the host's recall requests on the CRQ remain intact. They are eligible for processing by other active hosts, and the coupling facility notifies the remaining connected hosts of the failure. Also, requests that were in-process remain on the CRQ and are available for restart by other hosts.

Related reading

For more information about secondary host promotion, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Running automatic functions concurrently

Although the example system that has been used to explain the automatic functions of DFSMSHsm used different times for automatic backup, dump, and migration, any or all of the functions can run concurrently, either in the same host or in different hosts.

When running the automatic functions concurrently, you must remember that the first thing that automatic backup does on the primary host is to optionally back up the control data sets. While the control data sets are being backed up, automatic dump, automatic volume space management and automatic secondary space management are delayed. Also, if automatic dump, automatic volume space management or automatic secondary space management are running when automatic backup begins to back up the control data sets, backup of the control data sets is delayed until migration and dump tasks complete the volumes they are processing and automatic secondary space management completes one of its level functions. If automatic backup starts first, none of the other processes can start until the control data sets are backed up.

Note: Functions such as recycle, tape copy, and audit perform similarly to automatic functions in regards to contention with CDS backup.

Example: In the example system, assume that DFSMSHsm host 1 performs level functions and backup; DFSMSHsm host 2 performs migration; and DFSMSHsm host 3 performs dump. The commands provided are only those that would be particularly different from the examples that have been shown before.

First, the time when each of the DFSMSHsm hosts begins processing must be defined. Issue the following commands:

For DFSMSHsm host 1:

```
SETSYS AUTOBACKUPSTART(0001,0100,0200)
SETSYS SECONDARYSPMGMTSTART(0800,1600)
```

For DFSMSHsm host 2:

```
SETSYS PRIMARYSPMGMTSTART(0030,0500)
```

For DFSMSHsm host 3:

```
SETSYS AUTODUMPSTART(0030,0300,0500)
```

Automatic backup is started early in z/OS image 1 to allow it to back up the control data sets. After the control data sets are backed up, the other automatic functions are started.

For concurrent processing as it has been previously defined, the z/OS images that are to process each function must be defined to the storage groups. The MIGRATE SYSTEM/SYS GROUP NAME, BACKUP SYSTEM/SYS GROUP NAME, and DUMP SYSTEM/SYS GROUP NAME attributes are used in the storage groups to provide the definitions for SMS-managed volumes.

For example, the attributes in all storage groups would be set as follows:

```
BACKUP SYSTEM/SYS GROUP NAME ===> PRC1
```

```
MIGRATE SYSTEM/SYS GROUP NAME ===> PRC2
```

```
DUMP SYSTEM/SYS GROUP NAME ===> PRC3
```


Note: System group names can also be used for BACKUP SYSTEM/SYS GROUP NAME, MIGRATE SYSTEM/SYS GROUP NAME, and DUMP SYSTEM/SYS GROUP NAME.

Use the ADDVOL command to provide z/OS image definitions for non-SMS-managed volumes. For example, the following ADDVOL commands can be used to control the common volumes (GP0001, GP0002, GP0003).

For DFSMSHsm host 1:

```
ADDVOL GP0001 PRIMARY(NOAUTOMIGRATION, AUTOBACKUP, -  
NOAUTODUMP)  
ADDVOL GP0002 PRIMARY(NOAUTOMIGRATION, AUTOBACKUP, -  
NOAUTODUMP)  
ADDVOL GP0003 PRIMARY(NOAUTOMIGRATION, AUTOBACKUP, -  
NOAUTODUMP)
```

For DFSMSHsm host 2:

```
ADDVOL GP0001 PRIMARY(AUTOMIGRATION, NOAUTOBACKUP, -  
NOAUTODUMP)  
ADDVOL GP0002 PRIMARY(AUTOMIGRATION, NOAUTOBACKUP, -  
NOAUTODUMP)  
ADDVOL GP0003 PRIMARY(AUTOMIGRATION, NOAUTOBACKUP, -  
NOAUTODUMP)
```

For DFSMSHsm host 3:

```
ADDVOL GP0001 PRIMARY(NOAUTOMIGRATION, NOAUTOBACKUP, -  
AUTODUMP)  
ADDVOL GP0002 PRIMARY(NOAUTOMIGRATION, NOAUTOBACKUP, -  
AUTODUMP)  
ADDVOL GP0003 PRIMARY(NOAUTOMIGRATION, NOAUTOBACKUP, -  
AUTODUMP)
```

Updating the DFSMSHsm control data sets

You can use the UPDTCDS command to update specific fields within the control data sets. Specifically, you can update the expiration date of copy pool dump copies. DFSMSHsm updates all of the corresponding control data set records with the value specified on the UPDTCDS command.

To display the current versions, dump classes and expiration dates for existing dump copies associated with a copy pool, use the LIST COPYPOOL(*cpname*) command. You can then use this information to issue the UPDTCDS command to update an expiration date for a dump copy.

The following command updates the expiration date for copy pool COPYPOOL1, version 124, class ONSITE to 2050/12/31.

```
UPDTCDS COPYPOOL(COPYPOOL1) VERSION(124) DUMPEXPIRATION(DUMPCCLASS(ONSITE) NEWDATE(2050/12/31))
```


Chapter 18. RACF considerations

Throughout this document, the use of the Resource Access Control Facility (RACF) is recommended because DFSMSHsm requests RACF authorization before it allows access to data sets or UNIX files by non-DFSMSHsm-authorized users. RACF can also prevent access to DFSMSHsm-owned data sets except by users with specific RACF authorization.

In addition, by utilizing FACILITY class profiles, RACF can control command-level access for end-users and storage administrators.

You can use SECLABEL CLASS to control data set access. During initialization, DFSMSHsm checks to see if SECLABEL CLASS is active. If SECLABEL CLASS or RACF is activated or deactivated after DFSMSHsm startup, you must stop and restart DFSMSHsm to reset the SECLABEL indicator.

All RACF requests from DFSMSHsm are made through the system authorization facility (SAF). SAF conditionally directs control to RACF (if present) or to a user-supplied processing routine. RACF is determined to be installed by inspection of control block fields. If a RACF-equivalent product sets these fields the same as RACF, DFSMSHsm operates as if RACF were installed. Data management checks the user's authority during data set open when the user attempts to access the data.

Related reading

- For more information about RACF FACILITY class profiles, see [“Controlling command access with RACF FACILITY class” on page 625 or *z/OS DFSMSHsm Implementation and Customization Guide*.](#)
- For more information about the system authorization facility, see [z/OS MVS Programming: Assembler Services Guide](#).

RACF basics

To understand the relationship between RACF and DFSMSHsm, you must understand the RACF concepts of RACF profiles, RACF indication, always-call support, and protect-all environment.

RACF data set profiles

RACF controls data set access by identifying and verifying authorized users and by preventing unauthorized users from accessing data. RACF uses the information in a data set profile to determine whether a user is authorized to access the data set.

RACF has two types of RACF data set profiles: *generic* and *discrete*. The generic data set profile describes one or more data sets that have a similar name structure. For example, all data sets with the high-level name qualifier 'JJJ' can be described in a single generic data set profile. The discrete data set profile describes a specific data set on a particular volume.

Generic profile

RACF TSO commands create and maintain generic data set profiles. Generic profiles are never dynamically created, copied, renamed, or deleted by the operating system or DFSMSHsm. You must activate generic profile support with a RACF SETROPTS command before RACF can use generic profiles for authorization checking.

Generic profiles are designed to work with data sets that conform to a particular naming convention. A single access list then describes the access authorities to all those data sets.

Examples: The following are three generic profiles:

GRP12.LIB.*	Access list A
GRP12.*.CNTL	Access list B

GRP12.*

Access list C

When RACF searches a profile for data set GRP12.SRC.CNTL, it finds that the first profile does not match. The second profile does match; RACF verifies the user's authority with access list B. Note that the third profile also matches, but because it is less specific than the second, RACF does not use it.

In a generic profile definition, an * (asterisk) means any characters that are acceptable to the syntax of a data set name.

Generic profiles contain the following elements:

- Allow for security management by exception. In most cases, data sets that share the same naming conventions have the same security requirements. One profile can apply to many data sets with the same security requirements.
- Are more static than discrete profiles. They are not deleted when a data set is deleted.
- Make data set protection insensitive to DASD migration.
- Enhance performance as compared to discrete profiles. The most recently used generic profiles are kept in storage and are more often reused. Fewer updates to the RACF data set are needed.

Discrete data set profiles

Discrete data set profiles provide a one-to-one relationship between a specific data set on a specific volume and its security authorization requirements. To maintain the one-to-one relationship, discrete profiles are:

- Optionally created when a DASD data set is allocated
- Renamed when the data set is renamed
- Deleted when the data set is scratched

RACF indication implies that a data set is protected by a discrete RACF data set profile.

Related reading

For more information about defining RACF discrete and generic data set profiles, see [z/OS Security Server RACF Security Administrator's Guide](#) and [z/OS Security Server RACF Command Language Reference](#).

RACF indication

A RACF-indicated data set is one whose RACF indicator is on. For a non-VSAM data set, the RACF indicator is a bit in the volume table of contents (VTOC) of the DASD volume on which the data set resides. For a VSAM data set, the RACF indicator is a bit in the catalog record. The indicator remains with the data set even if the data set is moved to another system. If the data set profile is not moved also or is somehow lost, a RACF security administrator must take some action before anyone can access the data set.

You can use the SETSYS RACFIND command to cause DFSMSHsm to set the RACF indicator in the VTOC entry for migrated data sets, or for backup versions if the original data sets are RACF-indicated when they are migrated or backed up. Use the RACFIND parameter to RACF protect DFSMSHsm-owned data sets when you use RACF discrete data set profiles.

Always-call support

A RACF always-call environment exists when all data accesses on behalf of non-DFSMSHsm-authorized users result in the invocation of RACF for authorization checking, regardless of the setting of the RACF indicator in the VTOC entry or catalog record. Always-call is a system-wide convention for protection that is available to data sets that are allocated on DASD. With always-call support, resource managers in the operating system call RACF, regardless of the setting of the RACF indicator in the data set's DSCB. Some of the resource managers of the operating system are:

- OPEN/CLOSE/EOF
- DADSM ALLOCATE/EXTEND/RENAME/SCRATCH
- IEHMOVE

For details on the security facilities in your system, see your Security Administrator.

Without an always-call environment, generic profiles do not necessarily provide protection for the data sets they cover, because an unauthorized user might be able to access the data sets if the RACF indicator is not on. In this environment, use only discrete profiles. When the RACFIND parameter of the SETSYS command is in effect, DFSMSHsm turns on the VTOC entry RACF indicator to prohibit access to the backup versions and migration copies of RACF-indicated data sets.

RACF indicator with always-call

RACF's response to resource management authorization requests depends on the RACF indicator setting.

If the authorized requester (the caller) informs RACF that it believes the data set is protected by a discrete profile (by using the RACFIND=YES parameter), RACF first looks for a discrete profile. If none is found, RACF checks for a matching generic profile. If no generic profile is found, RACF rejects the request.

If the authorized requester (the caller) informs RACF that it believes the data set is not protected by a discrete profile (by using the RACFIND=NO parameter), RACF checks only for a generic profile.

A fundamental difference exists between RACF-indicated and non-RACF-indicated data sets. If no profile is found, RACF denies requests for RACF-indicated data sets but allows access for non-RACF-indicated data sets.

RACF protect-all environment

You have the option to fail access to data sets that are covered by neither a discrete nor a generic RACF profile. This environment is specified with the RACF SETROPTS command and the PROTECTALL keyword. If this option is in effect, RACF returns a "not-authorized" return code when normal access checking or data set creation authorization checking is done. For the DFSMSHsm user, it means that failure caused by a lack of profile does not appear to be different from failure caused by a lack of authorization.

DFSMSHsm functions are affected as follows in a protect-all environment:

- Recall or deletion of migrated data sets
 - Automatic: DFSMSHsm performs authority checking to ensure that the user initiating automatic recall is at least authorized to use the data set.
 - Command: Security checking is performed when the request originates with a non-DFSMSHsm-authorized user. The request is failed if no profile covers the data set to be recalled or deleted. A generic profile can be created before requesting recall or deletion, or the ADDSD command can be used to simultaneously initiate an implicit recall and create a discrete profile.
- HALTERDS command: If the data set does not yet exist, a generic profile must be created, or the command fails when originated by a non-DFSMSHsm-authorized user.
- HRECOVER command: If a backup profile exists, DFSMSHsm creates a discrete profile. If no backup profile exists, create a discrete or generic profile before issuing the command, or the request fails.
- Remaining DFSMSHsm commands (HBDELETE, HBACKDS, HMIGRATE): Create a discrete or generic profile before issuing the command, or the request fails.

RACF environment for DFSMSHsm

This discussion assumes the following environment:

- All data sets are cataloged in catalogs
- RACF security is based on protection with generic profiles
- RACF always-call support has been established

DFSMSHsm and RACF concepts

Migrated data sets and backup versions are stored in a DFSMSHsm internal format. Data set names for migrated data sets and backup versions are different from the names of the corresponding data sets on the level 0 volume; thus the original RACF protection may not apply.

When DFSMSHsm migrates a cataloged DASD data set, it scratches the data set on the level 0 volume. However, it does not delete the catalog entry; Instead DFSMSHsm changes the original volume serial number to a dummy volume serial number named MIGRAT.

TSO users can explicitly recall (only with DFSMSHsm) their data sets to a level 0 volume if they are authorized by RACF.

In addition to automatic DFSMSHsm functions, TSO commands are available for users to selectively control DFSMSHsm space and availability management activities for their data sets. These TSO commands can also be issued by batch jobs using the terminal monitor program (TMP). Your installation may require users to identify themselves to RACF with the USER and PASSWORD parameters on the JOB card.

When executing a DFSMSHsm function started by a TSO command, DFSMSHsm asks RACF for authorization for the user, specifying the original data set name on the level 0 volume. Thus, when DFSMSHsm and TSO are used, backup and migration operations use the RACF-protection rules of their related data set on the level 0 volume. This occurs regardless of whether the migration or backup support is DASD or tape.

No access authorizations to the migrated or backed up data sets themselves are needed for users issuing DFSMSHsm commands.

Migrated or backed up data sets should not be accessed as normal MVS data sets. RACF generic profiles should be set up to protect these data sets by their real names on DASD, and RACF tape protection must protect the corresponding tape volumes.

Only DFSMSHsm should be authorized to use migrated and backed up data sets as MVS data sets. The RACF generic profiles for these data sets should contain an empty access list, and the default universal access authority (UACC) should be NONE. The DFSMSHsm started task does not need to be on the access lists, because DFSMSHsm sets itself up as a privileged user to RACF. It may be desirable to allow a few storage administrators on these access lists, to repair DASD or tape volumes. The authorization is automatically granted if these users have the RACF OPERATIONS attribute.

Failure to protect migrated and backed up data sets with generic profiles creates integrity exposures such as DFSMSHsm having a record of a backup version or migration copy of a data set when the version or copy has been deleted.

RACF authority needed to perform DFSMSHsm functions

DFSMSHsm bypasses any security checking during automatic volume space management, automatic secondary space management, and availability management.

Undirected automatic recall is caused by reference from JCL, or under TSO, to a cataloged data set that is, in fact, migrated. In such a reference, the target volume is not specified. Once recall has occurred, standard RACF protection applies through OPEN. [Table 53 on page 620](#) shows the authority needed by TSO users to issue DFSMSHsm commands.

Table 53. RACF Authorization Required for DFSMSHsm Functions

DFSMSHsm Function	RACF Resource Access Authority Required
Migrate a data set	UPDATE
Recall a data set	EXECUTE
Delete a migrated data set	ALTER
Back up a data set	UPDATE

Table 53. RACF Authorization Required for DFSMSHsm Functions (continued)

DFSMSHsm Function	RACF Resource Access Authority Required
Recover a backup version without specifying NEWNAME	ALTER
Recover a backup version and specify NEWNAME	READ to original data set; ALTER on the NEWNAME
Delete a backup version	ALTER
Change backup characteristics	ALTER
Aggregate backup	READ

Special considerations apply to the TSO user commands HBDELETE (to delete backup versions) and HALTERDS (to modify backup characteristics):

- **HBDELETE:** If the backed up data set does not exist and the data set was protected by a discrete profile, DFSMSDfp asked RACF to delete the profile when the data set was deleted. DFSMSHsm knows that the backed up data set was RACF-indicated. DFSMSHsm fails the request on this command if there is no backup profile or generic profile honoring access to the data set name, or if the requester does not have ALTER authority on either the generic profile or the backup profile.

If the backed up data set does not exist and if the data set was not RACF-indicated but was protected by a generic profile, DFSMSHsm fails the request only if a generic profile that matches the original data set name does exist and the user does not have ALTER authority on that profile.

- **HALTERDS:** A user might set up backup characteristics for a data set that does not exist yet. As in the preceding item, DFSMSHsm fails the request only if a generic profile that matches the original data set name exists and the requester does not have ALTER authority on that profile. If the data set exists, DFSMSHsm fails the request if no generic profile, discrete profile, or backup profile exists honoring access to the data set name or if the requester does not have ALTER authority on the generic, the discrete, or the backup profile.

RACF security bypassing for DFSMSHsm commands

The previously discussed TSO commands can also be issued by using the operator console. DFSMSHsm operator commands can be entered from the operator console or by a DFSMSHsm-authorized user (a user who has received the DFSMSHsm DATABASEAUTHORITY attribute) under TSO by using the HSEND CMD command. The command keywords of these DFSMSHsm commands are identical to the corresponding TSO commands, except that the leading H of these commands is omitted. For example, for the TSO HDELETE command, the operator enters the DELETE command or the authorized TSO user enters the HSEND CMD DELETE command.

In general there is no RACF checking for DFSMSHsm commands using the operator console. Therefore, the number of DFSMSHsm-authorized users should be limited. An installation can use the ISPF-supplied SVC to run authorized, thus providing a way to issue requests to DFSMSHsm. If DFSMSHsm TSO commands are issued by a user who runs MVS-authorized, no RACF checking is done. If RACF FACILITY class is being used to protect aggregates, a user ID representing the console user must be defined and authorized to the aggregate group profiles before issuing the ABACKUP or ARECOVER commands from a console, or processing fails.

Viewing DFSMSHsm Facility Class Access by using the Security Configuration Assistant (SCA) for z/OS Management Facility (z/OSMF)

z/OS Management Facility (z/OSMF) utility, Security Configuration Assistant (SCA), provides a visual framework for examining the different elements of DFSMSHsm security. The Security Configuration Assistant layout consists of tabbed sections and tabular reports that are expanded or compressed, as needed. This framework provides a comprehensive perspective on your DFSMSHsm security.

DFSMSHsm provides the necessary files that are needed by the Security Configuration Assistant. The necessary files are located in the following directory:

```
/usr/lpp/dfsms/hsm/samples/DFSMSHsm_Admin.json  
/usr/lpp/dfsms/hsm/samples/DFSMSHsm_User.json
```

The files must be copied to the configuration directory for z/OSMF Security Configuration Assistant and is at:

```
USER_DIR/configuration/security/ (defaults to  
/global/zosmf/configuration/security)
```

For information about enabling and using the Security Configuration Assistant, see:

- [Configure the Security Configuration Assistant service](#) in *IBM z/OS Management Facility Configuration Guide*.
- For information about how to use the Security Configuration Assistant, see the z/OSMF online help.

The RACF user ID for DFSMSHsm

DFSMSHsm must be able to bypass RACF during migration and backup of user data sets. You must, therefore, define DFSMSHsm to RACF and give it the necessary level of authority. You do this by defining an entry in the RACF started-procedures module ICHRIN03, which contains the DFSMSHsm startup procedure name and a user ID that you define to RACF by an ADDUSER command. Do not define this user ID with the automatic data set protection (ADSP) attribute.

The following is an example of the RACF command you can use to define DFSMSHsm to RACF.

```
ADDUSER (hsmid) DFLTGRP (grpname)
```

In this example, *hsmid* is the user ID specified in the RACF started-procedure module ICHRIN03, or the RACF user ID associated with the DFSMSHsm STARTED class profile. The default group name is not important, but if it is not specified, RACF sets it to the current connect group of the user issuing the command.

Notes:

1. The user who issues the ADDUSER command must have the RACF SPECIAL attribute.
2. The UID parameter in the DFSMSHsm startup procedure is not related to the RACF user ID.
3. If you are using remote sharing functions (RRSF) to propagate RACF commands to remote hosts, it is suggested that the RACF user ID for DFSMSHsm be defined with the SPECIAL and OPERATIONS attributes on all recipient systems.
4. It is recommended that you define the RACF user ID for DFSMSHsm with the OPERATIONS and SPECIAL attributes to ensure that HSM has the proper settings and its functions do not modify the ACEE.

RACF started task table

Because a started task has a system-generated JOB card and cannot, therefore, supply USER, GROUP, and PASSWORD information, it must be identified to RACF by a different mechanism. SYS1.LPALIB contains a module, ICHRIN03, that contains the information that RACF requires for started tasks, and you must add the name of the DFSMSHsm procedure (or procedures if there are more than one) to this module. You also need to add the name of the ABARS procedure and any DFSMSdss ASID started by DFSMSHsm to this module. In fact, this must be done for any started task that accesses DFSMSHsm data sets (for example, a customized external writer to save the DFSMSHsm log). The module is a nonexecutable table containing 32 bytes of character information for each entry. Each entry contains four 8-byte fields.

<i>procname</i>	<i>user ID</i>	<i>grpname</i>	Flag
-----------------	----------------	----------------	------

The first three fields must be left-justified and padded with blanks. The *gname* field can be left blank; RACF then uses the default specified on the ADDUSER command. The eight flag bytes are set to hexadecimal zeros. For example, if the DFSMSHsm startup procedure is called DFHSM and you want to define a user ID of HSMUSR for DFHSM, the format of the entry you must create is:

DFHSM	HSMUSR		8 x X'00'
-------	--------	--	-----------

See your security administrator to have the correct entries made in ICHRIN03.

Note: If you are using remote sharing functions (RRSF) to propagate RACF commands to remote hosts, see the notes under [“The RACF user ID for DFSMSHsm” on page 622](#).

An alternative method of using the RACF started task table to associate a RACF user ID with a started task is to use the RACF STARTED class method. The RACF STARTED class is a RACF general resource class.

To associate the HSMUSR user ID to the DFHSM procedure in the previous example, the following commands can be used to define a STARTED class profile:

1. **SETR GENERIC(STARTED)**
2. **RDEFINE STARTED DFHSM.* STDATA(USER(HSMUSR) PRIVILEGED(YES))**
3. **SETOPTS RACLIST(STARTED) REFRESH**

If security policies do not allow a customer to run HSM as privileged, the RACF STARTED class for the user identified with the DFHSM started procedure can be set to TRUSTED(YES).

```
RDEFINE STARTED DFHSM.* STDATA(USER(HSMUSR) TRUSTED(YES))
SETOPTS RACLIST(STARTED) REFRESH
```

Use of the TRUSTED attribute makes authorization requests for the started task subject to auditing. This is based on the audit options specified in SETOPTS LOGOPTIONS and the UAUDIT setting in the user ID profile for the HSM started task.

Defining the STARTED class profile with PRIVILEGED or TRUSTED ensures that HSM has the proper settings to perform functions without modifying the ACEE.

When the STARTED class profile is defined for the DFHSM or DFHSMABR procedures, the RACF user ID associated with it should be defined with the OPERATIONS and SPECIAL attributes to prevent the ACEE from being modified.

Always-call support for migrated data sets and backup copies

The DFSMSHsm option NORACFIND in the SETSYS command requires that you protect DASD migration copies and backups with generic profiles in an always-call environment. DFSMSHsm does not set the RACF indicator on for these data sets.

The SETSYS command has MIGRATEPREFIX and BACKUPPREFIX parameters that allow all migration data sets and backup data sets to have a high-level qualifier selected by the installation.

When the SETSYS MIGRATEPREFIX and BACKUPPREFIX parameters are used, you can define one or two generic profiles to protect all the migrated and backed up data sets. These profiles should have an empty (or nearly empty) access list and a UACC value of NONE. The following example shows how RACF definitions can be set up to protect the migrated and backed up data sets.

```
SETOPTS GENERIC (*)
ADDGROUP (prefix)
ADDGROUP (bprefix)
ADDSD 'prefix.*' UACC(NONE)
ADDSD 'bprefix.*' UACC(NONE)
```

For an explanation of *prefix* and *bprefix*, see [“Specifying the first qualifier for names of migrated data sets” on page 53](#) and [“Specifying the first qualifier for names of backup versions” on page 189](#).

- Set the system-wide RACF options to include generic profiles.

- If you choose MIGPREF as the migration prefix and BAKPREF as the backup prefix, generic profiles MIGPREF.* and BAKPREF.* protect all the DFSMSHsm migrated and backed up data sets.
- The ADDSD command builds a generic profile for MIGPREF and another for BAKPREF.

Migration and recall from DASD

When performing DFSMSHsm TSO commands for a user, DFSMSHsm never uses the real migrated data set name to ask RACF for authorization. It always asks for authorization on the original data set name on the original volume.

If the original data set was protected by a generic profile, DFSMSHsm asks for authorization with the RACFIND=NO option, causing RACF to check for generic profile protection. The same protection that was in effect for the original data set applies, regardless of whether a discrete or generic profile was in effect.

The same protection also continues to apply when a data set is moved to an ML2 volume.

Backup and recovery from DASD

As with migration, DFSMSHsm refers to the RACF profile of the active data set for all TSO user interactions with their backups.

To recover a RACF-indicated data set, the corresponding profile must exist. This can be a problem with discrete profiles because deleting the data set also deletes the discrete profile. When the discrete profile has been deleted, a new one must be rebuilt by an appropriately authorized person before the data set can be recovered by a non-DFSMSHsm-authorized user.

DFSMSHsm provides the capability to back up RACF discrete profiles when a RACF-indicated data set is backed up. If you specify the SETSYS PROFILEBACKUP command, the discrete profiles are backed up. Non-DFSMSHsm-authorized users can then recover deleted data sets that had discrete profiles, because DFSMSHsm recovers the backup profile if no discrete profile exists for a RACF-indicated data set.

When a cataloged non-VSAM data set is being recovered to another volume, DFSMSHsm RECOVER gives some assistance by updating the volume serial information of the original RACF discrete profile, if it still exists, or of the recovered backup profile.

When you specify the NEWNAME and REPLACE parameters to recover a VSAM data set, DFSMSHsm deletes any information about newly named data sets from the catalog before it recovers the data set. The newly named data set gets the catalog information that the original data set had, including password and RACF-indication. You must provide the proper RACF protection for the newly named data set.

With generic profiles, most of these considerations do not apply, and no extensive security administration is required. If a generic profile exists that covers data sets that were backed up with discrete profiles, the data sets can be recovered by non-DFSMSHsm-authorized users whether the discrete or backup profiles exist, provided the user is authorized access to the data sets.

Aggregate backup and recovery data set security

When an ABACKUP or ARECOVER command is issued, DFSMSHsm passes the RACF FACILITY class profile name for the command, along with the user's or console operator's user ID, to RACF for authorization checking. If authorization is denied by RACF, DFSMSHsm rejects the request to issue the command. If RACF responds that no profile is defined for the command, DFSMSHsm processes the request only if the user is a DFSMSHsm-authorized user or a console operator. If the user is not a DFSMSHsm-authorized user, DFSMSHsm fails the request to issue the command. See [“Checking authorization during aggregate backup” on page 340](#) for a discussion of comprehensive and restricted ABACKUP and ARECOVER commands.

Backing up RACF-indicated data sets

When an existing data set with a RACF discrete profile is backed up to DASD and the NORACFIND parameter of the SETSYS command has been specified, DFSMSHsm does not set the RACF indicator on in the VTOC entry of the backup version.

Authorizing the use of concurrent copy for data set backup commands

DFSMSHsm uses the STGADMIN.ADR.DUMP.CNCURRNT FACILITY class resource to limit concurrent copy use to those users who are authorized to use the CC option.

By default, all users can use the CC option. However, if RACF indicates a lack of authority, DFSMSHsm fails the data set backup request if the concurrent copy request is specified as REQUIRED, VIRTUALREQUIRED or CACHEREQUIRED. If the concurrent copy request is not specified as REQUIRED, VIRTUALREQUIRED or CACHEREQUIRED and RACF indicates a lack of authority, DFSMSHsm continues to back up the data set as if the concurrent copy keyword were not specified on the backup command.

Small data set packing data set security

Because data sets that migrate to small data set packing (SDSP) data sets are treated as records in the SDSP data set, they must be protected by applying security to the SDSP data set. The security applied to the SDSP data set prevents non-DFSMSHsm access to the data in the data sets treated as records.

To maintain data set security on your system's small data set packing data sets, RACF-protect them with a discrete or generic profile. Erase-on-scratch should be specified in the discrete or generic profile if erase-on-scratch is specified for DFSMSHsm data sets not in SDSP data sets.

If data sets on volumes managed by DFSMSHsm are protected by a password and you specified SETSYS SMALLDATASETPACKING, protect the SDSP data sets on ML1 volumes with a discrete or generic profile to extend the protection of the data sets contained in the SDSP data sets. DFSMSHsm can still access the data sets in a protected SDSP data set without being given access authority to the SDSP data set.

Uncataloging migrated data sets

The DELETE *data set name* NOSCRATCH command can be used to uncatalog migrated data sets. For this command to be used when RACF protection is used for DFSMSHsm-owned data sets, two conditions must exist:

- A RACF group of ARCCATGP must exist.
- The user who issues the DELETE NOSCRATCH command must be authorized to group ARCCATGP.

Controlling command access with RACF FACILITY class

RACF FACILITY class profiles, both generic and discrete, can be used to allow access to all DFSMSHsm commands. If the RACF FACILITY class is active, DFSMSHsm always uses it to protect all DFSMSHsm commands. If the RACF FACILITY class is not active, DFSMSHsm uses the AUTH command to protect all DFSMSHsm commands. The storage administrator must work with the security administrator to give authorization to user and storage administrator commands.

Related reading

For more information about using the RACF FACILITY class environment and the AUTH command environment, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Chapter 19. Health Checker for DFSMSHsm

The IBM Health Checker for z/OS includes checks for DFSMSHsm. These checks are designed to help you determine if DFSMSHsm is correctly configured and is consistent with IBM's recommendations.

[IBM Health Checker for z/OS User's Guide](#) explains each check provided by IBM.

Check registration

The first MAIN host to initialize on the z/OS image will register all DFSMSHsm Health Checks to Health Checker. Checks will run immediately after this registration is complete. Checks will run subsequently based on intervals and triggering events specific to each check.

In the event the MAIN DFSMSHsm host is not active when a check runs, the check will become disabled. The check is re-enabled when the MAIN DFSMSHsm host initializes.

Chapter 20. Data recovery scenarios

In the data loss and recovery scenarios that follow, a description of a problem is presented, followed by the corrective measures that you can use for that problem. When you have a data-loss situation, you can quickly scan our examples to see if one of the examples matches your problem. If so, you may be on your way to a speedy recovery.

Note: Many data loss situations can be avoided by performing administrative steps before DFSMSHsm implementation.

Case 1: Damaged CDS, full journal

It's a cold Monday morning when Rick, a storage administrator, arrives at work to find the following note on his desk:



Figure 89. Handwritten note from Steve to Rick

What does Rick do after he regains consciousness?

The first thing Rick does is review the DFSMSHsm command activity log to see what caused the ARC0026E journaling error. He sees that the ARC0026E error specifies an EOVS problem; the journal has been completely filled up. Rick then examines the backup activity log to find out what caused the journal to completely fill up. He sees that control data set (CDS) backups fail each time the BACKVOL CDS command is issued. Since the BACKVOL CDS command is failing, the journal is not getting backed up.

Rick also sees that IDCAMS has issued error messages indicating out-of-sequence records in the backup control data set (BCDS). The failures recorded in the backup activity log indicate a “broken” BCDS.

To review the backup activity log, Rick used the DFSMSHsm RELEASE HARDCOPY command. HARDCOPY is an optional parameter of the RELEASE command that is used to specify that DFSMSHsm close any of the four activity logs if they contain data, and either print the logs or send them to DASD. The DFSMSHsm SETSYS ACTLOGTYPE command determines whether the logs are printed or sent to DASD.

When the activity logs are closed, new copies of the logs are allocated for use.

Once Steve had reset the journal, DFSMSHsm was able to operate because DFSMSHsm addresses the virtual storage access method (VSAM) control intervals directly by key. Backup of the CDS failed because a valid VSAM sequence set is required to read CDSs in key sequence.

What happened to the BCDS? Two MVS images, both having DFSMSHsm installed, updated the shared BCDS concurrently. When this happened, duplicate records were created in the BCDS, and the index structure of the BCDS was damaged.

After Rick figures out what happened and what damage was done, he plans his salvage operation. Although the journal records were lost when Steve deleted and reallocated the journal, the same information still exists in the BCDS. The BCDS, however, has duplicate records. So the first thing Rick must do is to delete the duplicate records in the BCDS and back up the CDSs. To delete the duplicate records in the BCDS and back up the CDSs, Rick performs the following tasks:

1. Stops DFSMSHsm on all images.
2. Uses the IDCAMS REPRO command to copy the data component of the BCDS to a sequential file. This can be performed by one of the following methods:
 - a. Direct the IDCAMS REPRO command to make a copy of the data component of the BCDS, as in the following example:

```
REPRO IDS(HSM.BCDS.DATA) ODS(RICK.BCDS.DATA)
```

- b. Direct the IDCAMS REPRO command to make a copy of the BCDS by relative byte address, as in the following example:

```
REPRO IDS(HSM.BCDS) ODS(RICK.BCDS.DATA) FADDR(0)
```

- -
 3. Uses DFSORT to perform the following:
 - a. Sort the sequential file by key, eliminating records with duplicate keys. The record that was most recently updated, where there are duplicates, is the one that is retained.
 - b. Reload the BCDS. The damaged BCDS must be deleted and a new BCDS defined.
 4. Starts DFSMSHsm, and issues the DFSMSHsm BACKVOL CDS command to create a good copy of the BCDS and nullify the journal. The LRECL and RECORDSIZE used in the DEFINE will depend on how your BCDS was defined. Issue LISTCAT ENT('yourbcds') ALL to determine maximum record size. LISTCAT will show you AVGLRECL and MAXLRECL which could be the following:

Maximum record size:

- 6544 bytes, if you need 100 or less active backup versions of any data set, if you use dump class stacking values greater than 99, or if you use fast replication.
- 2093 bytes, if you need 29 or less active backup versions of any data set, or if you use dump class stacking values of 99 or less.
- 2040 bytes, if you need 29 or less active backup versions of any data set, or if you use dump class stacking values of 97 or less.

Average record size:

- When Maximum record size is 6544, RECORDSIZE(334 6544)
- When Maximum record size is 2093, RECORDSIZE(334 2093)
- When Maximum record size is 2040, RECORDSIZE(200 2040)

Rick used the following JCL to accomplish the preceding steps:

The example below uses 2040 as maximum record size. For sequential files created in STEP1 and STEP2, LRECL used will be maximum record size + 4 for record descriptor word(RDW). RDW is a 4 byte field describing the record. In this example, using 2040 as maximum record size, you would use 2044 for sequential files created in these two steps.

```
//RECOVER JOB
//*****
//PLEASE CHANGE ALL THE FOLLOWING TO THE APPROPRIATE VALUES:
//*?USERID -TO YOUR USER ID
//*?DFSMSHSM -TO THE DFSMSHSM PREFIX FOR YOUR BCDS
//*****
//*THE FIRST STEP WILL CREATE A COPY OF ALL OF THE RECORDS CONTAINED
//*IN THE DATA COMPONENT OF THE BCDS.THESE RECORDS WILL NOT BE IN
//*ORDER,AND MAY CONTAIN MULTIPLE RECORDS WITH THE SAME KEY.IN
//*ADDITION,A RECOVERY BCDSWILL BE CREATED IN THIS STEP.THE USER
//*MUST SUPPLY THE PROPER SPACE INFORMATION,VOLUME INFORMATION,AND
//*DATA SET INFORMATION FOR THE DEFINE CLUSTER.
//*****
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//DDIN1 DD DISP=SHR,DSN=?DFSMSHSM.BCDS.DATA
//DDOUT1 DD DISP=(,CATLG),UNIT=SYSDA,
// DCB=(LRECL=2044,BLKSIZE=0,RECFM=VB),
// SPACE=(CYL,(50,10),RLSE),DSN=?USERID.BCDS.RECORDS
//SYSIN DD *
  REPRO INFILE(DDIN1)OUTFILE(DDOUT1)FADDR(0)
  DEFINE CLUSTER(NAME(?USERID.RECOVER.BCDS)-
    CYL(004)SPEED BUFFERSPACE(530432)-
    VOL(XXXXXX)-
    CISZ(12288)-
    FREESPACE(0 0)-
    KEYS(44 0)-
    RECORDSIZE(200 2040))-
    DATA(NAME(?USERID.RECOVER.BCDS.DATA))-
    INDEX(NAME(?USERID.RECOVER.BCDS.INDEX))
/*
//*****
//*THE NEXT STEP USES DFSORT AS FOLLOWS:
//*-THE SORT OPERATOR WILL SORT THE RECORDS EXTRACTED FROM
//*THE BCDS DATA COMPONENT.THEY WILL BE SORTED BY THEIR
//*VSAM KEYS,AND BY THE TIME STAMP OF THE LAST UPDATE MADE
//*TO THE RECORD.THE KEYS WILL BE SORTED IN ASCENDING ORDER,
//*AND THE UPDATE TIME IN DESCENDING ORDER;THE MOST CURRENT
//*UPDATED RECORD BEING FIRST IN THE OUTPUT DATA SET.
//*-THE SELECT OPERATOR (NEW IN DFSORT R12)WILL SORT THE
//*RECORDS BY THEIR VSAM KEYS.THE ORDER ESTABLISHED BY THE
//*SORT OPERATOR,WHERE THE MOST RECENTLY UPDATED VERSION OF
//*A DUPLICATE RECORD IS PLACED FIRST IN THE FILE,IS MAINTAINED.
//*THE DUPLICATE RECORD THAT IS NOT THE MOST RECENTLY UPDATED
//*VERSION IS DISCARDED.THE OUTPUT IS LOADED INTO THE RECOVERY
//*COPY OF THE BCDSTHAT WAS DEFINED IN THE FIRST STEP.
//*****
//STEP2 EXEC PGM=SORT,REGION=4096K,COND=(0,NE)
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?USERID.BCDS.RECORDS
//SORTOUT DD DISP=(,PASS),DSN=&&TFILE1,
// UNIT=SYSDA,SPACE=(CYL,(50,10),RLSE),
// DCB=(LRECL=2044,BLKSIZE=0,RECFM=VB,DSORG=PS)
//*SORT THE OLD BCDS BY ASCENDING SAM KEY AND
//*DESCENDING TIME STAMP OF LAST UPDATE.
//SYSIN DD *
  SORT FIELDS=(5,44,CH,A,53,8,BI,D)
/*
//STEP3 EXEC PGM=SORT,REGION=4096K,COND=(0,NE)
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=&&TFILE1
//SORTOUT DD DISP=OLD,DSN=?USERID.RECOVER.BCDS
//*LOAD ONLY THE MOST RECENTLY UPDATED RECORDS INTO THE NEW BCDS
//SYSIN DD *
  SORT FIELDS=(5,44,CH,A),EQUALS
  SUM FIELDS=(NONE)
/*
```

Note: The same procedure can be used for the MCDS and OCDS as well. Be sure to update the example JCL appropriately for the CDS being recovered. For example, the DEFINE CLUSTER definition to use will

depend on the CDS being recovered. For MCDS, BCDS and OCDS example definitions see [Starter set example](#) in *z/OS DFSMSHsm Implementation and Customization Guide*.

Case 2: Damaged journal, undamaged CDS

A damaged journal is often the result of a hardware-caused error on the journal volume. DFSMSHsm issues error message ARC0026E when it encounters an input/output (I/O) error in the journal data set. Message ARC0026E states “JOURNALING DISABLED DUE TO I/O ERROR. MIGRATION, BACKUP, DUMP, TAPECOPY, TAPEREP, RECYCLE, ARECOVER, AUDIT, AND EXPIREBV HELD.”

Once the problem occurs, follow this procedure:

1. Verify the DFSMSHsm CDSs are structurally sound.

Using IDCAMS, run EXAMINE against each of the CDSs. The procedure runs very quickly, and does not require that you bring DFSMSHsm down. The sample JCL to perform this procedure (which follows) comes from [z/OS DFSMS Access Method Services Commands](#).

```
//EXAMCDS JOB
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
EXAMINE -
  NAME(?hlqual.MCDS) -
  ERRORLIMIT(0)
//STEP2 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
EXAMINE -
  NAME(?hlqual.BCDS) -
  ERRORLIMIT(0)
//STEP3 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=A
//SYSIN DD *
EXAMINE -
  NAME(?hlqual.OCDS) -
  ERRORLIMIT(0)
```

If there are no errors reported by EXAMINE, continue to the next step.

If errors **are** reported by EXAMINE, do not proceed to the next step. Go to [“Case 1: Damaged CDS, full journal”](#) on page 629.

2. Stop DFSMSHsm on all images.
3. Do **not** copy the journal.
4. Rename the current journal.
5. Allocate a new journal with the old name. The following JCL allocates a new journal with the old name.

```
//JRNLDEF JOB
//STEP1 EXEC PGM=IEFBR14
//JRNLDD DD DISP=(,CATLG),UNIT=?3390,VOL=SER=?JNLVOL,
// SPACE=(CYL,(100),,CONTIG),DSN=?hlqual.JRNL
```

6. Start DFSMSHsm in EMERG=YES mode, a proclib option.
7. Issue the DFSMSHsm BACKVOL CDS command to keep your CDSs in synchronization with the new journal.
8. Delete the old journal.
9. Issue the DFSMSHsm SETSYS NOEMERGENCY command and then the DFSMSHsm RELEASE command to resume normal processing.

Case 3: Full journal, undamaged CDS

The problem of a full journal with an undamaged CDS occurs because the journal is not large enough or the CDSs are not being backed up often enough. DFSMSHsm issues error message ARC0909E when the

DFSMSHsm journal threshold is met, and continues to issue the error message each time an additional one percent (or more) of the journal is filled. When the journal is full, ARC0909E states: “JOURNAL DATA SET IS ABOUT 100% FULL.” When the journal is 100% full, message ARC0026E is also issued, which states: “JOURNALING DISABLED DUE TO EOV ERROR ON JOURNAL. MIGRATION, BACKUP, DUMP, TAPECOPY, TAPEREP, RECYCLE, ARECOVER, AUDIT, AND EXPIREBV HELD.”

Once the problem has occurred, take this step:

- Issue the BACKVOL CDS command. The BACKVOL CDS command will back up the journal and the control data sets, and null the journal.

Consider using a large format data set for the DFSMSHsm journal if your installation is not already doing so. A large format data set is a physical sequential data set with the ability to grow beyond 65,535 tracks per volume. Using a larger journal data set can allow more DFSMSHsm activity to take place between journal backups and helps to minimize occurrences of journal full conditions.

For information about large format data sets, see *z/OS DFSMS Using Data Sets*.

Case 4: Structurally damaged CDS, missing journal records

A structurally damaged CDS is typically caused by multiple updates occurring at the same instant in time within the same CDS. Simultaneous updates to the same CDS can occur in a multiple-MVS-image environment, if a job running on one image reorganizes the CDS while DFSMSHsm is still running on another image. Without proper share options and serialization methods, the problem of a structurally damaged CDS along with missing journal records can easily happen.

Problems arise when there are no CDS and journal backup versions before the damage. This commonly occurs when damage to a CDS is not immediately detected. As new CDS and journal backups are being created, the older versions are being deleted. By the time the damaged CDS is detected, the backup versions that were created before the error have been deleted.

When you have a structurally damaged CDS and missing journal records, you must work with the records currently in your damaged CDS. To repair your damaged CDS, follow the same steps that Rick followed in “Case 1: Damaged CDS, full journal” on page 629, substituting your damaged CDS for Rick's backup control data set.

Case 5: Time sequence gaps in CDS records, journal records also missing

During a reorganization, user error may cause time sequence gaps in the CDS records, along with missing journal records.

Take the case where an installation uses generation data sets (GDSs) when reorganizing the CDSs. The user attempts to use IDCAMS REPRO to create an output data set for a migration control data set (MCDS). The MCDS is not successfully copied to the output data set, but the user does not notice. Then the user employs IDCAMS REPRO to create a new, reorganized MCDS, not knowing that the input data set is actually the last version of the MCDS successfully copied and is not the current MCDS at all.

After the reorganization completes, the administrator thinks that everything went as planned, and continues with everyday processing.

DFSMSHsm may run with an old copy of the MCDS for a week before a high number of data set recall failures signal that there is a problem. The old MCDS that is being used, of course, does not contain the more recently migrated data sets.

When the storage administrator finally discovers what has happened, he looks to see if he can recover from a backup version. He backs up once a day, and keeps three backup versions. The reorganization, however, was performed seven days ago. That means he doesn't have any backups before when the reorganization occurred, for either the journal or the MCDS.

Recovery in this instance is very complicated. The best course of action is to call your IBM Area Storage Specialist for recovery assistance.

Case 6: Overwritten migration tape

Important: This recovery scenario applies only to an overwritten header label. If the damage to the tape is more extensive than an overwritten label, the following recovery procedure will not work. Also, if the tape has been encrypted, there is similarly no way to recover it. If the tape cannot be recovered due to hardware problems related to the drive itself, consider sending it to your tape drive manufacturer for recovery. This scenario applies only to native tape; it will not work for the recovery of data from virtual tape.

If you have a duplex or a tape copy, perform a tape replace. This procedure is described in [Chapter 69, “TAPEREREPL command: Replacing cartridge-type tape volumes with their alternate volumes,”](#) on page 1235.

In either case, you should route the recovery assistance through IBM. You may open a software problem report to diagnose the root cause of the damage, if it is not known.

Ensure that you have a backup copy of any data set that is going to tape (migration level 2). If a backup copy is available, you can recover from a backup version without having to recover from the damaged tape.

The following information is relevant to know or have available when you open a problem record:

1. Specific failure indications—full message text of any related message that was issued
2. Type of drives that you are using
3. Microcode levels for the drive and controller
4. Reasonably available history for the tape
5. Pervasiveness of the problem
6. Attempted recovery actions taken—RECALL, RECYCLE, DELVOL, and so forth.
7. Customers should gather the following information when the problem arises:
 - List TTOC of the affected volume
 - Library Manager Log for an ATL
 - CTRACE (if you are already writing to an external writer)
 - SYSLOG
 - DFSMSHsm PDA trace
 - A dump on the failure, if it can be trapped using a slip (ABEND637, for example).

In some cases, you cannot recover some or any of the data (depending on the extent of the damage). A backup of the tape ensures that this is not a concern. Back up the data, if the data matters, before going to migration level 2 (ML2).

The main reason for this type of damage is that the wrong tape volume is mounted to satisfy a mount request for a scratch tape. The problem typically remains undiscovered unless you need the overwritten tape volume to recall a data set.

Use the following recovery procedure to restore access to DFSMSHsm data on tape cartridges. This procedure will not work for reel-type tape because of a difference in file format.

1. Copy the damaged tape volume.

(This step is optional.) If the user data that was written over the DFSMSHsm data is wanted, you will need to create a copy of that data. The damaged tape volume will be returned to scratch or to DFSMSHsm for its later use, and the user data that was written on this volume will be lost at the end of this procedure.
2. Reinitialize the damaged tape volume.

Reinitialize the tape volume with the data set name that is used by DFSMSHsm for all of its tape volumes. Write a dummy data set with the first file on the volume named as below:

prefix.**HMIGTAPE.DATASET**

The *prefix* can be any valid qualifier. For example, you can use the migration prefix as specified on the SETSYS MIGRATEPREFIX command, or the UID. This step will create a good tape label on the damaged tape. Once the tape label is valid, the volume can be processed by RECYCLE in step “4” on page 636 of this procedure.

IEBGENER can be used to perform the reinitialization task.

The sample JCL that follows can be used to reinitialize a damaged migration volume. Modify the JCL to fit your needs by doing the following tasks:

- Change the UNIT parameter on the SYSUT2 DD statement, if necessary.
- Change *prefix* to the appropriate prefix.
- Change *ml2volser* to your volume serial number.

Included in the example that follows are the various messages that are generated from this job and the appropriate replies.



Attention: Do not perform the AUDIT MEDIACONTROLS function. AUDIT MEDIACONTROLS will remove all of your data.

```
//GENR      JOB
//STEP1     EXEC PGM=IEBGENER
//SYSUT1    DD DUMMY,LRECL=16384,BLKSIZE=16384,RECFM=F
//SYSUT2    DD DISP=(NEW,KEEP),DCB=*.SYSUT1,UNIT=?3480,
//          DSN=prefix.HMIGTAPE.DATASET,VOL=SER=?ml2volser
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
/*
```

J E S 2 J O B L O G

```

JOB09371  $HASP373 HAMMERTA STARTED - INIT 3 - CLASS A - SYS ..
JOB09371  *IEF233A M 5A8,HAMMRT,,HAMMERTA,STEP1
JOB09371  IEC512I LBL ERR 5A8,SMFDMP,SL,HAMMRT,SL,HAMMERTA,STEP1
JOB09371  *IEC507D E 5A8,SMFDMP,HAMMERTA,STEP1,DMP.T241T240.DATA
JOB09371  IEC507D REPLY 'U'-USE OR 'M'-UNLOAD
JOB09371  R 78,U
JOB09371  *IEC534D A 5A8,SMFDMP,SL,HAMMERTA,STEP1
JOB09371  IEC534D REPLY 'U'-USE OR 'M'-UNLOAD
JOB09371  R 84,U
JOB09371  *IEC704A L 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
JOB09371  IEC704A REPLY 'VOLSER,OWNER INFORMATION','M'OR'U'
JOB09371  R 87,'HAMMRT,DFSMSHSM'
JOB09371  IEC705I TAPE ON 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
JOB09371  IEF234E K 5A8,HAMMRT,PVT,HAMMERTA,STEP1
JOB09371  HAMMERTA STEP1 : 00:17:30/ 00:00:00/ 53 - R000
JOB09371  $HASP395 HAMMERTA ENDED

```

3. Determine if the first data set on the damaged tape volume spanned from another volume.

- Issue the LIST TTOC(*volser*) ODS(*dsname*) command to determine if your damaged tape volume is associated with a previous tape volume. If your damaged tape is associated with a previous tape, it will be listed under the heading “PREV VOL.” When there is a previous tape, it means that the first data set on your damaged tape began on the previous tape.

To verify whether your damaged tape is associated with a previous tape, enter the LIST TTOC(*prev_volser*) command and check to see that the last data set name on the previous tape is the same as the first data set name on your damaged tape. Note that if your installation is using extended tape table of contents, the LIST TTOC command might require an extended period of time to process each tape volume.

- Delete the first data set on your damaged tape volume if it spanned from a previous tape volume. To delete the first data set on a damaged migration tape, issue the following command:

```
DELETE dsn
```

- Issue the following command to recover the first data set that spanned to the previous volume:

```
RECOVER dsn
```


- Depending on the type of error that you are recovering from, it may be necessary to run RECYCLE several times. The following patch command causes DFSMSHsm to re-drive the POINT MACRO up to ten times when an I/O error is encountered during recycle, in order to recover data from a tape that has been overwritten and contains valid DFSMSHsm data beyond the tape marks created by the overwrite. The following patch value (X'0A') is appropriate for IBM 3590-technology tapes. Contact your tape-drive vendor for other tape technologies to determine the correct number of re-points.

```
HSEND PATCH .MCVT.+267 X'0A' VERIFY(.MCVT.+267 X'00')
```

At the end of the recovery procedures, issue the following patch command to prevent the re-drive of the POINT macro.

```
HSEND PATCH .MCVT.+267 X'00' VERIFY(.MCVT.+267 X'0A')
```

Issue the following command:

```
RECYCLE VOLUME(volser) EXECUTE
```

This step should be repeated to ensure that all available data sets are moved without the use of the FORCE parameter of the RECYCLE command.

Note: Readable, complete, and valid data sets are subsequently recycled to a new volume.

- Identify data sets that have been overwritten and build appropriate recovery commands.

Create a list showing which migrated data sets you need to recover (because the RECYCLE command was not able to remove them from the volume), by issuing the following command:

```
LIST TTOC(volser) ODS(dsn)
```

Build RECOVER commands for each data set on the list, but do not process the commands until step “7” on page 637.

An alternate to using the LIST TTOC command is to use DFSORT. The following is an example of a DFSORT job that identifies and builds RECOVER commands for each migrated data set that was overwritten. This job only builds the RECOVER commands; it does not process them.

```
//RCVCMDS JOB
//*****
/* Please change all the following to the appropriate values:
/* ?USERID - to your user id
/* ?DFSMSHsm - to the DFSMSHsm prefix for your MCDS
/* ?VLSER - to the volser of your damaged tape volume
//*****
/* This step uses SORT utility to build RECOVER
/* commands.
//*****
//STEP1 EXEC PGM=SORT,REGION=4096K
//*****
/* The INCLUDE statement selects MCD records (X'00'
/* at offset 51), that have the volser of your damaged tape and
/* have the MCDFASN bit on.
/* The OUTREC statement builds a RECOVER command for each
/* migrated data set found.
//*****
//SYSOUT DD SYSOUT=*
//MCDS DD DSN=?DFSMSHSM.MCDS,DISP=SHR
//CMDS DD DSN=?USERID.TEST.SORTMCD,DISP=(NEW,CATLG),
// DCB=(LRECL=255,BLKSIZE=0,RECFM=VB),
// SPACE=(CYL,(5,1),RLSE),UNIT=SYSDA
//SYSIN DD *
SORT FIELDS=(COPY)
INCLUDE COND=((51,1,BI,EQ,X'00'),AND,
              (69,6,CH,EQ,C'?VLSER'),AND,
              (75,1,BI,EQ,B'1.....'))
OUTREC FIELDS=(1,4,C' HSEND RECOVER ',5,44)
/*
```

- Issue the RECYCLE VOLUME(*volser*) FORCE EXECUTE command for the damaged volume.

The FORCE parameter of the RECYCLE command causes all data sets with errors to be deleted from the control data sets and the catalog. The deletion makes the tape empty, and the tape is removed from DFSMSHsm's inventory of tapes that contain data.

If the RECYCLE command fails with return code 16, you must issue a second RECYCLE VOLUME command to recycle the valid data sets while processing in single buffer mode. For an explanation of return code 16, refer to *z/OS MVS System Messages, Vol 2 (ARC-ASA)* and the table in the appendix titled "Reason Codes for Message ARC0734I when the Action is RECYCLE and Return Code is 16."

7. Issue the RECOVER commands for the migrated data sets that were lost. The RECOVER commands were generated in step "5" on page 636.

Case 7: Overwritten backup tape

This recovery scenario applies only to an overwritten header label. If the damage to the tape is more extensive than an overwritten label, the following recovery procedure will not work. Also, if the tape has been encrypted, there is similarly no way to recover it. If the tape cannot be recovered due to hardware problems related to the drive itself, consider sending it to your tape drive manufacturer for recovery. This scenario applies only to native tape; it will not work for the recovery of data from virtual tape.

If you have a duplex or a tape copy, perform a tape replace. You can reference this procedure in the Chapter 69, "TAPERPL command: Replacing cartridge-type tape volumes with their alternate volumes," on page 1235.

In either case, route the recovery assistance through IBM Support Center. Open a software problem report to diagnose the root cause of the damage, if it is not known.

When you open a problem record, you should be aware of the following information:

1. Specific failure indications—full message text of any related message issued
2. Type of drives
3. Microcode levels for the drive and controller
4. Reasonably available history for the tape
5. Pervasiveness of the problem
6. Attempted recovery actions taken—RECALL, RECYCLE, DELVOL, and so forth.
7. Customers should gather the following information when the problem arises:
 - List TTOC of the affected volume
 - Library Manager Log for an ATL
 - CTRACE (if already writing to an external writer)
 - SYSLOG
 - DFSMSHsm PDA trace
 - A dump on the failure if it can be trapped using a slip (for example, ABEND637).

The main reason for this type of damage is that the wrong tape volume gets mounted to satisfy a mount request for a scratch tape. The problem typically remains undiscovered unless you need the overwritten tape volume to recover a data set.

Use the following recovery procedure to restore access to DFSMSHsm data on single-file-format backup tapes that were partially overwritten by user data.

1. Copy the user data to a safe place.

(This step is optional.) If the user data that was written over the DFSMSHsm data is wanted, you will need to create a copy of that data. The damaged tape volume will be returned to scratch or to DFSMSHsm for its later use, and the user data that was written on this volume will be lost at the end of this procedure.
2. Reinitialize the damaged tape volume.

Reinitialize the tape volume with the data set name that is used by DFSMSHsm for all of its tape volumes. Write a dummy data set with the first file on the volume named as below:

prefix.**BACKTAPE.DATASET**

The *prefix* is one of the following:

- The backup prefix as specified on the SETSYS BACKUPPREFIX command
- The UID if prefixes are not specified with the SETSYS commands

This step will create a good tape label on the damaged tape. Once the tape label is valid, the volume can be processed by RECYCLE in step “5” on page 639 of this procedure.

IEBGENER can be used to perform the reinitialization task.

The sample JCL that follows can be used for a damaged backup volume. Modify the JCL to fit your needs by doing the following tasks:

- Change the VOL=SER parameter, as needed.
- Change the *prefix*, as needed.

Included in the example that follows are the various messages that are generated from this job and the appropriate replies.

```
//GENR      JOB
//STEP1     EXEC PGM=IEBGENER
//SYSUT1    DD DUMMY,LRECL=16384,BLKSIZE=16384,RECFM=F
//SYSUT2    DD DISP=(NEW,KEEP),DCB=*.SYSUT1,UNIT=?3480,
//          DSN=prefix.BACKTAPE.DATASET,VOL=SER=?m12volser
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD DUMMY
/*
                J E S 2   J O B   L O G

JOB09371  $HASP373 HAMMERTA STARTED - INIT 3 - CLASS A - SYS ..
JOB09371  *IEF233A M 5A8,HAMMRT,,HAMMERTA,STEP1
JOB09371  IEC512I LBL ERR 5A8,SMFDMP,SL,HAMMRT,SL,HAMMERTA,STEP1
JOB09371  *IEC507D E 5A8,SMFDMP,HAMMERTA,STEP1,DMP.T241T240.DATA
JOB09371  IEC507D REPLY 'U'-USE OR 'M'-UNLOAD
JOB09371  R 78,U
JOB09371  *IEC534D A 5A8,SMFDMP,SL,HAMMERTA,STEP1
JOB09371  IEC534D REPLY 'U'-USE OR 'M'-UNLOAD
JOB09371  R 84,U
JOB09371  *IEC704A L 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
JOB09371  IEC704A REPLY 'VOLSER,OWNER INFORMATION','M'OR'U'
JOB09371  R 87,'HAMMRT,DFSMSHSM'
JOB09371  IEC705I TAPE ON 5A8,HAMMRT,SL,NOCOMP,HAMMERTA,STEP1
JOB09371  IEF234E K 5A8,HAMMRT,PVT,HAMMERTA,STEP1
JOB09371  HAMMERTA STEP1 : 00:17:30/ 00:00:00/ 53 - R000
JOB09371  $HASP395 HAMMERTA ENDED
```

3. Determine if the first data set on the damaged tape volume spanned from another volume.

Issue the LIST TTOC(*volser*) ODS(*dsname*) command to determine if your damaged tape volume is associated with a previous tape volume. If your damaged tape is associated with a previous tape, it will be listed under the heading “PREV VOL.” When there is a previous tape, it means that the first data set on your damaged tape began on the previous tape.

To verify whether your damaged tape is associated with a previous tape, issue the LIST TTOC(*prev_volser*) command and check to see that the last data set name on the previous tape is the same as the first data set name on your damaged tape.

4. Delete the first data set on your damaged tape volume if it spanned from a previous volume.

Issue the following command:

```
BDELETE dsn VERSIONS(n)
```

The original data set name and the backup generation number will be required. To get this information, issue the FIXCDS command to display the MCC record for the backup version. The backup version

generation number is a 2-byte field contained in the MCCGEN field at offset X'46' in the FIXCDS display output. For example, issuing the following command:

```
HSEND FIXCDS C DFHSM.BACK.T500718.TONY3.AHRENS.I3006
```

gives you the following results:

```
MCH= 01602400 A6F7B92B F4BBC221 A6DAB593 9C8BBA03
*      7 4 B      *
+000 E3D6D5E8 F34BC1C8 D9C5D5E2 4BC2C3C4 E2404040 40404040 40404040 40404040
*TONY3.AHRENS.DATA      *
+020 40404040 40404040 40404040 E7E8E9F9 F0F50003 78048081 18075106 0093006F
*      XYZ905      *
+040 00081000 0000001C 000004DD 0121C0E2 0000243C 00003060 0000E3D6
D5E8F0F5
      .
      .
```

The following patch command causes DFSMSHsm to re-drive the POINT MACRO up to ten times when an I/O error is encountered during recycle in order to recover data from a tape that has been overwritten and contains valid DFSMSHsm data beyond the tape marks created by the overwrite.

```
HSEND PATCH .MCVT.+267 X'0A' VERIFY(.MCVT.+267 X'00')
```

At the end of the recovery procedures, issue the following patch command to prevent the re-drive of the POINT Macro.

```
HSEND PATCH .MCVT.+267 X'00' VERIFY(.MCVT.+267 X'0A')
```

Issue the following command to show all versions of the data set if the tape spanned before BDELETE:

```
LIST dsn
```

The output gives you the original data set name: TONY3.AHRENS.BCDS and the version number: +040 00081000 000000 **1C** 000004DD

The deletion would incorporate the information as follows:

```
BDELETE DSN(TONY3.AHRENS.DATA) VERSION(28)
```

The version number gets converted from 1C to decimal (28).

- Depending on the type of error that you are recovering from, it may be necessary to run RECYCLE several times. Issue the following command:

```
RECYCLE VOLUME(volser) EXECUTE
```

Note: Readable, complete, and valid data sets are recycled to a new volume.

Repeat this step to ensure that all available data sets are moved without the use of the FORCE parameter of the RECYCLE command.

- Identify data sets that were overwritten.

Build BACKDS commands for all the backup data sets that were overwritten and that are the most recent backup version. The following DFSORT job can identify and build the BACKDS commands for these data sets. This job only builds the BACKDS commands; it does not process them.

Note: The LRECL on how your BCDS was defined. Issue LISTCAT ENT('yourbcds') ALL to determine maximum record size. LISTCAT will show you MAXLRECL which could be the following:

Maximum record size:

- 6544 bytes, if you need 100 or less active backup versions of any data set, if you use dump class stacking values greater than 99, or if you use fast replication.
- 2093 bytes, if you need 29 or less active backup versions of any data set, or if you use dump class stacking values of 99 or less.

- 2040 bytes, if you need 29 or less active backup versions of any data set, or if you use dump class stacking values of 97 or less.

The following example uses 2040 as maximum record size. For sequential files created in STEP1, STEP2 and STEP3, LRECL used will be maximum record size + 4 for record descriptor word(RDW). RDW is a 4 byte field describing the record. In this example, using 2040 as maximum record size, you would use 2044 for sequential files created in these two steps.

```
//BDSCMDS JOB
//*****
//*?USERID -TO YOUR USER ID
//*?DFSMSHSM -TO THE DFSMSHSM PREFIX FOR YOUR BCDS
//*?VLSER -TO THE VLSER OF YOUR DAMAGED BACKUP VOLUME
//*****
//*THIS STEP USES SORT UTILITY TO FIND ALL
//*ASSOCIATED MCB AND MCC RECORDS,IDENTIFYING THE MOST RECENT
//*BACKUP VERSION THAT WAS OVERWRITTEN ON THE DAMAGED BACKUP
//*VOLUME.IT ALSO GENERATES THE BACKDS COMMANDS TO CREATE A
//*NEW BACKUP VERSION.
```

```
//*
//* STEP1 - Extracts records greater than or equal to X'00C1' or A.
//*
//*      First byte of the control data set record key. It
//*      represents the entry type for BCDS, MCDS, and OCDS
//*      records for all but 'A', 'B', 'C', and 'D' records.
//*
//* STEP2 - Extracts MCHTYPE='20'X, DATASET BACKUP ENTRY - CODE 'B'
//*      (MCB),DSN, and MCC data set backup version, and appends
//*      X'0001' count
//*
//* STEP3 - Extracts MCHTYPE='24'X, BACKUP COPY ENTRY - CODE 'C'
//*      (MCC), DSN, and MCC data set backup version, and appends
//*      X'0001' count
//*
//* STEP4 - SORT by data set name, sum fields on x'0001'.
//*      Whenever two records are found with equal sort or
//*      merge control fields, the contents of their summary
//*      fields are to be added, the sum (X'0002')is to be
//*      placed in one of the records, and the other record
//*      is to be deleted.
//*
//* STEP5 - When X'0002' is found, build HSEND BACKDS commands
//*****
//STEP1 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?DFSMSHSM.BCDS,AMP='BUFND=50'
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.BCDSSRT1,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=2044,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN DD *
SORT FIELDS=(COPY)
INCLUDE COND=(1,4,BI,GE,X'00C1')
/*
//STEP2 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?USERID.BCDSSRT1
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.BCDSSRT2,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=2044,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN DD *
SORT FIELDS=(COPY)
INCLUDE COND=(51,1,BI,EQ,X'20')
OUTREC FIELDS=(1,4,5,44,X,149,44,X'0001')
/*
//STEP3 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?USERID.BCDSSRT1
//SORTOUT DD DISP=(MOD,CATLG),DSN=?USERID.BCDSSRT2,
// DCB=(LRECL=2044,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN DD *
SORT FIELDS=(COPY)
INCLUDE COND=(51,1,BI,EQ,X'24',AND,113,6,CH,EQ,C'?VLSER')
OUTREC FIELDS=(1,4,69,44,X,5,44,X'0001')
/*
//STEP4 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
```



```
//SORTIN DD DISP=SHR,DSN=?USERID.BCDSST2
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.MERGED.BCDS.RECORDS,
// UNIT=SYSDA,SPACE=(CYL,(5,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN DD *
SORT FIELDS=(50,44,CH,A)
SUM FIELDS=(94,2,BI)
/*
//STEP5 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=?USERID.MERGED.BCDS.RECORDS,DISP=SHR
//BKDSREC DD DSN=?USERID.BKDS.RECORDS,DISP=(NEW,CATLG),
// DCB=(LRECL=255,BLKSIZE=0,RECFM=FB,DSORG=PS),
// SPACE=(TRK,(5,1),RLSE),UNIT=SYSDA
//*BUILD HSEND BACKDS COMMANDS.
//*****
//*THE INCLUDE STATEMENT SELECTS MCB RECORDS THAT CONTAIN
//*THE VOLSER OF YOUR DAMAGED ML1 VOLUME.
//*THE OUTREC STATEMENT BUILDS AN HSEND BACKDS COMMAND
//*FOR EACH DATA SET FOUND.
//*****
//SYSIN DD *
SORT FIELDS=(COPY)
OUTFIL FNAMES=BKDSREC,
      INCLUDE=(94,2,BI,EQ,X'0002'),CONVERT,
      OUTREC=(C'HSEND BACKDS ',5,44,255:X)
/*
```

7. Issue the DELVOL *volser* BACKUP(PURGE) command. The DELVOL ... (PURGE) command will perform the same functions as a RECYCLE ... FORCE command against the volume without causing a tape mount. The damaged backup tape volume will be deleted from DFSMSHsm inventory and any other control records describing the contents of the volume. DFSMSHsm deletes any association with a day in the backup cycle or with a set of spill backup volumes.
8. Issue the BACKDS commands that were generated in step “6” on page 639.

Note: If you are backing up a data set (with the BACKDS command) that resides on an ML2 tape volume, you need to use the RECALL command to recall the data set before issuing the BACKDS command.

Case 8: Damaged ML1 volumes

To handle a damaged Migration Level 1 (ML1) volume, perform the following steps:

1. Delete the ML1 volume entry of the damaged volume with the following command:

```
DELVOL volser MIGRATION(PURGE)
```

2. Initialize another DASD volume of the same device type as the damaged volume, using the volume serial number of the damaged volume.
3. Restore the damaged ML1 volume from a dump copy. If DFSMSHsm was used to create the dump copy, you can use the following command to restore the ML1 volume:

```
RECOVER * TOVOLUME(volser) UNIT(unittype) FROMDUMP
```

4. If there was a small data set packing (SDSP) data set on the damaged ML1 volume, check to see if there is a backup copy of the SDSP data set that is more recent than the dump copy that is used to restore the volume. If so, recover the backup copy of the SDSP. If an IDCAMS EXPORT command was used to create the backup copy, use the IDCAMS IMPORT command to recover the data set.
5. Add the restored ML1 volume to the list of volumes that DFSMSHsm owns. Use the ADDVOL command, specifying the DRAIN parameter. The DRAIN parameter prevents the volume from being selected for migration output. For example, issue the following command:

```
ADDVOL volser UNIT(unittype) MIGRATION(ML1 DRAIN)
```

If there was an SDSP data set on your ML1 volume, be sure to specify the SDSP parameter on your ADDVOL command. For example, issue the following command:


```
ADDVOL volser UNIT(unittype) MIGRATION(ML1 DRAIN SDSP)
```

6. Move all valid migration data sets and VTOC copy data sets off the ML1 volume with the following FREEVOL command:

```
FREEVOL MIGRATIONVOLUME(volser) AGE(0) TARGETLEVEL(ML1)
```

It is possible to generate tape mounts for ML2 tape volumes if there are valid, migrated, SMS-managed data sets that have reached the criteria for migrating to ML2.

7. Move all valid backup versions to backup volumes with the following command:

```
FREEVOL ML1BACKUPVERSIONS
```

The FREEVOL command causes all valid backup versions from all ML1 volumes to be moved to backup volumes.

8. Identify all lost migrated data sets.

The following is an example of a DFSORT job that identifies migrated data sets that are lost, and builds HDELETE and HRECOVER commands for each data set. This job only builds the HDELETE and HRECOVER commands; it does not process them.

```
//SMCD JOB
//*****
/* PLEASE CHANGE ALL THE FOLLOWING TO THE APPROPRIATE VALUES:
/* ?USERID - TO YOUR USER ID
/* ?DFSMSHSM - TO THE DFSMSHSM PREFIX FOR YOUR MCDS
/* ?VLSER - TO THE VLSER OF YOUR DAMAGED ML1 VOLUME
//*****
/* THIS STEP USES DFSORT TO LOCATE ALL MIGRATED DATASETS LOST
/* FROM THE DAMAGED ML1 VOLUME, AND BUILDS HDELETE & HRECOVER
/* COMMANDS FOR EACH DATA SET FOUND.
/* THE HDELETE COMMANDS ARE PLACED IN THE DATASET
/* ?USERID.HDELETE.DATA.
/* THE HRECOVER COMMANDS ARE PLACED IN THE DATASET
/* ?USERID.HRECOVER.DATA.
//*****
//STEP1 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=?DFSMSHSM.MCDS,DISP=SHR
//HDELETE DD DSN=?USERID.HDELETE.DATA,
// DISP=(NEW,CATLG),UNIT=SYSDA,
// DCB=(LRECL=255,BLKSIZE=0,RECFM=FB,DSORG=PS),
// SPACE=(TRK,(5,1),RLSE)
//HRECOVER DD DSN=?USERID.HRECOVER.DATA,
// DISP=(NEW,CATLG),UNIT=SYSDA,
// DCB=(LRECL=80,BLKSIZE=0,RECFM=FB,DSORG=PS),
// SPACE=(TRK,(5,1),RLSE)
//*****
/* THE INCLUDE STATEMENT SELECTS MCD RECORDS ('00'AT +51)
/* THAT CONTAIN THE VLSER OF YOUR RECOVERED ML1 VOLUME
/* (AT OFFSET 69),AND HAVE THE MCDFASN BIT ON (A VALUE GREATER
/* THAN X'80'AT OFFSET 75).
/* THE OUTREC STATEMENT BUILDS HDELETE AND HRECOVER COMMANDS
/* FOR EACH MCD RECORD SELECTED. THE DATA SET NAME IS COPIED
/* FROM THE MCD RECORD AT OFFSET 5.
//*****
/* BUILD HDELETE COMMANDS.
/* BUILD HRECOVER COMMANDS.
//SYSIN DD *
OPTION COPY
OUTFIL FNAMES=HDELETE,
INCLUDE=((51,1,BI,EQ,X'00'),AND,
(69,6,CH,EQ,C'?VLSER'),AND,
(75,1,BI,EQ,B'1.....')),
CONVERT,OUTREC=(C'HDELETE ',5,44,255:X)
OUTFIL FNAMES=HRECOVER,
INCLUDE=((51,1,BI,EQ,X'00'),AND,
(69,6,CH,EQ,C'?VLSER'),AND,
(75,1,BI,EQ,B'1.....')),
CONVERT,OUTREC=(C'HRECOVER ',5,44,80:X)
/*
```


9. Identify all lost backup versions.

The following is an example of a DFSORT job that identifies backup versions of data sets that are lost and builds HSEND BDELETE commands for those data sets. This job only builds the HSEND BDELETE commands; it does not process them.

```
//SMCC JOB
//*****
//* PLEASE CHANGE ALL THE FOLLOWING TO THE APPROPRIATE VALUES:
//* ?USERID - TO YOUR USER ID
//* ?DFSMSHSM - TO THE DFSMSHSM PREFIX FOR YOUR BCDS
//* ?VLSER - TO THE VOLSER OF YOUR DAMAGED ML1 VOLUME
//*****
//* THIS JOB EXTRACTS DATA FROM MCC RECORDS TO CREATE
//* HSEND BDELETE COMMANDS FOR ALL BACKUP VERSIONS THAT WERE
//* LOST FROM THE ML1 VOLUME.
//* THE HSEND BDELETE COMMANDS ARE PLACED IN DATA SET NAMED
//* ?USERID.BDEL.DATA.
//*****
//* STEP ONE USES THE SORT UTILITY TO EXTRACT MCC RECORDS
//* FROM BCDS THAT MATCH THE SEARCH CRITERIA.
//* OUTPUT DATA IS STORED IN A TEMP DATASET.
//*****
//STEP1 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=SHR,DSN=?DFSMSHSM.BCDS,AMP='BUFND=50'
//SORTOUT DD DISP=(NEW,PASS),DSN=&&SORTOUT1,
// UNIT=SYSDA,SPACE=(TRK,(15,5),RLSE),
// DCB=(LRECL=2048,BLKSIZE=23476,RECFM=VB)
//SYSIN DD *
        SORT FIELDS=COPY
        INCLUDE COND=(51,1,BI,EQ,X'24',AND,
                     113,6,CH,EQ,C'?VLSER')
        OPTION VLSHRT
//*****
//* THIS STEP USES SORT TO EXTRACT THE DSNAME AND HSM.DSNAME
//* FROM THE MCC RECORDS IN THE TEMP DATASET. THIS DATA IS
//* COMBINED WITH TEXT TO PRODUCE SKELETON COMMANDS IN FORMAT:
//* " HSEND BDELETE <DSNAME> VERSIONS(0000) <HSM.DSNAME>"
//* OUTPUT IN DATASET ?USERID.MCC.BDELETE
//*****
//STEP2 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=(OLD,PASS),DSN=&&SORTOUT1
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.MCC.BDELETE,
// UNIT=SYSDA,SPACE=(TRK,(10,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=0,RECFM=VB,DSORG=PS)
//SYSIN DD *
        SORT FIELDS=COPY
        OUTREC FIELDS=(1,4,C' HSEND BDELETE ',69,44,
                     C' VERSIONS(0000) ',5,44)
//*****
//* THIS STEP USES SORT TO EXTRACTS THE BINARY "VERSION" FIELD
//* AND HSM.DSNAME FROM THE MCC RECORDS IN THE TEMP DATASET.
//* OUTPUT OF THIS STEP IS TEMP DATASET &&VERS1
//*****
//STEP3 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=(OLD,PASS),DSN=&&SORTOUT1
//SORTOUT DD DISP=(NEW,PASS),DSN=&&VERS1,UNIT=SYSDA,
// SPACE=(TRK,(10,1),RLSE)
//SYSIN DD *
        SORT FIELDS=COPY
        OUTFIL CONVERT,OUTREC=(X,5,44,3X,139,2,BI,
                     EDIT=(+TTTTTTTTTTTTTTT),121:X)
```



```

//*****
//* THIS STEP USERS SORT TO FORMAT THE OUTPUT FROM THE PREVIOUS
//* STEP TO RESEMBLE THE HSEND BDELETE COMMANDS FROM STEP TWO.
//* OUTPUT IN DATASET ?USERID.MCC.VERS2
//*****
//STEP4 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=(OLD,PASS),DSN=&&VERS1,UNIT=SYSDA
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.MCC.VERS2,
// UNIT=SYSDA,SPACE=(TRK,(10,1),RLSE),
// DCB=(LRECL=251,BLKSIZE=0,RECFM=FA,DSORG=PS)
//SYSIN DD *
// SORT FIELDS=COPY
// OUTREC FIELDS=(69X,61,4,2X,2,44,251:X)
//*****
//* THIS STEP USES IDCAMS TO CONVERT THE FIXED-LENGTH RECORDS
//* OF ?USERID.MCC.VERS2 TO VARIABLE-LENGTH RECORDS COMPATIBLE
//* WITH ?USERID.MCC.BDELETE SO THESE TWO DATA SETS CAN BE
//* FURTHER PROCESSED TOGETHER.
//*****
//STEP5 EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//IN DD DISP=OLD,DSN=?USERID.MCC.VERS2
//OUT DD DISP=(NEW,CATLG),DSN=?USERID.VB2.VERSION,
// UNIT=SYSDA,SPACE=(TRK,(1,1),RLSE),
// DCB=(LRECL=255,BLKSIZE=23476,RECFM=VB)
//SYSIN DD *
// REPRO INFILE(IN) OUTFILE(OUT)
//
//*
//*****
//* THIS STEP USES SORT TO COMBINE THE .BDELETE AND .VERSION
//* DATASETS BASED ON THE HSM.DSNAME CONTROL FIELD.
//* THE VERSION FIELDS ARE SUM'D AND THE FIRST RECORD KEPT.
//* THEN THE HSM.DSNAME FIELD IS DROPPED LEAVING THE RECORDS
//* IN THE FINAL FORMAT:
//* " HSEND BDELETE <DSNAME> VERSIONS(XXXX) "
//* OUTPUT IN DATASET ?USERID.BDEL.DATA
//*****
//STEP6 EXEC PGM=SORT,REGION=4096K,COND=(0,NE)
//SYSOUT DD SYSOUT=*
//SORTIN DD DISP=OLD,DSN=?USERID.MCC.BDELETE
// DD DISP=OLD,DSN=?USERID.VB2.VERSION
//SORTOUT DD DISP=(NEW,CATLG),DSN=?USERID.BDEL.DATA,
// UNIT=SYSDA,SPACE=(TRK,(5,1),RLSE),
// DCB=*.SORTIN
//SYSIN DD *
// SORT FIELDS=(80,44,CH,A)
// SUM FIELDS=(74,4,ZD)
// OUTREC FIELDS=(1,4,5,75)
// OPTION ZDPRINT
//
//*

```

10. Identify which of the lost backup-version data sets were the most recent backup versions of the original data sets.

The following is an example of a DFSORT job that identifies these data sets, and also builds HSEND BACKDS commands for each of the identified data sets. This job only builds the HSEND BACKDS commands; it does not process them.

You will have to assess whether to create new backup versions at this time. You may also want to inform the owners of the data sets that the most recent backup versions of their data sets were lost.


```
//SMCB JOB
//*****
//*PLEASE CHANGE ALL THE FOLLOWING TO THE APPROPRIATE VALUES:
//* ?USERID - TO YOUR USER ID
//* ?DFSMSHSM - TO THE DFSMSHSM PREFIX FOR YOUR BCDS
//* ?VLSER - TO THE VLSER OF YOUR DAMAGED ML1 VOLUME
//*****
//* THIS STEP USES SORT UTILITY TO LOCATE THE MOST
//* RECENT OF ANY BACKUP VERSIONS THAT WERE LOST FROM THE
//* DAMAGED ML1 VOLUME.
//* IT BUILDS HSEND BACKDS COMMANDS FOR EACH BACKUP VERSION.
//* THE HSEND BACKDS COMMANDS ARE PLACED IN THE DATA SET
//* ?USERID.BACKDS.CMDS.
//*****
//STEP1 EXEC PGM=SORT,REGION=4096K
//SYSOUT DD SYSOUT=*
//SORTIN DD DSN=?DFSMSHSM.BCDS,DISP=SHR
//CMDS DD DSN=?USERID.BACKDS.CMDS,
// DISP=(NEW,CATLG), UNIT=SYSDA,
// DCB=(LRECL=255,BLKSIZE=0,RECFM=FB,DSORG=PS),
// SPACE=(TRK,(5,1),RLSE)
//* BUILD HSEND BACKDS COMMANDS.
//*****
//* THE INCLUDE STATEMENT SELECTS MCB RECORDS THAT CONTAIN
//* THE OUTREC STATEMENT BUILDS AN HSEND BACKDS COMMAND
//* FOR EACH DATA SET FOUND.
//*****
//SYSIN DD *
SORT FIELDS=(COPY)
OUTFIL FNAMES=CMDS,
INCLUDE=((51,1,BI,EQ,X'20'),AND,
(69,6,CH,EQ,C'?VLSER')),
CONVERT,OUTREC=(C' HSEND BACKDS ',5,44,255:X)
/*
```

11. Delete all lost migrated data sets by issuing the HDELETE commands that were generated in step “8” on page 642. This step corrects the MCDS so it no longer lists the lost migration data sets as residing on the ML1 volume, and deletes the catalog entries for the data sets.
12. Recover all lost migrated data sets by issuing the HRECOVER commands that were generated in step “8” on page 642. This step recovers the migrated data sets that were lost on the damaged ML1 from backup copies. The data sets are recovered to a level-0 volume.
13. Delete all backup versions that were lost by issuing the HSEND BDELETE commands that were generated in step “9” on page 643. This step corrects the BCDS so it no longer lists the lost backup version data sets as residing on the ML1 volume.
14. Identify all lost VTOC-copy data sets.

ISMF can be used to easily identify which VTOC-copy data sets are lost. The data sets with the following message are no longer on the ML1 volume:

```
DATA SET NOT FOUND ON VOLUME: volser
```

The backup VTOC-copy data set is named as below:

bprefix.VTOC.Tssmmhh.Vvolser.Dyyddd

The dump VTOC-copy data set is named as below:

bprefix.DUMPVTOC.Tssmmhh.Vvolser.Dyyddd

15. Delete the catalog entries for those VTOC-copy data sets that no longer exist on the ML1 volume.
16. Reinitialize the ML1 volume. This will delete all residual data.
17. Define an SDSP data set, if you want one on the ML1 volume.
 - a. If you had an SDSP data set on the damaged volume, you need to delete the catalog entry for that data set by issuing a DELETE NOSCRATCH command.
 - b. Issue the IDCAMS DEFINE command to define a new SDSP data set for the ML1 volume.
18. Change the ML1 volume so it can be selected for migration output by issuing the ADDVOL command with the NODRAIN parameter. For example, issue this command:


```
ADDVOL volser UNIT(unittype) MIGRATION(ML1 NODRAIN)
```

Be sure to specify other parameters as needed, such as SDSP and AUTODUMP.

Case 9: Reestablish access to previously deleted migrated data sets (no backup exists, ML2 only)

To reestablish access to previously deleted migrated data sets, perform the following steps:

1. Create a catalog entry for the data set. Use the IDCAMS DEFINE NONVSAM command with the VOLSER(MIGRAT) and UNIT(*tapeunit*) parameters.
 - a. If the data set is NON-VSAM, run IDCAMS DEFINE data set to VOLSER of MIGRAT to create the catalog entry.

Example JCL:

```
//IDCAMS JOB ,CLASS=A,MSGCLASS=H
//STEP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE NONVSAM -
    (NAME(dsname) -
    DEVT(device type) -
    VOL(MIGRAT))
/*
```

- b. If the data set is VSAM, use IDCAMS DEFINE base and data component to VOLSER of MIGRAT to create the catalog entry.

Example JCL:

```
//IDCAMS JOB ,CLASS=A,MSGCLASS=H
//STEP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE NONVSAM -
    (NAME(xxxxx.ksds) -
    DEVT(device type) -
    VOL(MIGRAT))
  DEFINE NONVSAM -
    (NAME(xxxxx.ksds.data) -
    DEVT(device type) -
    VOL(MIGRAT))
/*
```

Note: For a VSAM data set, define the NONVSAM entry for the base and data component.

2. The MCD record must be altered, if it exists, or created, if it does not exist.
 - a. Use the following command to display the MCD record, if it exists:

```
FIXCDS D data.set.name DISPLAY
```

Then issue the ASSIGNEDBIT command to turn on the bit, which indicates that the data set has migrated:

```
FIXCDS D data.set.name ASSIGNEDBIT(ON)
```

- b. If the MCD record does not exist (the FIXCDS DISPLAY command failed), the following AUDIT command will create it. The ML2 volser is needed for the audit command. If you do not know the volser, check in one of your journal backup versions to locate the journal entry of the MCD record for the migrated data set. Note the migration volume serial number, which is found at offset X'40' in the MCD record. Run the following command with the volser parameter of the ML2 volume:

```
AUDIT MEDCTL VOLUMES(ML2volser) FIX SERIALIZATION(DYNAMIC)
```


- c. For VSAM data sets, if AUDIT issues ERR169 with the data component name, then the MCD was created with the data component name instead of the base cluster name. As explained in the ERR169 message, change this name to the base cluster name before you perform a recall. Issue the following command to change it:

```
FIXCDS D data.component.name NEWKEY(base.name)
```

3. If the data set spans across tape volumes, you must repeat the AUDIT MEDCTL command for each ML2 volser and issue the following FIXCDS commands for the MCD record:

- a. Expand the MCD record:

```
FIXCDS D data.set.name EXPAND(z)
```

where z = number of volumes to which the data set spans $\times 6$

Example: Data set starts on volume **A** and spans to **B** and **C**; $z = 2 \times 6 = 12$. (2 = additional volumes, 6 = 6 characters for each *volser*)

- b. Add additional volsers to the MCDAVSNO array at offset X'194'

```
FIXCDS D data.set.name PATCH(X'194' volserxvolser)
```

where *volserx* = the first additional volume and *volser* equals the second additional volume.

- c. Alter MCDNVSN with the number of additional tape volumes and MCDNVSN0 with the offset of the list of volsers:

```
FIXCDS D data.set.name PATCH(X'D4' X'zzzz0194')
```

where *zzzz* = the number of *volsers* and 0194 equals the offset to the list of volsers as determined in the note below the following example.

Example: An example command altering the MCDNVSN

```
FIXCDS D data.set.name PATCH(X'D4'X'00020194')
```

Note: To determine if a data set has spanned to additional volumes, inspect the MCD record found in the journal backup. The number of "spanned to" volumes is found in a two-byte field starting at offset X'D4'. The location of the list of "spanned to" volumes is found at offset X'D6'. Offsets are given from the beginning of the MCD record. The MCD record found in the backup copy of the journal is appended at the beginning by a 10 byte MLOG, a 44 byte MCK, and a 20 byte MCH header record.

4. Recall the data set.
5. If you follow all steps correctly and you are still unable to recall the data set, contact IBM Support.

Case 10: Correcting errors within the common recall queue

The following indicate the corruption of the common recall queue (CRQ):

- DFSMSHsm issues message ARC1506E.
- DFSMSHsm issues message ARC1187E.
- DFSMSHsm does not select recall requests for processing.

For this last circumstance, issue the QUERY ACTIVE command and examine message ARC1541I. This message displays the factors that affect the selection of requests from the CRQ. A status of anything other than CONNECTED or a hold level of anything other than NONE is a probable reason why certain recall requests are not selected.

Note: An unexpected loss in connectivity to the CRQ may also introduce errors, but in most cases DFSMSHsm will be able to automatically correct those errors. When this occurs, you may see the multiple issuance of message ARC1102I. DFSMSHsm issues message ARC1102I when it attempts to recall a data

set that was previously recalled. Occurrences of this message at the time of a loss in connectivity is normal and not a problem with the CRQ list structure.

If you believe that the CRQ has become corrupted, issue the AUDIT COMMONQUEUE(RECALL) FIX command. When DFSMSHsm issues this command, it scans the entries in the CRQ and attempts to correct logical inconsistencies within the structure. Note that the audit function cannot correct all types of errors and may report zero errors, even though there are errors in the structure.

If messages ARC1506E and ARC1187E recur, or if certain data sets are not selected for recall processing after running the audit function, then perform the following procedure:

1. Capture a dump of the CRQ structure.

- a. As a general rule for dumps, include COUPLE and XESDATA information in SDUMPs.

To display the current options, issue the following command:

```
D D,0
```

If DFSMSHsm does not list COUPLE or XESDATA as options for SDUMP, then issue the following command:

```
CD SET,SDUMP=(COUPLE,XESDATA)
```

- b. Issue this dump command:

```
DUMP COMM=(CRQ LIST STRUCTURE)
```

See: * *id* IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

- c. Issue the following command:

```
R id,STRLIST=(STRNAME=SYSARC_basename_RCL,  
(LNUM=ALL,ADJ=CAP,EDATA=UNSER),CONT
```

where *basename* is the base name of the CRQ structure that the following command specified:

```
SETSYS COMMONQUEUE(RECALL(CONNECT(basename)))
```

See: * *id2* IEE094D SPECIFY OPERAND(S) FOR DUMP COMMAND

- d. Issue the following command:

```
R id2,LOCKE,(EMC=ALL),ACC=NOLIM),END
```

2. Reallocate the structure to remove the errors.

- a. Issue the SETSYS COMMONQUEUE(RECALL(DISC)) command on all DFSMSHsm hosts that are connected to the CRQ structure. This will move all recall requests from the CRQ back to the local DFSMSHsm hosts for processing. Each host will issue message ARC1502I after it disconnects from the structure.

The errors within the CRQ may prevent a particular DFSMSHsm host from disconnecting. If this is the case, DFSMSHsm cannot perform the remainder of the steps until that DFSMSHsm host shuts down.

Recommendation: Discontinue using the CRQ until the DFSMSHsm host can be shut down with minimal impact on other DFSMSHsm functions. At this point, no further action is required to prevent DFSMSHsm from attempting to place new requests onto the CRQ.

- b. Once all DFSMSHsm hosts disconnect from the common recall structure, delete the structure.

To determine if all DFSMSHsm hosts have disconnected, issue the following command:

```
D XCF,STR,STRNAME=SYSARC_basename_RCL
```

The number of connections reported should be zero.

To delete the structure, issue the following command:

```
SETXCF FORCE,STR,STRNAME=SYSARC_<basename>_RCL
```

- c. Issue the following command on each DFSMSHsm host that was previously connected:

```
SETSYS COMMONQUEUE(RECALL(CONNECT(<basename>)))
```

This command will cause each host to automatically move all of its local recall requests back onto the CRQ and to reallocate the structure.

3. Capture the Problem Determination Aid (PDA) data.

See [z/OS DFSMSHsm Diagnosis](#) for information on how to perform this action. The collected PDA includes the data from the time that a problem first occurred, through the time that the audit was performed.

4. Contact IBM Support to report the problem.

Case 11: Recovering a deleted ML1 data set without a backup

This procedure assumes that you have a DUMP of the ML1 volume and that you know when the ML1 data set was deleted. You must determine the ML1 volume and the HSM-generated name from either the journal or activity log. This scenario does not account for data sets in an SDSP on a ML1 volume.

To reestablish access to previously deleted migrated data sets, perform the following steps:

1. Determine the ML1 volume and HSM-generated name from the journal. Use the following jobs to obtain this information:
 - a. Copy the journal using IEBGENER before you use SORT.

```
//COPY JOB ,
//
//STEP1 EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSIN DD DUMMY
//SYSUT1 DD DISP=SHR,
//          DSN=journal name (the journal data set where the information
//                           about the deleted migrated data set resides)
//SYSUT2 DD DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=6556,BLKSIZE=13112,RECFM=VB),
//          SPACE=(CYL,(10,5),RLSE),
//          DSN= copied journal name
/*
```

- b. Process the SORT job.

```
//SORTANY JOB ,
//
//SORT1 EXEC PGM=SORT
//SYSPRINT DD SYSOUT=*
//SORKWK01 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORKWK02 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTIN DD DISP=SHR,
//          DSN=copied journal name
//SORTOUT DD DISP=(NEW,CATLG),UNIT=SYSDA,
//          DCB=(LRECL=121,BLKSIZE=27951,RECFM=FB),
//          DCB=(LRECL=32756,BLKSIZE=32760,RECFM=VB),
//          SPACE=(CYL,(500,5),RLSE),
//          DSN=output name
//MSGOUT1 DD SYSOUT=*
//SYSOUT DD SYSOUT=*
//SYSIN DD *
//          OPTION VLSHRT
//          MERGE FIELDS=COPY
//          RECORD TYPE=V
//          INCLUDE COND=((63,1,BI,EQ,X'00'),and,(17,1en,CH,EQ,C'dsn'))
//          END
/*
```


Substitute the length of the user data set name for *len* and the data set name for *dsn*. This job will retrieve the MCD records for the data set that was erroneously deleted.

You will have the journal records for the deleted data set in the sorted output data set. The ML1 volume can be determined by going to +76 into the journal record. The HSM-generated name will be at +232. Record the volser of the ML1 volume and the HSM-generated name.

2. Create a NON-VSAM catalog entry that specifies a DEVT(*device type*) and VOL(MIGRAT) where device type is the device type of the ML1 volume.

```
//IDCAMS JOB ,CLASS=A,MSGCLASS=H
//STEP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
        DEFINE NONVSAM -
                (NAME(dsname) -
                DEVT(device type) -
                VOL(MIGRAT))
/*
```

Note: For VSAM data sets, you must create a NON-VSAM entry for both the base and data components.

3. Run the following command to determine which dump volumes you should use:

```
LIST VOLUME(ml1volser) BCDS ALLDUMPS
```

This will list all of the tape volumes that this ML1 volume was dumped to, and when.

4. Restore the data set. Use the list of volumes created by Step 3 as the source volumes to a DFSMSdss RESTORE job specifying the DFSMSHsm-generated name in DATASET(INCL(dsn)).
5. Run the following command to rebuild the MCD and MCA records for the volume:

```
AUDIT MEDIACONTROLS VOLUMES(ml1volser) FIX
```

6. RECALL the data set.

Case 12: Recovering from a large-scale cyber event

Recovering from a large-scale cyber event may require all of an enterprise's disk storage to be recovered to a point-in-time prior to the cyber event. This recovery may result in the various storage media types (disk, tape, cloud object) representing data at disparate points in time. To prepare your DFSMSHsm environment for this type of scenario:

- **Tape:** Utilize the TS7700 capabilities for Expire Hold and logical WORM. (Or, similar capabilities in non-IBM tape products). These functions ensure that tape data that expired after the recovery point is still available when recovering back to the recovery point.
- **Cloud Object Storage:** Enable DFSMSHsm to maintain migration copies in cloud object storage for a specific number of days beyond the day migration copies would normally be deleted. Refer to the *clouddays* parameter of the SETSYS MIGRATIONCLEANUPDAYS command.

Steps to take to resynchronize DFSMSHsm after a recovering disk environment to a prior point-in-time:

- The disk storing of the DFSMSHsm control data sets, all DFSMSHsm owned disk volumes (ML1, ML2 disk, Backup) and all disk containing DFSMSHsm control data set backup copies must be recovered to the same point-in-time.
- Each DFSMSHsm should be placed in Emergency mode. This will quiesce DFSMSHsm data set processing during the recovery process. Refer to [DFSMSHsm libraries and procedures in z/OS DFSMSHsm Implementation and Customization Guide](#) for instructions on starting DFSMSHsm in emergency mode.
- When utilizing tape, on each DFSMSHsm host, issue the DFSMSHsm **SETSYS PARTIALTAPE (MARKFULL)** command. This ensures that DFSMSHsm will not select any tapes for

append activity from the point to which the DFSMSHsm control data sets were recovered, which will potentially vary from the state of the virtual tapes.

- If the DFSMSHsm control data sets are backed up to disk:
 - Issue the **QUERY CDSVERSIONBACKUP** command.

```
ARC0375I
CDSVERSIONBACKUP, MCDSBACKUPDSN=uid.MCDS.BACKUP, BCDSBACKUPDSN=uid.BCDS.BACKUP,
OCDSBACKUPDSN=uid.OCDS.BACKUP, JRNLBACKUPDSN=uid.JRNL.BACKUP
ARC0376I
BACKUPCOPIES=n, BACKUPDEVICECATEGORY=type, LATESTFINALQUALIFIER=Xnnnnnnn, DATAMOVER=type
```

- Issue catalog command **LISTCAT LEVEL(uid)**, using the *uid* displayed in message ARC0375I.
- If the highest final qualifier from the backup data sets provided by LISTCAT is higher than the value displayed by LATESTFINALQUALIFIER in message ARC0376I, then patch the DFSMSHsm control data set with the highest final qualifier as shown from the data sets listed in the catalog:
 - **FIXCDS S MHCR PATCH(X'B1' nnnnnnn)**
- On the Primary host, issue **BACKVOL CDS** to create a current backup of the control data sets.
- On each host, issue the DFSMSHsm **SETSYS NOEMERGENCY** command.
- For each of the DFSMSHsm control data sets:
 - Perform the EXAMINE function. Refer to [z/OS DFSMS Using Data Sets](#).
 - Perform the DIAGNOSE function. Refer to [z/OS DFSMS Access Method Services Commands](#).
- Run the DFSMSHsm AUDIT commands from a single host. These will take a long time to run. It is recommended that the DFSMSHsm control data sets be accessed using VSAM RLS, which can significantly reduce the elapsed time of these commands. For each command, specify NOFIX and an output data set. From the output of each command, take the appropriate action to correct any reported errors.
 - **AUDIT DATASETCONTROLS(MIGRATION)**
 - Based on functional usage:
 - **AUDIT DATASETCONTROLS(BACKUP)**
 - **AUDIT ABARSCONTROLS**
 - **AUDIT COPYPOOL**
 - **AUDIT FILECONTROLS**
 - If errors are encountered on specific tapes, issue **AUDIT MEDIACONTROLS** for those specific volumes.

Chapter 21. Consolidated commands, storage group definitions, and management classes for the example system

This topic presents the consolidation of all of the DFSMSHsm commands, storage group attributes, and management class attributes that were used to show the automatic storage management of the example system. The examples are separated by the kind of command (ADDVOL, DEFINE, SETSYS, and so forth) and by storage group attributes and management class attributes.

ADDVOL command

The ADDVOL command is used to define DFSMSHsm-owned volumes and non-SMS-managed volumes to DFSMSHsm. It is not used for SMS-managed volumes.

Note: A maximum of eight processing units can ADDVOL a primary or migration volume at the same time.

The consolidated ADDVOL command for the example system is:

```
/* DEFINING ML1 VOLUMES TO DFSMSHsm IN ALL PROCESSING UNITS */
ADDVOL ML1001 UNIT(3390) -
  MIGRATION(MIGRATIONLEVEL1 -
    SMALLDATASETPACKING) THRESHOLD(85)

ADDVOL ML1002 UNIT(3390) -
  MIGRATION(MIGRATIONLEVEL1 -
    SMALLDATASETPACKING) THRESHOLD(90)

ADDVOL ML1003 UNIT(3390) -
  MIGRATION(MIGRATIONLEVEL1 -
    NOSMALLDATASETPACKING) THRESHOLD(90)

/* DEFINING AUTOMATIC DUMP OF ML1 VOLUMES IN PROCESSING */
/* UNIT 2 */
ADDVOL ML1001 UNIT(3390) MIGRATION(AUTODUMP -
  (ONEWEEK,TWOWEEK))

ADDVOL ML1002 UNIT(3390) MIGRATION(AUTODUMP -
  (ONEWEEK,TWOWEEK))

ADDVOL ML1003 UNIT(3390) MIGRATION(AUTODUMP -
  (ONEWEEK,TWOWEEK))

/* DEFINING NOAUTOMATIC DUMP OF ML1 VOLUMES IN PROCESSING */
/* UNITS 1 AND 3 */
ADDVOL ML1001 UNIT(3390) MIGRATION(NOAUTODUMP)

ADDVOL ML1002 UNIT(3390) MIGRATION(NOAUTODUMP)

/* DEFINING PRIMARY VOLUMES FOR MIGRATION AND BACKUP */
/* IN PROCESSING UNIT 1 */
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0002 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0003 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
  AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1001 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1002 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1003 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
  NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2001 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
```



```

NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2002 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2003 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(96 80)

ADDVOL VP0001 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0002 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0003 UNIT(3390) PRIMARY(AUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

/* DEFINING PRIMARY VOLUMES FOR MIGRATION AND BACKUP */
/* IN PROCESSING UNIT 2 */
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0002 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0003 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1001 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1002 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1003 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2001 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2002 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2003 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0001 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0002 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0003 UNIT(3390) PRIMARY(AUTOMIGRATION AUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

/* DEFINING PRIMARY VOLUMES FOR MIGRATION AND BACKUP */
/* IN PROCESSING UNIT 3 */
ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
AUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -

```



```

NOAUTORECALL MIGRATE(12)) THRESHOLD(96 80)

ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTOMIGRATION NOAUTOBACKUP -
NOAUTORECALL MIGRATE(12)) THRESHOLD(95 80)

/* DEFINING PRIMARY VOLUMES FOR DUMP IN PROCESSING */
/* UNIT 2 */
ADDVOL GP0001 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL GP0002 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL GP0003 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL UG1001 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL UG1002 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL UG1003 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL UG2001 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL UG2002 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL UG2003 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,TWOWEEK))

ADDVOL VP0001 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,FOURWEEK))

ADDVOL VP0002 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,FOURWEEK))

ADDVOL VP0003 UNIT(3390) PRIMARY(AUTODUMP -
(ONEWEEK,FOURWEEK))

/* DEFINING PRIMARY VOLUMES FOR DUMP IN PROCESSING */
/* UNITS 1 AND 3 */
ADDVOL GP0001 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL GP0002 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL GP0003 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG1001 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG1002 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG1003 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG2001 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG2002 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL UG2003 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL VP0001 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL VP0002 UNIT(3390) PRIMARY(NOAUTODUMP)

ADDVOL VP0003 UNIT(3390) PRIMARY(NOAUTODUMP)

```

DEFINE command

The DEFINE command is used to define pools, dump classes, cycles, and migration level 2 key ranges. You need the DEFINE commands for pools in the ARCCMDxx member. The definitions of dump classes, Chapter 21. Consolidated commands, storage group definitions, and management classes for the example system **655**

The consolidated DEFINE command for the example system is:

SETSYS command

The consolidated SETSYS command for the example system is:

656 z/OS: z/OS DFSMSHsm Storage Administration


```

OBJECTNAMES(OBJECT, LINKLIB) -
SOURCENAMES(ASM, PROJECT) OPTIMUMDASDBLOCKING -
CONVERSION(REBLOCKTOANY) EXPORTESDS(CIMODE)

/* DEFINE DFSMShsm REPORTING AND MONITORING */
SETSYS ACTLOGMSGVL(EXCEPTIONONLY) -
  ACTLOGTYPE(DASD) MONITOR -
  (BACKUPCONTROLDATASET(80) JOURNAL(80) -
  MIGRATIONCONTROLDATASET(80) NOSPACE -
  OFFLINECONTROLDATASET(80) NOSTARTUP -
  NOVOLUME) SYSOUT(A 1) -
  NOSMF

/* DEFINE DFSMShsm'S TAPE ENVIRONMENT */
SETSYS INPUTTAPEALLOCATION(NOWAIT) DEFERMOUNT -
  TAPEUTILIZATION(UNITTYPE(3490) PERCENTFULL(90) MOUNTWAITTIME(15) -
  OUTPUTTAPEALLOCATION(NOWAIT) -
  USERUNITTABLE(ACL) SELECTVOLUME(SCRATCH) -
  RECYCLETAPEALLOCATION(NOWAIT) -
  TAPESECURITY(RACF) -
  TAPEDELETION(SCRATCHTAPE) -
  TAPEHARDWARECOMPACT -
  PARTIALTAPE(MARKFUL)

/* SPECIFY THE DFSMShsm INSTALLATION-WIDE EXITS THAT ARE USED */
SETSYS EXITON(CD)

/* SPECIFY DFSMShsm'S RECOVERABILITY */
SETSYS JOURNAL(RECOVERY)

/* SPECIFYING WHEN TO START PRIMARY SPACE MANAGEMENT */
/* PROCESSING UNIT 1 */
SETSYS PRIMARYSPMGMTSTART(0200 0400)

/* SPECIFYING WHEN TO START SECONDARY SPACE MANAGEMENT */
/* PROCESSING UNIT 2 */
SETSYS SECONDARYSPMGMTSTART(0030 0200)

/* SPECIFYING INTERVAL MIGRATION */
/* PROCESSING UNIT 2 */
SETSYS INTERVALMIGRATION

/* SPECIFYING NOINTERVAL MIGRATION */
/* PROCESSING UNITS 1 AND 3 */
SETSYS NOINTERVALMIGRATION

/* SPECIFYING MAXIMUM AUTOMATIC MIGRATION TASKS */
/* PROCESSING UNITS 1 AND 2 */
SETSYS MAXMIGRATIONTASKS(5)

/* SPECIFYING LEVEL 2 MIGRATION */
SETSYS TAPEMIGRATION(ML2TAPE(TAPE(ACL)))

/* SPECIFYING EXTENT REDUCTION */
SETSYS MAXEXTENTS(6)

/* SPECIFYING SCRATCHING EXPIRED DATA SETS */
SETSYS EXPIREDDATASETS(SCRATCH)

/* SPECIFYING RECALL TASKS */
SETSYS MAXRECALLTASKS(15) -
  TAPEMAXRECALLTASKS(6)

/* SPECIFYING SMALL DATA SET PACKING */
SETSYS SMALLDATASETPACKING(KB(110))

/* SPECIFYING MIGRATED DATA SET NAME PREFIX */
SETSYS MIGRATEPREFIX(HSM)

/* SPECIFYING MIGRATION CLEANUP */
/* PROCESSING UNIT 1 ONLY */
SETSYS MIGRATIONCLEANUPDAYS(25 10)

/* SPECIFYING AGE FOR SCRATCHING LIST */
/* DATA SETS */
SETSYS SCRATCHFREQUENCY(5)

/* SPECIFYING A MINIMUM AGE FOR SPACE MANAGEMENT */
/* PROCESSING (NON-SMS-MANAGED DATA SETS) */
SETSYS DAYS(0)

/* SPECIFYING INACTIVE AGE FOR ELIGIBILITY FOR */
/* MIGRATION FROM MIGRATION LEVEL 1 */
SETSYS MIGRATIONLEVEL1DAYS(60)

```



```
/* SPECIFYING KIND OF VOLUMES TO USE FOR RECALL */
SETSYS RECALL(ANYSTORAGEVOLUME(LIKE))

/* DEFINING WHEN TO START AUTOMATIC DUMP */
SETSYS AUTODUMPSTART(0300 0400 0500)

/* SPECIFYING THE BUFFERING TECHNIQUE */
SETSYS DUMPIO(3,1)

/* SPECIFYING THE MAXIMUM NUMBER OF DUMP TASKS */
SETSYS MAXDUMPTASKS(3)

/* SPECIFYING WHEN AUTOMATIC BACKUP IS TO START (PROCESSING UNIT 1)*/
SETSYS AUTOBACKUPSTART(0300 0400 0500)

/* SPECIFYING WHEN AUTOMATIC BACKUP IS TO START */
/* (PROCESSING UNITS 2 AND 3) */
SETSYS AUTOBACKUPSTART(0330 0400 0500)

/* SPECIFYING TO DIRECT BACKUP TO TAPE, AND TO RETRY BACKUP */
/* WHEN A DATA SET IS IN USE. */
SETSYS BACKUP(TAPE(ACL) INUSE(RETRY(Y) DELAY(10) +
SERIALIZATION(PREFERRED)))

/* SPECIFYING FIRST QUALIFIER FOR NAMES OF BACKED UP DATA SETS */
SETSYS BACKUPPREFIX(HSM)

/* SPECIFYING MAXIMUM NUMBER OF BACKUP TASKS */
SETSYS MAXBACKUPTASKS(3)

/* SPECIFYING BACKUP OF RACF DISCRETE DATA SET PROFILES */
SETSYS NOPROFILEBACKUP

/* SPECIFYING SPILL PROCESSING */
SETSYS NOSPILL

/* SPECIFYING BACKUP OF ONLY CHANGED DATA SETS */
SETSYS INCREMENTALBACKUP(CHANGEDONLY)

/* SPECIFYING MINIMUM DAYS BETWEEN BACKUPS */
/* NON-SMS-MANAGED DATA SETS */
SETSYS FREQUENCY(1)

/* SPECIFYING MAXIMUM NUMBER OF BACKUP VERSIONS */
/* TO KEEP (NON-SMS-MANAGED DATA SETS */
SETSYS VERSIONS(5)

/* SPECIFYING BACKUP OF CONTROL DATA SETS */
SETSYS CDSVERSIONBACKUP(BACKUPCOPIES(4) -
BACKUPDEVICECATEGORY(TAPE) BCDSBACKUPDSN(BHSM.BCDS.BACKUP) -
MCDSBACKUPDSN(BHSM.MCDS.BACKUP) -
OCDSBACKUPDSN(BHSM.OCDS.BACKUP) -
JRNLBKUPDSN(BHSM.JRNL.BACKUP))

/* SPECIFYING RECYCLE PERCENTAGES AND OUTPUT DEVICES */
SETSYS ML2RECYCLEPERCENT(25) RECYCLEPERCENT(25) -
RECYCLEOUTPUT(BACKUP(ACL),MIGRATION(ACL)) -
MAXRECYCLETASKS(3) ML2PARTIALSNOTASSOCIATEDGOAL(8)
```

Storage groups

Figure 90 on page 659 shows the storage group attributes used to control DFSMSHsm management of storage at a volume level.


```

NAME ==> SG1
MIGRATE SYSTEM/SYSPLEX NAME ==>
AUTO MIGRATE ==> Y
ALLOCATION/MIGRATION THRESHOLD: HIGH ==> 95 LOW ==> 80

AUTO DUMP ==> Y
DUMP CLASS ==> ONEWEEK
DUMP CLASS ==> TWOWEEK
DUMP SYSTEM/SYSPLEX NAME ==>

AUTO BACKUP ==> Y
BACKUP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG2
MIGRATE SYSTEM/SYSPLEX NAME ==>
AUTO MIGRATE ==> Y
ALLOCATION/MIGRATION THRESHOLD: HIGH ==> 95 LOW ==> 80

AUTO DUMP ==> Y
DUMP CLASS ==> ONEWEEK
DUMP CLASS ==> TWOWEEK
DUMP SYSTEM/SYSPLEX NAME ==>

AUTO BACKUP ==> Y
BACKUP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG3
MIGRATE SYSTEM/SYSPLEX NAME ==>
AUTO MIGRATE ==> Y
ALLOCATION/MIGRATION THRESHOLD: HIGH ==> 95 LOW ==> 80

AUTO DUMP ==> Y
DUMP CLASS ==> ONEWEEK
DUMP CLASS ==> FOURWEEK
DUMP SYSTEM/SYSPLEX NAME ==>

AUTO BACKUP ==> Y
BACKUP SYSTEM/SYSPLEX NAME ==>

NAME ==> SG4
MIGRATE SYSTEM/SYSPLEX NAME ==>
AUTO MIGRATE ==> Y
ALLOCATION/MIGRATION THRESHOLD: HIGH ==> 95 LOW ==> 80

AUTO DUMP ==> Y
DUMP CLASS ==> ONEWEEK
DUMP CLASS ==> FOURWEEK
DUMP SYSTEM/SYSPLEX NAME ==>

AUTO BACKUP ==> Y
BACKUP SYSTEM/SYSPLEX NAME==>

```

Figure 90. Consolidated Storage Group Attributes

Management classes

This example shows the management class attributes used to control DFSMSHsm management of storage on a data set basis.

```

NAME ==> STANDMC
EXPIRE AFTER DAYS NON-USAGE ==> 120
EXPIRE AFTER DATE/DAYS ==> 360
PARTIAL RELEASE ==> Y
PRIMARY DAYS NON-USAGE ==> 10
LEVEL 1 DAYS NON-USAGE ==> 60
COMMAND OR AUTO MIGRATE ==> BOTH

BACKUP FREQUENCY ==> 3
NUMBER OF BACKUP VERSIONS ==> 3
  (DATA SET EXISTS)
NUMBER OF BACKUP VERSIONS ==> 1
  (DATA SET DELETED)
RETAIN DAYS ONLY BACKUP VERSION ==> 60
  (DATA SET DELETED)
RETAIN DAYS EXTRA BACKUP VERSIONS ==> 20
ADMIN OR USER COMMAND BACKUP ==> BOTH
AUTO BACKUP ==> Y

```



```

NAME ==> GDSMC
EXPIRE AFTER DAYS NON-USAGE ==> 7
EXPIRE AFTER DATE/DAYS ==> 90
PARTIAL RELEASE ==> N
PRIMARY DAYS NON-USAGE ==> 5
LEVEL 1 DAYS NON-USAGE ==> NOLIMIT
COMMAND OR AUTO MIGRATE ==> BOTH
# GDG ELEMENTS ON PRIMARY ==> 2
ROLLED-OFF GDS ACTION ==> EXPIRE

BACKUP FREQUENCY ==> 1
NUMBER OF BACKUP VERSIONS ==> 1
  (DATA SET EXISTS)
NUMBER OF BACKUP VERSIONS ==> 1
  (DATA SET DELETED)
RETAIN DAYS ONLY BACKUP VERSION ==> 14
  (DATA SET DELETED)
RETAIN DAYS EXTRA BACKUP VERSIONS ==>
ADMIN OR USER COMMAND BACKUP ==> ADMIN
AUTO BACKUP ==> Y

NAME ==> LARGEMC
EXPIRE AFTER DAYS NON-USAGE ==> 90
EXPIRE AFTER DATE/DAYS ==> 180
PARTIAL RELEASE ==> Y
PRIMARY DAYS NON-USAGE ==> 15
LEVEL 1 DAYS NON-USAGE ==> 0
COMMAND OR AUTO MIGRATE ==> BOTH

BACKUP FREQUENCY ==> 2
NUMBER OF BACKUP VERSIONS ==> 4
  (DATA SET EXISTS)
NUMBER OF BACKUP VERSIONS ==> 1
  (DATA SET DELETED)
RETAIN DAYS ONLY BACKUP VERSION ==> 60
  (DATA SET DELETED)
RETAIN DAYS EXTRA BACKUP VERSIONS ==> 20
ADMIN OR USER COMMAND BACKUP ==> ADMIN
AUTO BACKUP ==> Y

NAME ==> NOMIGMC
EXPIRE AFTER DAYS NON-USAGE ==> 20
EXPIRE AFTER DATE/DAYS ==> NOLIMIT
PARTIAL RELEASE ==> Y
PRIMARY DAYS NON-USAGE ==> NOLIMIT
LEVEL 1 DAYS NON-USAGE ==> NOLIMIT
COMMAND OR AUTO MIGRATE ==> NONE

BACKUP FREQUENCY ==> 1
NUMBER OF BACKUP VERSIONS ==> 1
  (DATA SET EXISTS)
NUMBER OF BACKUP VERSIONS ==> 1
  (DATA SET DELETED)
RETAIN DAYS ONLY BACKUP VERSION ==> 30
  (DATA SET DELETED)
RETAIN DAYS EXTRA BACKUP VERSIONS ==>
ADMIN OR USER COMMAND BACKUP ==> BOTH
AUTO BACKUP ==> N

```


Chapter 22. Preventive methods for avoiding data loss

This topic presents methods for avoiding data loss.

Steps for preventing a full journal (damaged CDS)

To prevent a full journal, perform the following steps:

1. Ensure that no reorganization occurs if DFSMSHsm is running on any image in the complex. You must do two things to accomplish this.
 - Make sure that SYSDSN enqueues are in the system's inclusion list for global resource serialization (GRS). This allows the SYSDSN enqueues to be passed to all images in the complex that share the CDSs.
 - Modify the JCL used for reorganization of the CDSs by using a disposition of OLD whenever a DD statement refers to a CDS.
2. Define the DFSMSHsm CDSs with VSAM SHAREOPTIONS(2 3). You can find information about defining VSAM SHAREOPTIONS for DFSMSHsm in the topic about VSAM SHAREOPTIONS for control data sets and associated CDS considerations in [z/OS DFSMSHsm Implementation and Customization Guide](#).
3. Implement a procedure where the backup of the CDSs is closely monitored.

One method of monitoring the backups of the CDSs is to review the CDS backup and export messages in the DFSMSHsm backup activity log. Specifically, review the number of records processed by the export function in message IDC0005I, for each CDS. Compare the record count of the current CDS backup with the record count from the preceding days. The record count may not be the same, but it should be close.

Normal update and deletion activity causes some fluctuation in the number of records in the CDSs. However, when a large decrease in records is detected, you should be able to account for the large difference.

Some large fluctuations in the number of CDS records are normal, and result from processes that reduce the number of records in a CDS. For example, running the EXPIREBV command deletes records in the BCDS. DFSMSHsm secondary space management deletes records in the MCDS. However, some large fluctuations in the number of records in a CDS can indicate that the CDS has been damaged. For example, in a multiple-MVS-image environment, the reorganization of a CDS occurring on one image while DFSMSHsm is running on another image can damage the structure of the CDS, and cause a large fluctuation in CDS records.

To eliminate any doubt about the soundness of the CDS structure, run the IDCAMS EXAMINE command against the CDS in question. Running the EXAMINE command against your CDSs every day minimizes the chance of CDS damage going undetected.

Preventing a full journal (undamaged CDS)

To help prevent the problem of a full journal, ensure that your journal is of adequate size to record all the activity between backups of the CDSs. In conjunction with that, schedule CDS backups frequently enough so that the journal can handle the activity. Most installations run automatic backup once a day, seven days a week; the journal data set therefore needs to be large enough to contain data for a 24-hour time period.

Consider using a large format data set for the DFSMSHsm journal. A large format data set is a physical sequential data set with the ability to grow beyond 65,535 tracks per volume. Using a larger journal data set can allow more DFSMSHsm activity to take place between journal backups and helps to avoid journal full conditions.

Also, use the SETSYS MONITOR (JOURNAL) command to send your warnings when the journal data set is getting full. When the journal data set is filled to the specified threshold, which you can set to any percentage from 0 to 100 or leave at the DFSMSHsm default value of 80%, a warning message is issued to the system operator. Messages continue to be issued until the journal data set is 100% full. Set your threshold at a value that gives you adequate time to back up your CDSs and journal and to allocate a new journal.

Customers that have NetView or another automation product can monitor the ARC0909E and ARC0026E messages, keying in on the 100% and EOVS message variables. When 100% and EOVS are issued, NetView can automatically issue the BACKVOL CDS command.

Preventing structural damage to CDS and missing journal records

Perform the following steps to prevent structural damage to CDS and missing journal records:

1. Schedule reorganizations of CDSs to occur only when necessary. If the CDS is of adequate size, you should not have to reorganize often.
2. Use the proper procedures to reorganize your CDSs.
3. Ensure that no reorganization can occur when DFSMSHsm is running on any MVS image in the complex. To do this:
 - Make sure that SYSDSN enqueues are in the system's inclusion list for global resource serialization (GRS); this allows the SYSDSN enqueues to be passed to all images in the complex that share the CDSs.
 - Modify the JCL used for reorganization of the CDSs by using a disposition of OLD whenever a DD statement refers to a CDS.
4. Define the DFSMSHsm CDSs with VSAM SHAREOPTIONS(2 3).
5. Use the IDCAMS EXAMINE command to do preventive examines.
6. Reassess the number of backup versions you keep for the CDSs and journal. The more backup versions you have, the greater your chances are of having a good CDS and journal backup version to use should the need arise.

Related reading

- For more information about the procedures used to reorganize your control data sets, see the topic about reorganizing the control data sets in [z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about defining VSAM SHAREOPTIONS for DFSMSHsm, see the topic about VSAM SHAREOPTIONS for control data sets and the topic about control data set considerations in [z/OS DFSMSHsm Implementation and Customization Guide](#).

Preventing time sequence gaps in CDS records (journal records also missing)

To prevent time sequence gaps in CDS records, perform the following steps:

1. Call on IDCAMS REPRO to create an output data set, check for a successful completion code on the creation of the output data set, and create an input data set.

If a successful completion code is not received for the creation of the output data set, the job ends, and the input data set is not created.
2. Make CDS data sets large enough that the need for a reorganization is infrequent.

If your CDS is currently too small and you want to change that, the next time you reorganize the data set, add 10 cylinders. If the data set is still too small, the next time you reorganize add another 10 cylinders. Do this until the data set is large enough that the need for reorganizations is infrequent.
3. Do a backup of the CDS before every reorganization, and keep those backup tapes until it is time for the next reorganization.

4. Reorganize your CDSs by utilizing the proper procedures.
5. Protect DFSMSHsm's resources with RACF or a comparable product.

Related reading

For more information about reorganizing your control data sets, see the topic about reorganizing the control data sets in [*z/OS DFSMSHsm Implementation and Customization Guide*](#).

Preventing overwritten migration tape or overwritten backup tape

Perform the following steps to avoid overwriting migration tape or backup tape:

1. Protect DFSMSHsm's resources with RACF or a comparable product.
2. Protect your alternate volumes if your tape management system does not do it for you. DFSMSrmm protects your alternate tape volumes. Some products, however, do not.

Prepare for recovery of a head crash on an ML1 volume

To prepare for recovery of a head crash on an ML1 volume, perform the following steps:

1. Use the IDCAMS EXPORT TEMPORARY command to regularly export SDSPs with high activity.

Note: Be sure to use the TEMPORARY parameter with the EXPORT command so the export function does not delete your original data set.

2. Dump ML1 volumes regularly.

Reestablishing access to previously deleted, migrated data sets (no backup exists, ML2 only)

Perform the following steps to reestablish access to previously deleted, migrated data sets:

1. Keep multiple CDS backup versions so you can recover from situations such as this.
2. Create backup copies of data sets that are important. Without a backup copy, the migration copy of a data set is the only copy you have.
3. Use the DCOLLECT function to track what data exists in active, migrated, or backed up format at the time the DCOLLECT function is run. If you run DCOLLECT daily or a couple of times a week, with output going to a generation data group (GDG), the DCOLLECT information could be a valuable asset in data recovery.

Chapter 23. Linux-z/OS DFSMSHsm dump or restore how-to

This topic describes how to back up your Linux partitions and volumes. z/OS is the IBM operating system. You can include your Linux volumes in an existing z/OS backup solution that uses DFSMSHsm. This backup solution works in two environments: (1) a z/OS focused environment with z/OS running on several LPARs and a few dozen Linux servers running under z/VM® in an LPAR; (2) a Linux-focused environment with z/VM running in BASIC mode, with hundreds to thousands of Linux guests, and one or more z/OS guests that will perform the DFSMSHsm processing.

LINUX is a registered trademark of Linus Torvalds.

To get the latest version of this HOW TO and related utilities, check with your Linux provider.

How DFSMSHsm supports linux

This topic provides a discussion of only the supported DFSMSHsm functions for Linux volumes.

For Linux volumes, DFSMSHsm supports only the dump and recover functions. DFSMSHsm dump is a physical copy of DASD tracks to tape. An automatic dump process initiates dump processing on a repeating schedule. After reading this HOW TO, you will know what to do to have your Linux volumes dumped by DFSMSHsm.

Note: Linux volumes should not be managed by SMS.

Using DFSMSHsm with Linux requirements

This topic presents a discussion of several requirements related to using Linux.

Hardware environment

Linux can run on a mainframe under the following three modes:

BASIC

A single operating system that owns the entire mainframe.

LPAR

A mainframe divided into logical partitions, with each partition running its own operating system image. There can be up to 15 LPARs.

VIRTUAL

z/VM supports hundreds to thousands of guests. Each guest runs its own operating system—for example, one guest using Linux and another using z/OS, all on the same hardware.

DASD background

The mainframe environment can store data on Direct Access Storage Device (DASD) by utilizing an extended count-key-data (ECKD) protocol. DASD is divided into logical units known as volumes. Volumes divide into fixed-size tracks, and the track's geometry determines its size. Linux supports both the 3380 and 3390 track geometries. You can reference a volume by using either a 16-bit device number or a six-character volume serial number (referred to as the *volser*). More than one operating system can access a volume at the same time. A volume can have a different device number for each operating system that has access to it, but only one *volser*. Linux allows multiple volumes to have the same volume serial number, while z/OS allows only one volume per instance of z/OS to have the same volume serial number.

Linux requirements

In order for z/OS to process the Linux volumes, they must be formatted in the compatible disk layout (CDL) using **dasdfmt** version 1.0 or later and partitioned using **fdasd** version 1.0 or later.

Linux disk utilities

dasdfmt

The default disk layout for **dasdfmt** is CDL. Volumes in the original format, Linux disk layout (LDL), are not compatible with the z/OS system and cannot be accessed and backed up by z/OS. Below is an example of how to format a disk with **dasdfmt** at address 0198, having a byte block size of 4096, and a *volser* of LNX200:

```
dasdfmt -n 198 -b 4096 -l lnx200
```

A screen appears like the one below:

```
Drive Geometry: 3339 Cylinders * 15 Heads = 50085 Tracks
I am going to format the device 198 in the following way:
  Device number of device : 0x198
  Major number of device  : 94
  Minor number of device   : 8
  Labelling device         : yes
  Disk label               : VOL1
  Disk identifier          : LNX200
  Extent start (trk no)    : 0
  Extent end (trk no)      : 50084
  Compatible Disk Layout   : yes
  Blocksize               : 4096

----> ATTENTION! <---- All data in the specified range of that device
will be lost. Type "yes" to continue, no will leave the disk untouched:
```

You can see a 3339 cylinder volume that is attached to Linux at address 198 in the screen capture above. The volume serial number (*volser*) is LNX200, and its block size is 4096 bytes—the optimal and recommended block size. The *volser* must be six characters in length. If the output reads: Disk label: VOL1, then a z/OS system can process this volume.

If the output reads: Disk label: LNX1, then it is an LDL volume, and z/OS cannot process volumes that do not have VOL1.

fdasd

After you have formatted a volume with **dasdfmt**, you must partition it before Linux can use it. Use the **fdasd** program to partition the volume. This program is similar to the **fdisk** program that comes with the Linux version that runs on personal computers. One difference is that it creates partitions on ECKD DASD instead of hard drives. With **fdasd**, you can create up to three partitions on a volume. Each partition appears to z/OS as a data set. **fdasd** asks you to choose the beginning track and ending track or size of the partition. The naming conventions for the data sets and partitions are listed in [“Data set and partition naming conventions”](#) on page 675.

Follow these rules when you create partitions with **fdasd** that you want to back up with z/OS tools.

- Create partitions by starting from the lowest possible track.
- Do not leave gaps between partitions.
- If you want to restore a partition, **do not** delete it with **fdasd** first. When you delete the partition with **fdasd**, the naming convention between the partition names and data set names is no longer valid. If you then recover a partition on that volume, you could recover over the wrong partition, losing data. When you restore the partition, you replace the existing partition.
- If you follow the previous rules of thumb above, then you can add partitions to a volume at any time up to a maximum of three partitions or until the entire volume is used. If you need to reorganize the

partitions, you should back up the data on the partitions and then delete all partitions and create new ones in the desired configuration.

z/OS requirements

The tools that IBM supports for backing up Linux volumes are DFSMSdss (Data Facilities Storage Management Subsystem data set services) and DFSMSHsm. DFSMSdss and DFSMSHsm are elements of z/OS for backing up data and managing disk space. The z/OS image must also be able to attach and vary online the volumes that Linux is using that you want backed up.

Authorization requirements

You need to have root authority for Linux in order to mount and unmount the partitions, as well as to format and partition the volumes using **dasdfmt** and **fdasd**. For z/OS, you need authority to submit DFSMSHsm commands and, optionally, to run ADRDSSU, which is the program that is invoked when using DFSMSdss. z/OS sees the Linux partitions as data sets. You can prevent unauthorized access of the Linux partitions by z/OS applications and users by using the IBM RACF, a component of the Security Server for z/OS.

Setting up your dump environment for Linux volumes

A Linux volume is formatted with **dasdfmt** in the CDL format and partitions put on it with **fdasd**. It has a valid VTOC on it and a data set for each Linux partition. A Linux volume holds only Linux partitions. No MVS or VM data sets or any non-Linux data should be allowed on a Linux volume. A Linux volume can hold one to three partitions and can be a 3380 or 3390.

The data sets created by **fdasd** are not added to or removed from the catalog by **fdasd** as they are allocated and deleted. If you manually catalog the data sets, you must uncatalog them if you delete them with **fdasd**. It is recommended that you leave the data sets uncataloged. The partition and data set naming conventions are listed in [“Data set and partition naming conventions”](#) on page 675.

Performing only dumps

DFSMSHsm can only dump and recover from dump those Linux volumes that are formatted with **dasdfmt** and partitioned with **fdasd**. All other DFSMSHsm functions are unsupported for Linux volumes. No data sets should be migrated off or recalled to Linux volumes. Backup is not supported for Linux volumes or partitions. Use Linux volumes only as PRIMARY volumes and use them with the NOAUTOBACKUP, NOAUTOMIGRATION, and NOAUTORECALL options. For the dump class that you use with Linux, use the option VTOCCOPIES(0). VTOCCOPIES is not supported for Linux volumes.

When do you want your dumps performed?

If you already use automatic dump, then a time of day and dump cycle is already in place. If not, then typically the time and day of the week is when the system is not very busy. The daily dump cycle consists of a repeating pattern of up to 31 days where dumps are made or not made. Most DFSMSHsm administrators use a weekly or daily dump cycle. You can choose any days where dumps are made to have your volumes dumped; however, some dump days might be busier than others, limiting tape hardware availability on those days. You can also specify how frequently you perform dumps by the minimum number of days that must pass from one dump to the next. If you do not already have automatic dump set up, then tape hardware availability and DASD write activity are the only factors limiting when you can dump your Linux volumes. You can also perform a manual dump of your volumes for special situations.

When scheduling your dumps, you must ensure that no writes to the volumes that you are dumping occur while the dumps are being made. The best way to ensure the data integrity of your partitions and their dumps is to unmount them or to have them mounted as read-only. By unmounting a partition, you flush all buffered data that has not yet been written to the partition.

How long do you want to keep your dumps?

You can specify how many days you want to retain your dumps. You may want to specify how many days you retain your dumps in multiples of how frequently in days that you dump. Then you will always have the same number of dump generations at your disposal.

How many dumps are stacked on each dump tape?

The capacity of current tape technology allows several DASD volumes to be dumped to one dump tape. Only one dump of a specific volume can exist on a dump tape. Once a dump tape has dumps on it, no new dumps can be added to it on any subsequent dump day. Also, a dump tape holds only dumps from one dump class.

Consideration: If you have data that belongs to customers, you may not want to have data from more than one customer on the same tape. If your customer has a right to request all copies of data, and if the data co-exists on a dump tape with another customer's data, you will have to copy the data to another tape. Assigning different dump classes to different customers ensures that their data is on separate tapes.

How many tape drives do you have?

Your mainframe environment probably has many tape drives, but you probably will not be able to use all of them. To limit the number of tape drives that DFSMSHsm uses for dump, use the SETSYS command with the MAXDUMPTASKS option. Once you set the maximum number of dump tasks, you prevent DFSMSHsm from taking up too many tape drives and delaying other functions that also require tape drives.

Recovering your data

When recovering data, unmount any partitions that are being recovered during the recovery process to ensure data integrity. There are several ways you can recover your data from a DFSMSHsm dump. You can recover an entire volume with DFSMSHsm or DFSMSdss. You can also recover a partition from a dump with DFSMSHsm or DFSMSdss. To use DFSMSdss to process Linux volumes, see [z/OS DFSMSdss Storage Administration](#). When a volume or partition is recovered, the dump overwrites it; the partition or volume then returns to the state at which the dump was made. All changes since the dump was made are lost. See [“Data set and partition naming conventions” on page 675](#) on the data set and partition naming conventions to make sure that the right partition is recovered.

There are some limitations in how you can recover to a Linux volume. For a volume of more than 65536 tracks, you can recover only the full volume from a dump; no individual data sets can be recovered to it with DFSMSHsm. However, you can recover a data set to a volume over 65536 tracks from a DFSMSHsm dump with DFSMSdss. To use DFSMSdss on Linux volumes and partitions, see [z/OS DFSMSdss Storage Administration](#). Either a full volume or an individual data set can be recovered to a volume of less than 65536 tracks by DFSMSHsm. When you recover a data set or volume with DFSMSHsm, you must specify the FROMDUMP(DUMPVOLUME(volser)) option. You can list the dump volumes with the LIST DUMPVOLUME command.

Recovering a volume with DFSMSHsm

To recover a volume with DFSMSHsm, you need to know only the volser of the volume you want recovered and from how far back you want your volume recovered. Remember, you can go only as far back as the number of days you have been keeping your dumps. You can go back by date or by the generation of dump where generation 0 is the most recent dump, generation 1 is the second most recent dump, and so on.

The following command recovers volume LNX400 from the third most recent dump.

```
RECOVER * TOVOLUME(LNX400) UNIT(3390) - FROMDUMP(DUMPGENERATION(2))
```

You can also recover a dump to a volume other than the one that was used to make the dump with DFSMSHsm. For instance, you can dump volume LNX400 and recover the dump to volume SPARE5; but in

the process, volume SPARE5 is renamed with volume LNX400's volser, which is not a problem for Linux as it is for z/OS. The z/OS operating system cannot have two or more volumes with the same volser online at the same time. The original volume must be varied offline before the recovery is started. You need to know the volsers of both the volume you want recovered and the volume you want the dump recovered to.

The following command recovers the most recent dump of volume LNX400 to volume SPARE5 and changes the volser of volume SPARE5 to LNX400. Volume LNX400 must be varied offline in MVS before this recovery command is issued.

```
RECOVER * TOVOLUME(LNX400) UNIT(3390) TARGETVOLUME(SPARE5) - FROMDUMP
```

If you do not have an extra volume attached to your Linux environment, you need to shut it down and reconfigure your environment with the extra volume and reboot your Linux environment. You do not have to do this in MVS.

Recovering a volume with DFSMSdss

To recover a volume from a DFSMShsm dump with DFSMSdss, use the LIST command to find the volser of the dump tape volume that contains the dump you want to recover. See [“The DFSMShsm LIST command” on page 675](#) on how to use the LIST command. You can search the output data set for the volser of the volume you want to recover. Once you find the dump that you want to recover in the output data set, you will find the volser of the dump volume and the data set name of your dump. If you recover the dump to an alternate volume, DFSMSdss does not change the volser to that of the volume that the dump was created from. You need to submit JCL to use DFSMSdss. Once you have the information that you need from the output data set, you can submit a job like the following to recover your volume.

The following JCL recovers the dump data set DFHSM.DMP.DAILY.VLNX300.D01244.T072510 to volume LNX300 from dump tape A02001.

```
//LNXJOB JOB , 'IBMUSER',MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=A,CLASS=A
//STEPDSS EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SOURCE   DD UNIT=3480,VOL=SER=A02001,DISP=OLD,
//          DSN=DFHSM.DMP.DAILY.VLNX300.D01244.T072510
//TARGET   DD UNIT=3390,VOL=SER=LNX300,DISP=OLD
//SYSIN    DD *
//          RESTORE FULL INDDNAME(SOURCE) OUTDDNAME(TARGET) PURGE
//          /*
```

Recovering a partition with DFSMShsm

When you recover a partition, leave the original damaged partition in place. If you do not still have the original partition or you want to recover the partition to another volume, then you first must preallocate a partition of the exact same number of tracks with **fdasd**. You must know the volser of the volume that contains the partition you want to recover, which partition to recover, and how far back you want to recover from. If you had to preallocate a partition, then you also need to know the volser of the preallocated partition and which partition you preallocated. The preallocated partition does not need a file system.

If a Linux user does not follow the rules of thumb in [“fdasd” on page 666](#), then the data set name of one or more of the partitions could be changed. The **fdasd** utility initially creates the data sets in order where PART0001 is always the data set that has the lowest starting track, PART0002 has the next lowest starting track, and PART0003 has the highest starting track. Once a partition is removed with **fdasd**, any other changes made after the first removal may cause the order and naming not to be valid. See the naming conventions in [“Naming conventions” on page 675](#).

You also need to know the volser of the dump tape that contains the dump from which you want to recover. To find the volser of the dump tape, use the LIST command. See [“The DFSMShsm LIST command” on page 675](#) on how to use the LIST command.

The following command recovers data set LINUX.VLNX300.PART0001.NATIVE on volume LNX300 from the dump of LNX300 on dump tape LX0001. If the dump of volume LNX300 was the only dump on the

dump tape, omit the SOURCEVOLUME parameter, but if dumps from two or more volumes are stacked on the dump tape, then you must specify SOURCEVOLUME.

```
RECOVER LINUX.VLN300.PART0001.NATIVE TOVOLUME(LNX300) - UNIT(3390)
FROMDUMP(DUMPVOLUME(LX0001) SOURCEVOLUME(LNX300)) - REPLACE
```

To recover a data set to an alternate location, use the RENAME and TOVOLUME options to rename the data set (using the naming convention listed in [“Data set and partition naming conventions”](#) on page 675) and to specify the volume that you want the data set recovered to.

The following command recovers data set LINUX.VLN300.PART0002.NATIVE from the dump on dump tape LX0002 to volume LNX100 and renames it LINUX.VLN100.PART0001.NATIVE, replacing the preallocated data set on LNX100.

```
RECOVER LINUX.VLN300.PART0002.NATIVE - FROMDUMP(DUMPVOLUME(LX0002)) -
NEWNAME(LINUX.VLN100.PART0001.NATIVE) REPLACE - TOVOLUME(LNX100) UNIT(3390)
```

Recovering a partition with DFSMSdss

To recover a partition from a DFSMSshm dump with DFSMSdss, use the LIST command to get the same information needed to recover a volume with DFSMSdss. See [“The DFSMSshm LIST command”](#) on page 675 on how to use the LIST command. You also need the data set name for the data set that holds the partition you want recovered and the data set name for the partition that you want the dump recovered to.

The following JCL recovers LINUX.VLN300.PART0001.NATIVE from dump DFHSM.DMP.DAILY.VLN300.D01244.T072510 on dump tape A02001 to volume LNX200 and renames it LINUX.VLN200.PART0001.NATIVE, replacing the data set of the same name on LNX200. If you are restoring the data set to its original volume, then you can omit the RENAME portion.

```
//LNXJOB JOB , 'IBMUSER',MSGLEVEL=(1,1),TIME=(5,0),REGION=0M,
//      MSGCLASS=A,CLASS=A
//STEPDSS EXEC PGM=ADDRSSU
//SYSPRINT DD SYSOUT=*
//SOURCE DD UNIT=3480,VOL=SER=A02001,DISP=OLD,
//        DSN=DFHSM.DMP.DAILY.VLN300.D01244.T072510
//TARGET DD UNIT=3390,VOL=SER=LNX200,DISP=OLD
//SYSIN DD *
//      RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) -
//              DATASET(INCLUDE(LINUX.VLN300.PART0001.NATIVE)) -
//              RENAME(LINUX.VLN200.PART0001.NATIVE) REPLACE
//*
```

Example dump environment

This example consists of one 3390-3 volume of about 2.9 gigabytes and one 3390-9 volume of about 8.7 gigabytes and a 65 cylinder 3390 of about 45 megabytes. The volumes will be dumped at 10:00 am to 3480 tapes on two 3480 tape drives. The 3390-3 and 3390-9 will be dumped daily and the 65 cylinder will be dumped weekly. The dump cycle will be a seven day, weekly cycle. The **fdasd** utility will be used to partition each volume as follows.


```

Disk /dev/dasd/0199/disc:
    65 cylinders,
    15 tracks per cylinder,
    12 blocks per track
    4096 bytes per block
volume label: VOL1, volume identifier: LNX100
maximum partition number: 3

-----tracks-----
Device      start    end    length  Id  System
/dev/dasd/0199/part1      2     974     973   1  Linux native

Disk /dev/dasd/0198/disc:
    3339 cylinders,
    15 tracks per cylinder,
    12 blocks per track
    4096 bytes per block
volume label: VOL1, volume identifier: LNX200
maximum partition number: 3

-----tracks-----
Device      start    end    length  Id  System
/dev/dasd/0198/part1      2    3201    3200   1  Linux native
/dev/dasd/0198/part2   3202   50084   46883   2  Linux native

Disk /dev/dasd/019a/disc:
    10017 cylinders,
    15 tracks per cylinder,
    12 blocks per track
    4096 bytes per block
volume label: VOL1, volume identifier: LNX300
maximum partition number: 3

-----tracks-----
Device      start    end    length  Id  System
/dev/dasd/019a/part1      2    3201    3200   1  Linux native
/dev/dasd/019a/part2   3202    4267    1066   2  Linux native
/dev/dasd/019a/part3   4268   150254  145987   3  Linux native

```

The partitions will be mounted as follows.

```

/dev/dasd/0198/part1 on /example type ext2 (rw)
/dev/dasd/0199/part1 on /example/boot type ext2 (rw)
/dev/dasd/0198/part2 on /example/usr type ext2 (rw)
/dev/dasd/019a/part1 on /example/var type ext2 (rw)
/dev/dasd/019a/part2 on /example/tmp type ext2 (rw)
/dev/dasd/019a/part3 on /example/home type ext2 (rw)

```

This example is designed to demonstrate how one might implement a simple DFSMSHsm dump environment for a Linux guest on z/VM. It is not intended to demonstrate how to best partition volumes for Linux.

How to define your dump cycle

Before you start automatic dump, you must define a dump cycle. This environment uses a daily dump cycle. The start date must be today's date or a date that has already passed. You need to use a past date if you are using a dump cycle of more than one day and today is not the first day in the dump cycle.

The command below has the first Y day coincide with the day of 2001/08/26, which is a Sunday. Of the dump classes listed in the next topic, the WEEKLY dump class has a DAY value of 7, so it is dumped on Saturdays. The MONTHLY dump class has a DAY value of 1, so it is dumped on Sundays. Because the DAILY dump class has no DAY value specified, it dumps on all Y days. If an N were in the dump cycle, automatic dump would not dump any volumes on that day.

```
DEFINE DUMPCYCLE(YYYYYYY CYCLESTARTDATE 2001/08/26)
```

How to define your dump classes

LNX100 must be backed up only once a week because it will not change very often. LNX200 and LNX300 change more often and must be backed up daily. Because this environment uses a weekly dump cycle, the

DAY and FREQUENCY options determine which days each dump class is dumped. The daily and weekly dumps are short-term dumps that are not kept for long. Another dump class creates dumps monthly and keeps the dumps for three months. All of the dump classes have a STACK value of three to allow dumps from up to three volumes to be put on one dump tape. One is the default STACK value, meaning that no stacking will occur. The following commands define the dump classes.

```
DEFINE DUMPCLASS(DAILY AUTOREUSE FREQUENCY(0) -  
    RETENTIONPERIOD(3) STACK(3) UNIT(3480) VTCCOPIES(0))  
  
DEFINE DUMPCLASS(WEEKLY AUTOREUSE FREQUENCY(0) DAY(7) -  
    RETENTIONPERIOD(28) STACK(3) UNIT(3480) VTCCOPIES(0))  
  
DEFINE DUMPCLASS(MONTHLY AUTOREUSE FREQUENCY(28) DAY(1) -  
    RETENTIONPERIOD(90) STACK(3) UNIT(3480) VTCCOPIES(0))
```

If, for example, you have only two tape drives at your disposal, use the SETSYS command to set the maximum number of dumps that may be taken at any one time.

```
SETSYS MAXDUMPTASKS(2)
```

How to ADDVOL your volumes

Because the volumes will only be dumped, automatic migration and backup features are not allowed. When the dumps expire, the dump tapes are automatically reused for later dumps.

```
ADDVOL LNX100 UNIT(3390) PRIMARY(AUTODUMP(WEEKLY,MONTHLY) -  
    NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)  
  
ADDVOL LNX200 UNIT(3390) PRIMARY(AUTODUMP(DAILY,MONTHLY) -  
    NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)  
  
ADDVOL LNX300 UNIT(3390) PRIMARY(AUTODUMP(DAILY,MONTHLY) -  
    NOAUTOBACKUP NOAUTOMIGRATION NOAUTORECALL)
```

How to start automatic dumping

To automatically dump the volumes, issue the SETSYS AUTODUMPSTART command with the start and finish times.

```
SETSYS AUTODUMPSTART(1100 1200 0000)
```

The times are in a 24-hour format. The first time is when automatic can start, the second time is the latest that an automatic dump can start, and the third time is the latest that a dump of a volume can start. Because the third time is 0000, all volumes are dumped if automatic dump starts. To turn off automatic dump, resubmit the command with all three times equal to 0000.

To manually dump the volumes, issue the BACKVOL command.

```
BACKVOL VOLUMES(LNX100,LNX300,LNX400) -  
    DUMP(DUMPCLASS(WEEKLY,MONTHLY))
```

If you omit DUMPCLASS(WEEKLY,MONTHLY), the volumes are dumped to the dump classes listed for each volume with the AUTODUMP option when it was added with ADDVOL, but no stacking occurs. You can list up to five dump classes. Dumps made manually are taken into account when recovering from a dump by generation or date. If you are only going to be dumping manually, you can omit the AUTODUMP option when you ADDVOL your volumes. You can use any valid dump class, regardless of the AUTODUMP option.

If you decide to create your dumps manually, you still need to run automatic dump to have old dumps expire. To have automatic dump run without actually taking dumps, you can define a dump cycle of one N day.

Recovering from a dump

Before recovering any data, you must leave any partitions that you want to recover in place or reallocate them with **fdasd**; otherwise, Linux cannot see the partitions that have been recovered. Listed here are only a few basic examples of recovering data. For a full description on how to recover data, see [Chapter 60, “RECOVER command: Recovering a backup version or dump copy of a data set, or UNIX file, or a volume,”](#) on page 997.

Recovering with DFSMSHsm

To recover a volume with DFSMSHsm, issue the RECOVER command.

The following RECOVER command recovers LNX200 from the most recent dump.

```
RECOVER * TOVOLUME(LNX200) UNIT(3390) FROMDUMP
```

The following RECOVER command recovers LNX200 from the most recent dump of the MONTHLY dump class.

```
RECOVER * TOVOLUME(LNX200) UNIT(3390) -  
FROMDUMP(DUMPCLASS(MONTHLY))
```

The following RECOVER command recovers LNX200 from the third most recent dump. Generation 0 is the most recent dump, 1 is the second most recent, and so on.

```
RECOVER * TOVOLUME(LNX200) UNIT(3390) -  
FROMDUMP(DUMPGENERATION(2))
```

To recover a partition from a DFSMSHsm dump with DFSMSHsm, you again use the RECOVER command and the LIST command. See [“The DFSMSHsm LIST command”](#) on page 675 on how to use the LIST command. You must use the LIST command to find the dump tape that has the dump from which you want to recover your partition. You also need to know the data set and partition naming conventions so that you recover the correct data set for your partition. The naming conventions are listed in [“Naming conventions”](#) on page 675.

The following RECOVER command recovers data set LINUX.VLNX200.PART0002.NATIVE that holds partition part2 on LNX200. You must specify the SOURCEVOLUME parameter if the dump class has a STACK value greater than one because the dump tape may hold more than one dump data set.

```
RECOVER LINUX.VLNX200.PART0002.NATIVE -  
FROMDUMP(DUMPVOLUME(LX0001) SOURCEVOLUME(LNX200)) REPLACE
```

Recovering data from DFSMSHsm dump with DFSMSdss

Before you recover a volume from a DFSMSHsm dump with DFSMSdss, you must use the LIST command to see what dumps you have. See [“The DFSMSHsm LIST command”](#) on page 675 on how to use the LIST command. Once you find the dump you want to use, you can write and submit JCL to have DFSMSdss recover the volume.

The following JCL recovers dump data set DFHSM.DMP.MONTHLY.VLNX300.D01245.T234610 to volume LNX300. If there is more than one dump stacked on the tape, you must specify the file sequence number of the label keyword. You can get the file sequence number from the information provided by the LIST command in the next topic.


```
//LNXJOB JOB ,USER=IBMUSER,PASSWORD=IBM,MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=A,CLASS=A
//STEPDSS EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SOURCE   DD UNIT=3480,VOL=SER=(A02005,A02008,A02010),DISP=OLD,
//          LABEL=(2),DSN=DFHSM,DMP.MONTHLY.VLNX300.D01245.T234610
//TARGET   DD UNIT=3390,VOL=SER=LN300,DISP=OLD
//SYSIN    DD *
RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) PURGE
/*
```

To recover a partition from a DFSMSHsm dump with DFSMSdss, use the LIST command like before to find the dump tape and dump data set from which you want to recover your partition. You also need the data set name of your partition. See [“Data set and partition naming conventions” on page 675](#) for the partition and data set naming conventions.

The following JCL recovers data set LINUX.VLNX300.PART0001.NATIVE from dump data set DFHSM.DMP.MONTHLY.VLNX300.D01245.T234610 to volume LN300.

```
//LNXJOB JOB ,USER=IBMUSER,PASSWORD=IBM,MSGLEVEL=(1,1),TIME=(5,0),REGION=4096K,
//      MSGCLASS=A,CLASS=A
//STEPDSS EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SOURCE   DD UNIT=3480,VOL=SER=(A02005,A02008,A02010),DISP=OLD,
//          LABEL=(2),DSN=DFHSM,DMP.MONTHLY.VLNX300.D01245.T234610
//TARGET   DD UNIT=3390,VOL=SER=LN300,DISP=OLD
//SYSIN    DD *
RESTORE INDDNAME(SOURCE) OUTDDNAME(TARGET) -
        DATASET(INCLUDE(LINUX.VLNX300.PART0001.NATIVE)) REPLACE
/*
```

Submitting DFSMSHsm commands using JCL and FTP

You can submit DFSMSHsm commands by sending JCL files through ftp to z/OS and have output data sets sent back to your Linux system. You must have TCP/IP and ftp running and a userid and password authorized to use ftp. Once you log into the z/OS ftp server you must enter the following command.

```
site filetype=jes
```

```
//LNXDUMP JOB ,CLASS=A,USER=IBMUSER,PASSWORD=IBM,
//      MSGCLASS=H,REGION=4096K,MSGLEVEL=(1,1)
//LISTDUMP EXEC PGM=IKJEFT01
//SYSTSPRT DD SYSOUT=A
//SYSTSIN  DD *
HSEND LIST DUMPVOLUME ODS(LIST.DUMP.VOLUMES)
//FTPDUMP EXEC PGM=FTP,COND=(0,NE,LISTDUMP),
//          PARM='1.2.3.4'
//SYSPRINT DD SYSOUT=*
//OUTPUT  DD SYSOUT=*
//SAVE    DD *
/*
//INPUT    DD *
userid
password
ascii
put 'LIST.DUMP.VOLUMES' dump.list
quit
/*
```

The IKJEFT01 program in the above JCL allows you to submit TSO line commands. The HSEND command allows DFSMSHsm commands to be submitted from TSO. After HSEND you can use any DFSMSHsm command. In the second step, the data set LIST.DUMP.VOLUMES is sent through ftp to IP address 1.2.3.4 and given a file name of dump.list.

Dataset Names	Partition Names
LINUX.Vvolser.PART0001.type	/dev/dasd/yyyy/part1
LINUX.Vvolser.PART0002.type	/dev/dasd/yyyy/part2
LINUX.Vvolser.PART0003.type	/dev/dasd/yyyy/part3

Where *volser* is the volume serial number of the volume that the data set is on. *yyyy* is the device number of the volume in the Linux environment. Type can be NATIVE or SWAP. Only NATIVE partitions need to be backed up. SWAP partitions are the Linux equivalent to page packs.

Volume serial number naming conventions

The volume serial number of a volume must be 6 characters of upper case letters or numbers. The volser can also have @ , # , and \$.

Valid Volsers

LINUXA	LNK100	\$BILLS	VOL#11
123456	A@FB2#	\$@#32A	5#2\$XA

Part 2. DFSMSHsm Storage Administration Reference

This topic contains detailed explanations of the operator, storage administrator, and system programmer commands, which are listed in alphabetical order.

Purposes of DFSMSHsm commands

- Define DFSMSHsm values and volumes,
- Determine status,
- Maintain and recover the control data sets and the DFSMSHsm address space,
- Maintain data sets and volumes,
- Operate DFSMSHsm.

Use the following commands to define the values and volumes for DFSMSHsm:

Command	Purpose
ADDVOL	Adds a new volume to the set of volumes owned or managed by DFSMSHsm and changes existing volume information defined with earlier ADDVOL commands.
AUTH	If the RACF FACILITY class is not active, DFSMSHsm uses the AUTH command to protect storage administrator DFSMSHsm commands.
DEFINE	Defines the structure of DASD level 2 volumes for DFSMSHsm and defines a data set or volume pool related to non-SMS recalls. It also defines the automatic backup cycle, the secondary space management cycle, the dump cycle, dump classes, the volumes for aggregate recovery, and times for automatically demounting tapes currently in use by data set backup.
DELVOL	Removes a volume from the list of volumes managed by DFSMSHsm and changes existing volume information.
SETSYS	Changes the values of various DFSMSHsm parameters from their current values or defaults.

Use the following commands and parameters to determine status:

Command	Purpose
AUDIT	Detects and reports discrepancies about data set information in the computing system catalog, the aggregate backup and recovery ABR record, the DFSMSHsm control data set records, and the VTOC of the DASD storage volume on which the data set resides. Note: This command also corrects discrepancies in the common recall queue.
LIST	Lists data set, user, volume, TTOC, and control data set record-serialization records from the DFSMSHsm control data sets.
QUERY	Lists values of parameters specified with DFSMSHsm commands, lists DFSMSHsm statistics, or lists the status of pending requests.
REPORT	Generates reports based on the DFSMSHsm statistics recorded in the MCDS.

Use the following commands to maintain and recover the control data sets and the DFSMSHsm address space. Note, that the first four commands are only for diagnosis and repair.

Command	Purpose
DISPLAY	Displays the contents of locations within the DFSMSHsm address space identified by an absolute address, a relative address, or a qualified address.
FIXCDS	Displays, modifies, and adds and deletes records in the MCDS, BCDS, and OCDS.
PATCH	Modifies storage within the DFSMSHsm address space.
TRAP	Specifies whether DFSMSHsm should write a log entry or cause a dump to be taken when the error specified with the TRAP command occurs.

Command	Purpose
UPDATEC	Combines the entries in the journal data set with a restored backup copy of one or all of the control data sets. See “Recovering control data sets using the enhanced CDS recovery function” on page 508 for the recommended way to recover control data sets that run outside the DFSMSHsm address space.
UPDTCDS	Update specific fields within the control data sets: the expiration date of copy pool dump copies.

Use the following commands to directly manage data sets or volumes (as opposed to the automatic management functions of DFSMSHsm):

Command	Purpose
ALTERDS	Changes the number of backup versions maintained and the frequency of creating backup versions for a non-SMS-managed data set.
ABACKUP	Backs up aggregate groups of user-defined data sets for recovery at another location.
ARECOVER	Recovers the data sets that were backed up by the ABACKUP command.
BACKDS	Creates a backup version of a specific data set.
BACKVOL	Requests that DFSMSHsm back up eligible data sets on one or more volumes, in one or more storage groups, or on all primary volumes; or that it back up the control data sets. This command can also request that DFSMSHsm invoke DFSMSdss to dump one or more volumes, or one or more storage groups.
BDELETE	Deletes all the backup versions or specific backup versions of a specific data set.
DELETE	Deletes a migrated data set without recalling the data set.
EXPIREBV	Deletes all eligible expired backup versions from all backup volumes.
FRBACKUP	Creates a fast replication backup version for each volume in a specified copy pool.
FRDELETE	Deletes one or more unneeded fast replication backup versions.
FREEVOL	Moves all migration copies meeting a specified age criteria between DFSMSHsm migration or backup volumes.
FRRECOV	Uses fast replication to recover a single volume or a pool of volumes from the managed backup versions.
MIGRATE	Moves eligible data sets from DFSMSHsm-managed volumes to migration volumes or from migration volumes to other migration volumes. It can also delete non-SMS-managed data sets from either DFSMSHsm-managed volumes or migration volumes.
RECALL	Returns migrated data sets to DFSMSHsm-managed volumes.
RECOVER	Obtains backed-up or dumped data sets and copies them to DFSMSHsm-managed volumes.
RECYCLE	Consolidates onto fewer DFSMSHsm-owned tapes valid data that is interspersed with invalid data on many DFSMSHsm-owned tapes.
SETMIG	Changes the space management status of a specific data set or group of data sets on DFSMSHsm-managed, non-SMS-managed volumes.
TAPECOPY	Makes copies of classes of DFSMSHsm-owned tapes or of individual DFSMSHsm-owned tapes and identifies the copies as alternate tapes to the original tapes.
TAPEREPL	Replaces references to original tapes with references to alternate tapes.

Use the following commands to operate DFSMSHsm:

Command	Purpose
CANCEL	Cancels existing queued DFSMSHsm requests for: <ul style="list-style-type: none"> • A specific request number • All requests for a particular user • All requests for a particular data set
HOLD	Prevents all or part of DFSMSHsm’s functions from running. Also causes the data set backup switchtapes action to occur immediately.
HSEND CMD	Issues DFSMSHsm-authorized-user commands from a TSO terminal.

Command	Purpose
LOG	Enters information into the DFSMSHsm log.
RELEASE	Releases for processing all or part of the DFSMSHsm functions that have been held.
STOP	Causes an orderly shutdown of DFSMSHsm.
SWAPLOG	Causes DFSMSHsm to exchange its LOGX data set for its LOGY data set or exchange its PDOX data set for its PDOY data set.

Command function changes for SMS

In a non-SMS environment, DFSMSHsm is controlled by commands and several parameters that describe the configuration and determine how and when DFSMSHsm functions are performed. In an SMS environment, the storage management functions defined in the management class and the storage group control the SMS-managed data sets in place of the parameter or command definitions within DFSMSHsm. The parameters formerly used on the SETSYS command and shown in [Table 54 on page 679](#) do not apply to SMS-managed data sets.

Table 54. SETSYS Parameters Not Applicable to SMS-Managed Data Sets

DFSMSHsm Parameter, Non-SMS	SMS Equivalent
DAYS	PRIMARY DAYS NON-USAGE
MIGRATIONLEVEL1DAYS	LEVEL 1 DAYS NON-USAGE
None	PARTIAL RELEASE
SETMIG (command)	COMMAND OR AUTO MIGRATE
None	ADMIN OR USER COMMAND BACKUP
None	# GDG ELEMENTS ON PRIMARY
None	ROLLED-OFF GDS ACTION
FREQUENCY	BACKUP FREQUENCY
None	GUARANTEED BACKUP FREQUENCY
VERSIONS	NUMBER OF BACKUP VERSIONS (DATA SET EXISTS)
None	NUMBER OF BACKUP VERSIONS (DATA SET DELETED)
None	RETAIN DAYS ONLY BACKUP VERSION (DATA SET DELETED)
None	RETAIN DAYS EXTRA BACKUP VERSIONS
None	EXPIRE AFTER DAYS NON-USAGE
None	EXPIRE AFTER DATE/DAYS

The functions of certain DFSMSHsm commands are assumed by management class and storage group attributes for SMS-managed data sets and volumes. [Table 55 on page 679](#) shows these commands and the SMS construct that assumes their functions.

Table 55. DFSMSHsm Commands Not Used for SMS Storage Management

DFSMSHsm Command, Non-SMS	SMS Class
DEFINE POOL	Storage group
DEFINE VOLUMEPOOL	Storage group

Table 55. DFSMSHsm Commands Not Used for SMS Storage Management (continued)

DFSMSHsm Command, Non-SMS	SMS Class
ADDVOL	Storage group
ALTERDS	Management class
SETMIG	Management class
SETMIG VOLUME	Storage group

Long-running commands

The AUDIT, EXPIREBV, LIST, RECYCLE, REPORT, and TAPECOPY commands are known as long-running commands. Long-running commands are limited to having only one of each such command being performed by DFSMSHsm at any time.

Other commands can also run for a long time. If a command is running for a long time and is returning data to your terminal, you can be prevented from performing other operations at your terminal. An example might be a LIST or AUDIT command that is returning large amounts of data. To stop such a command from interfering with your use of your terminal, have the computer system operator issue a HOLD command for the function that is causing the interference. The operator can then issue a RELEASE command for the function, but the command that was being performed does not restart. For example, if a LIST command returns data to your terminal and prevents you from using it, have the operator issue a HOLD LIST and RELEASE LIST command sequence. The list command that is causing you trouble does not restart.

Chapter 24. DFSMSHsm commands

There are four groups of DFSMSHsm commands:

- User
- Operator
- Storage administrator
- System programmer

The user commands are listed here for your information, but *z/OS DFSMSHsm Managing Your Own Data* describes them in detail. This book contains detailed explanations of the operator, storage administrator, and system programmer commands, which are listed in alphabetical order.

Several commands listed below can be specified in the ARCCMDxx PARMLIB member at startup time. The descriptions for the appropriate commands are found in "DFSMSHsm Libraries and Procedures" in the *z/OS DFSMSHsm Implementation and Customization Guide*.

User commands

The following user commands are used by application programmers, TSO terminal users, and other data processing personnel involved in storage administration:

HALTERDS

Changes the data set parameters that affect backup

HBACKDS

Backs up a specific data set

HBDELETE

Deletes a backup version of a data set

HCANCEL

Cancels an existing queued request

HDELETE

Deletes a migrated data set on a level 1 or level 2 volume

HLIST

Lists information from the migration and backup control data sets

HMIGRATE

Migrates a specific data set

HQUERY

Lists DFSMSHsm requests

HRECALL

Recalls a specific data set

HRECOVER

Recovers a backup version of a data set

Operator commands

The following operator commands are submitted primarily by a system operator from the system console. They can also be submitted by an authorized user from a TSO terminal.

CANCEL

Cancels an existing request

HOLD

Suspends all or part of DFSMSHsm functions

LOG

Enters data into the DFSMSHsm log

QUERY

Lists the status of DFSMSHsm parameters and pending requests

RELEASE

Releases for processing all or part of the DFSMSHsm functions

STOP

Shuts down DFSMSHsm

SWAPLOG

Switches between the DFSMSHsm log data sets

Storage administrator commands

The following storage administrator commands are submitted primarily by an authorized storage administrator who uses the HSEND CMD command to issue these commands from a TSO terminal. All storage administrator commands, except the HSEND CMD and the ONLYIF commands, and BEGIN and END operators, can also be submitted by a system operator from the system console.

ABACKUP

Backs up a specified aggregate group

ADDVOL

Adds or changes the volumes DFSMSHsm manages or owns

ALTERDS

Changes the data set parameters that affect back up

ALTERPRI

Alters the priority of queued DFSMSHsm requests

ARECOVER

Recovers a specified aggregate group

AUDIT

Audits DFSMSHsm

AUTH

Identifies authorized DFSMSHsm users

BACKDS

Backs up a specific data set

BACKVOL

Backs up or dumps data sets from a volume

BDELETE

Deletes backup versions of a data set

DEFINE

Defines control structures for use by DFSMSHsm

DELETE

Deletes a migrated data set from a migration volume

DELVOL

Removes a volume from the list of volumes DFSMSHsm manages or owns

EXPIREBV

Deletes unwanted backup versions from DFSMSHsm-owned storage

FRBACKUP

Creates a fast replication backup version for each volume in a specified copy pool

FRDELETE

Deletes one or more fast replication backup versions in a specified copy pool

FREEVOL

Moves migration or backup copies

FRRECOV

Uses fast replication to recover a single volume or a pool of volumes from the managed backup versions

HSEND CMD

Issues authorized user commands from a TSO terminal

LIST

Lists information from the MCDS, BCDS, and OCDS

MIGRATE

Requests a space management function

ONLYIF

Allows conditional execution of the single command, or group of commands contained within a BEGIN...END block, immediately following the ONLYIF command in ARCCMDxx.

RECALL

Recalls a specific data set

RECOVER

Recovers a backup version or dump copy of a data set or volume

RECYCLE

Consolidates valid data on one tape from other tapes

REPORT

Requests reports based on daily or volume statistics records

SETMIG

Changes the space management status of data sets or volumes

SETSYS

Establishes or changes the values of various DFSMSHsm control parameters

TAPECOPY

Copies cartridge-type single-file tape volumes

TAPEREPL

Replaces cartridge-type single-file tape volumes

UPDTCDS

Updates the control data sets

System programmer commands

The following system programmer commands are submitted primarily by an authorized system programmer who uses the HSEND CMD command to issue these commands from a TSO terminal. They can also be submitted by an authorized user from a TSO terminal or by a system operator from the system console.

DISPLAY

Displays DFSMSHsm storage locations

FIXCDS

Displays or modifies MCDS, BCDS, and OCDS records

PATCH

Modifies storage within DFSMSHsm

TRAP

Requests a dump when a specified error occurs

UPDATEC

Recovers the control data sets

Note: DISPLAY, FIXCDS, PATCH, and TRAP are commands used only for maintenance and are explained in the *z/OS DFSMSHsm Diagnosis*.

Using RACF FACILITY class to control authorization of commands

The use of the RACF FACILITY class allows security administrators to control who can issue DFSMSHsm commands. Security administrators can define RACF profiles that DFSMSHsm honors.

In order to use RACF FACILITY class checking, the RACF FACILITY class must be active when DFSMSHsm is started. If the RACF FACILITY class is active, DFSMSHsm uses RACF FACILITY class checking for all authorized and user commands. Adding or modifying the RACF FACILITY class resources does not require a restart of DFSMSHsm. If the RACF FACILITY class is used to control access to DFSMSHsm commands, the resources defined at the time a command is processed will be used to determine authorization for that command.

If the RACF FACILITY class is not active when DFSMSHsm starts, DFSMSHsm uses the AUTH command to process all storage administrator commands.

Related reading

For a full list of RACF FACILITY class resource names, see the topic about authorizing and protecting DFSMSHsm resources in *z/OS DFSMSHsm Implementation and Customization Guide*.

Using DFSMSHsm commands

Before using the DFSMSHsm commands and their parameters, you should read and understand the following use and notational conventions for DFSMSHsm commands.

A command consists of a command name usually followed by one or more operands. The maximum size of a DFSMSHsm command and its operands is 1024 bytes. All operands described in this book are called parameters, which qualify certain commands to perform the requested functions.

Example: The parameters associated with the BACKDS command specify the name of the data set that you want DFSMSHsm to back up and, optionally, the volume and unit that contains the data set to be backed up. An example of the BACKDS command is:

```
BACKDS TEST.CASES VOLUME(VOL123) UNIT(3390) /*comment*/
```

where:

BACKDS

is the command name.

TEST.CASES

is the name of the data set you want to back up.

VOLUME(VOL123)

is the parameter, specifying the volume that contains the uncataloged data set to be backed up.

UNIT(3390)

is the parameter, specifying the unit where the volume can be allocated.

/*comment*/

is a comment about the command, the parameter, or both.

All TSO conventions apply to the syntax of DFSMSHsm commands. DFSMSHsm commands use positional and keyword parameters.

Positional parameters

Positional parameters follow the command name in a prescribed sequence. There are two types of positional parameters: required and optional. You must specify required positional parameters immediately after the command name. If you use optional positional parameters, you must specify

them immediately after any required positional parameters. If the command does not have any required positional parameters, you must specify any optional positional parameters immediately after the command name. If you specify a list of items to replace a positional parameter, you must put parentheses around the list; otherwise, you do not have to use the parentheses. For example, you can specify a list of data set names with the ALTERDS command by putting parentheses around the list of data set names. If you specify only one data set name, you do not have to use parentheses.

Keyword parameters

Keyword parameters are specific words or symbols that have a meaning to DFSMSHsm. They follow positional parameters and can occur in any order.

Italic type specifies values or variables that you supply to DFSMSHsm. You can specify values or variables with some keyword parameters by putting them in parentheses after the keyword parameter. An example of a keyword with a value follows:

```
VOLUME(volser)
```

Some keyword parameters are separated by a vertical bar, for example, `TERMINAL | SYSOUT | OUTDATASET`. This bar shows keywords that conflict with one another. You can consider the bar to be a logic **OR**. The above statement would read, `TERMINAL OR SYSOUT OR OUTDATASET`. Do not use the vertical bar when you submit the command.

Note:

1. If you enter conflicting parameters in a single command, the last parameter entered in the command overrides all previously entered conflicting parameters and is the only parameter of those in conflict that is processed. All the previous conflicting parameters within the command are ignored.
2. If you enter the same parameter more than once in a single command, only the last of the repeated parameters entered in the command is processed.
3. As long as valid parameters are specified on a command, extraneous parameters are ignored.
4. Unless you are specifically restricted in a parameter explanation, you can code numeric parameters in three different ways:
 - Decimal (EBCDIC)
 - Hexadecimal (X'n')
 - Binary (B'n')

If you issue commands from the system console, you must enter the DFSMSHsm commands in the following manner:

```
F DFSMSHSM,command
```

You should replace the **DFSMSHSM** abbreviation with the procedure name in the startup procedure in SYS1.PROCLIB.

Abbreviating commands and parameters

The TSO abbreviation convention applies for all DFSMSHsm storage administrator, system programmer, and operator command names and parameters. The TSO abbreviation convention requires you to specify as much of the command name or parameter as is necessary to distinguish it from the other command names or parameters. For example, the VERSIONS parameter of the ALTERDS command can be abbreviated as V (also VE, VER, VERS, VERSI, VERSIO, and VERSION).

Some DFSMSHsm keyword parameters allow unique abbreviations. All unique abbreviations are shown in the command syntax diagrams.

Delimiters

When issuing a DFSMSHsm command, use one or more blanks to separate the command name from the first parameter. To separate succeeding parameters use one or more blanks or a comma. Do not use a semicolon as a delimiter because any characters that follow a semicolon are ignored.

Line continuations

When you continue a command to the next line during batch processing (for instance from the batch reader or a PARMLIB member), insert a plus or minus sign as the last character of each line except the last line of the command. If a comment is included on the same line, the plus or minus sign follows the comment. Refer to [“Comments” on page 686](#) for information about comments. When you use a plus sign, leading delimiters are removed from the continued line.

For commands entered from a TSO terminal, do not use a line-continuation character. All command examples in this book are commands entered for batch processing.

Example: The following is an example of a line continuation with a comment and a line continuation that does not have a comment.

```
SETSYS AUTOBACKUPSTART(1800 2100 2300) /* set start times */ +
AUTOMIGRATIONSTART(2300 0600 0800) +
BACKUP BACKUPPREFIX(HSM)
```

Data set naming conventions

When you specify a data set name with a DFSMSHsm command, the data set name must conform to TSO data set naming conventions. With DFSMSHsm, you can use the characters \$, @, and # as the first character in the data set name. A hyphen (-) can be used as a character in a data set name; however, when a hyphen is used as the last character in a data set name, it may be mistaken for a continuation symbol.

You cannot specify an alias data set name for authorized commands. See [“Storage administrator commands” on page 682](#) for a list of authorized commands.

Comments

You can add comments to DFSMSHsm commands anywhere that a blank might appear. To set off your comments, enter them between the symbols /* and */. You can continue a comment to the next line by using a line continuation character (+ or -) at the end of the line following the */ of the comment.

Example: The following is an example of using comments:

```
COMMAND1 PARAMETER1 /* comment */ +
PARAMETER2 /* comment */ +
/* more comments */ +
PARAMETER3 /* comment */
COMMAND2 PARAMETER1 +
PARAMETER2 /* comment */
```

TSO attention key

If your terminal is locked because DFSMSHsm is executing a command with the WAIT parameter specified, message ARC1020I explains that you can cancel the wait state by pressing the attention key. If you press the attention key, you will receive the following message:

```
ARC1023A CONVERTING TO A NON-WAIT WILL FAIL THE USER REQUEST,
BUT WILL ALLOW THE {RECALL | RECOVER | MIGRATE | BACKUP |
DELETE | HSEND CMD} TO COMPLETE IN THE BACKGROUND.
CONVERT Y OR N?
```


If you respond Y, you are released from the wait state while the command completes processing. If you respond N, the wait state continues until the command has completed processing.

If your terminal is locked because DFSMSHsm is executing an HLIST command with the WAIT parameter specified and if you press the attention key, you will receive the following message:

```
ARC1025A DO YOU WISH TO CANCEL DFSMSHsm LIST REQUEST? ENTER Y OR N
```

If you respond Y, DFSMSHsm terminates processing of the HLIST command. If you respond N, DFSMSHsm completes processing of the HLIST command.

Note: If you are in full screen mode when the TSO attention key is pressed (for example, ISPF browse), the screen that is displayed after the attention interrupt does not show correct information. Press the refresh key, which is usually the PA2 key, to display the correct information on your screen.

How to read syntax diagrams

See [“How to read syntax diagrams”](#) on page liv.

Considering the command sequence

In the DFSMSHsm environment, certain commands must follow a particular sequence to ensure that the command does not malfunction or fail. The following table lists these sequences:

Table 56. Command sequence	
Issue This Command	Before This Command
ADDVOL	DEFINE POOL or DEFINE VOLUMEPOOL
DEFINE DUMPCLASS	ADDVOL with either AUTODUMP or DUMPCLASS parameters
SETSYS JES2 or JES3	ADDVOL commands
SETSYS SMALLDATASET PACKING	ADDVOL with SDSP
SETSYS SYSOUT	SETSYS ACTLOGTYPE
SETSYS UNITNAME	SETSYS CDSVERSIONBACKUP
SETSYS USERUNITTABLE	ADDVOL
	ARECOVERUNITNAME(unittype)
	DEFINE with a DUMPCLASS(unittype)
	ABACKUP
	ARECOVER
	SETSYS BACKUP(tape)
	SETSYS CDSVERSIONBACKUP
	SETSYS MIGUNITNAME
	SETSYS RECYCLEOUTPUT
	SETSYS SPILL
	SETSYS TAPEMIGRATION
	SETSYS UNITNAME

Submitting DFSMSHsm commands

A DFSMSHsm user with access to the system can submit the operator, storage administrator, and system programmer commands from the system console. If you want to submit these same commands from

a TSO terminal, you must be an authorized user. To issue these commands from a TSO terminal, the authorized user must preface each command with the HSEND CMD command. For example, if you want to submit the HOLD MIGRATION command from a TSO terminal, you specify the following:

```
HSEND CMD HOLD MIGRATION
```

If you are authorized by the database authority control attribute, you can use the AUTH command to add another user to the list of users with DFSMSHsm authorization. If you are an authorized user, you can submit the AUTH command from a TSO terminal. If you submit a command from a TSO terminal, DFSMSHsm checks your TSO identification for DFSMSHsm authorization.

In systems with RACF, a component of the Security Server for z/OS, you can submit the operator, storage administrator, and system programmer commands for processing in batch mode if you are an authorized user. To do this, your TSO identification must be specified on the JOB statement of the MVS JCL (`USER=userid`). RACF scans the *userid* field and builds the control blocks DFSMSHsm uses to confirm the authorization of the person who submitted the command.

If the batch submittal fails because the user ID is missing or invalid, DFSMSHsm cannot issue a message to your terminal but notes the exception in the DFSMSHsm log. The following example shows the MVS JCL for submitting the system programmer REPORT command to request DFSMSHsm daily and volume statistics reports:

```
//REPTJOB JOB ...,USER=userid,...
//STEP1 EXEC PGM=IKJEFT01,REGION=512K
//SYSPRINT DD SYSOUT=A
//SYSTSPRT DD SYSOUT=A
//SYSTSIN DD *
        HSEND CMD REPORT DAILY FUNCTION
        HSEND CMD REPORT VOLUME FUNCTION
/*
```

Commands in a system without RACF: Method 1

In a system without RACF, you can submit operator, storage administrator, and system programmer commands for batch processing by defining the HSEND CMD command to the terminal monitor program (TMP) as an authorized command and by providing a STEPLIB or JOBLIB card to an authorized program facility (APF) authorized version of module ARCMCMD.

To use method 1, do the following:

- Instead of specifying `USER=userid` on the JOB card, add the HSEND CMD command to the authorized commands table in TMP so that this command can be invoked and submitted to DFSMSHsm as an acceptable, authorized command.
- Modify the CSECT IKJEFT2 within the IKJEFT02 load module to indicate that HSEND CMD is an authorized command and should be attached with APF authorization.
 - Make this modification to the first entry in IKJEFT2 that contains eight blanks, and ensure that one blank entry remains in the authorized command table to indicate the end of the table.
 - In the *z/OS TSO/E Customization*, the authorized command table has been moved to a CSECT in a separate load module named IKJTABLS. The IKJTABLS load module is located in the SYS1.LPALIB.
- Link-edit the DFSMSHsm module ARCMCMD (the HSEND CMD command processor) into an APF-authorized library as an authorized program.
- Ensure that the job submitting the HSEND CMD command uses a STEPLIB or JOBLIB card to this library.
- Restrict access to this APF library to prevent unauthorized use of the HSEND CMD command.

The system programmer is responsible for ensuring that any DFSMSHsm maintenance to module ARCMCMD is also applied to the authorized copy of ARCMCMD. All concatenated STEPLIBs must be authorized. The APF library name must appear either in the system LINKLIST or in the appropriate IEAAPFxx of the SYS1.PARMLIB. (The IEAAPFxx member is used to identify program libraries that

require APF authorization). See [z/OS MVS Initialization and Tuning Guide](#) for more information about the IEAAPFxx member.

The following job step link-edits the ARCMCMD module to create an authorized copy of ARCMCMD:

```
//LINKED EXEC PGM=IEWL,PARM='LIST,LET,NCAL,XREF,RENT,REUS'
//SYSPRINT DD SYSOUT=A
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSLMOD DD DISP=SHR,DSN=DFHSM.AUTHLIB
//IN DD DISP=SHR,DSN=SYS1.CMDLIB
//SYSLIN DD *
INCLUDE IN HSENDCCMD
SETCODE AC(1)
ENTRY ARCMCMD
NAME HSENDCCMD(R)
/*
```

The successful execution of this link-edit results in message IEW0461 for ARCWTU2 and ARCCVT.

Commands in a system without RACF: Method 2

You can use another method to process authorized commands from a batch TMP on a system without RACF installed. To use this option, you must ensure that the TSO user ID is stored in the protected step control block (PSCB).

First, define the user ID as an authorized user by entering the following command:

```
AUTH userid DATABASEAUTHORITY(USER)
```

The AUTH command is submitted as part of the PARMLIB member being processed during DFSMSHsm startup.

Then enter the following command:

```
SETSYS ACCEPTPSCBUSERID
```

DFSMSHsm retrieves the user ID for the TSO batch requests from the PSCB and associates it with the previously defined user ID.

Related reading

For more information about adding authorized commands to the TSO/E environment, see [z/OS TSO/E Customization](#).

Chapter 25. ABACKUP command: Backing up aggregated data sets

The DFSMSHsm ABACKUP command creates a copy of a specified, user-defined aggregate group of data sets. Aggregated data sets can be any combination of data sets treated as an entity, for example, an application. The aggregated data sets must be cataloged. A BCDS must be defined at the backup site prior to issuing an ABACKUP command; otherwise, the command fails.

Note: In some topics, you will find the term *Application Backup and Recovery* used rather than *Aggregate Backup and Recovery*. The two terms are interchangeable.

To create a copy of aggregate data sets with the ABACKUP command, you must:

- Specify the name of the predefined aggregate group that identifies the user-defined aggregated data sets.
- Specify the EXECUTE parameter of the ABACKUP command.

Note: To determine if any changes are required to either the existing SMS constructs or SMS environment prior to actual data movement, run the ABACKUP command with the VERIFY parameter prior to running the ABACKUP command with the EXECUTE parameter. The VERIFY parameter detects whether any errors exist in the aggregate backup configuration and lists all input tape volumes that are required at the aggregate backup site when a subsequent EXECUTE is submitted.

The ABARS backup error installation exit (ARCBEEEXT) allows you to skip a data set when DFSMSdss errors occur while DFSMSdss is dumping level 0 DASD data sets in the INCLUDE list. Specifying the SETSYS EXITON(BE) command is only required if you wish to skip data sets in error and it must be specified prior to issuing the ABACKUP command.

If you have volumes protected by peer-to-peer remote copy (PPRC) or extended remote copy (XRC), you can direct ABACKUP processing to skip backing up data sets from those volumes.

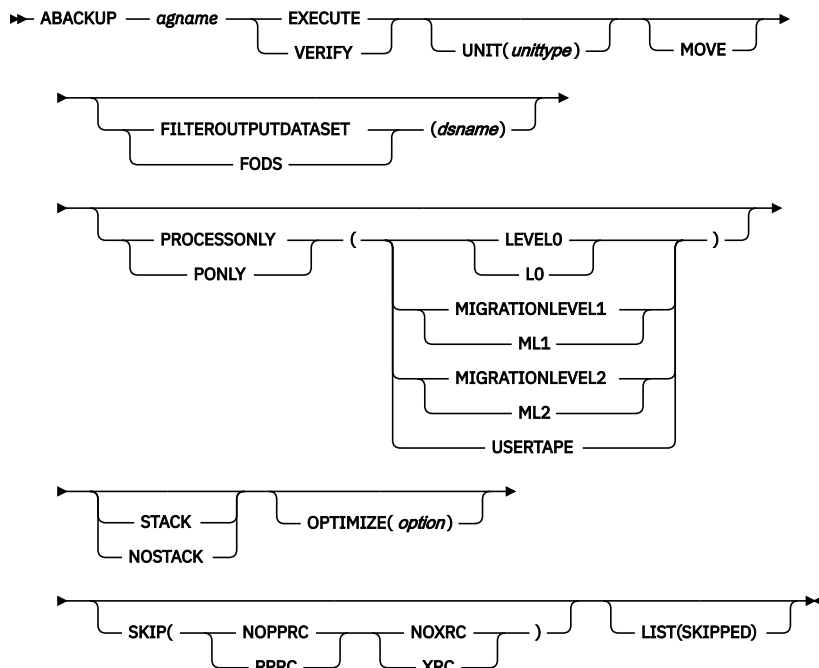
Upon successful completion of ABACKUP processing, an aggregate backup and recovery support (ABARS) activity record (ABR record) is created and written to the DFSMSHsm backup control data set (BCDS) at the backup site. When multiple ABACKUP copies are created, there is one ABR record created for each copy of the output files. See Chapter 8, “Aggregate backup and recovery support (ABARS),” on page 319 for a description of the information contained in the ABR record.

You can use the ARECOVER command to recover the data sets previously backed up with the ABACKUP command.

Note: Issuing the ABACKUP and ARECOVER commands while SETSYS DEBUG is in effect causes the commands to be processed as if the VERIFY parameter were in effect. For additional information on SETSYS DEBUG, see “Using DEBUG to determine the results of DFSMSHsm processing without moving data sets” on page 587.

For information on defining aggregate groups using ISMF panels, see [z/OS DFSMSdfp Storage Administration](#).

Syntax of the ABACKUP command



Note:

1. SMS must be active on your system before you can issue the ABACKUP command.
2. The SETSYS MAXABARSADDRESSSPACE command determines the number of ABACKUP commands that can run at one time. The maximum number is 64.
3. Only one ABACKUP command for each aggregate group name can process at one time.
4. In a multiple address space for DFSMSHsm environment, ABARS commands, like ABACKUP, can only be issued by the host identified as the main host. DFSMSHsm ignores the ABARS commands if directed to a host identified as HOSTMODE=AUX.

Required parameters of the ABACKUP command

This topic describes the required parameters of the ABACKUP command.

agname: Specifying the aggregate group name

Explanation: *agname* is a required parameter specifying the SMS aggregate group name that defines the data sets you want backed up. It contains the necessary control information to perform the aggregate backup. For *agname*, substitute the name of the SMS aggregate group.

Defaults: None.

Note: You must define the aggregate group before issuing the ABACKUP command. If you have specified a management class name in the aggregate group definition, make sure that you also define or update that management class prior to issuing the ABACKUP command. Refer to [z/OS DFSMSdfp Storage Administration](#) for information on how to define the aggregate group using ISMF panels.

EXECUTE and VERIFY: Specifying whether data movement processing or data verification occurs

Explanation: **EXECUTE** | **VERIFY** are mutually exclusive required parameters specifying whether actual data movement should occur or whether a validation of the aggregate data sets should be performed.

EXECUTE specifies that DFSMSHsm is to create the backup copies of the aggregate data sets.

VERIFY determines if errors exist that would prevent ABACKUP from completing successfully. If no errors have occurred during verification processing, VERIFY specifies that DFSMSHsm issue message ARC6166I to the aggregate backup and recovery activity log and list all data sets (migrated and nonmigrated) backed up as a result of inclusion in the INCLUDE/EXCLUDE, ACCOMPANY/ACCOMPANYEXCLUDE, ALLOCATE/ALLOCATEEXCLUDE lists of a selection data set. The data sets are not actually backed up.

Note: Either the EXECUTE or the VERIFY parameter must be used. If not, the ABACKUP command will fail.

Defaults: None.

Optional parameters of the ABACKUP command

This topic describes the optional parameters of the ABACKUP command.

FILTEROUTPUTDATASET: Specifying the output data set

Explanation: **FILTEROUTPUTDATASET(dsname)** is an optional parameter specifying the name of a data set to which the lists of data sets selected from INCLUDE/EXCLUDE, ALLOCATE/ALLOCATEEXCLUDE, ACCOMPANY/ACCOMPANYEXCLUDE filtering are written.

dsname specifies the name of the output data set. The data set name may be 1 to 44 characters in length. If the data set already exists on DASD and has the correct data set attributes, then the existing data set is used. If the data set does not exist, then a new data set is created.

Note: If the user has restricted ABACKUP command authority, then that user must have a minimum of update authority in order to append to the existing RACF protected filter output data set or allocation authority in order to create the new filter output data set.

When the FILTEROUTPUTDATASET parameter is specified along with the PROCESSIONLY parameter, then the filter output data set contains those data sets that passed the filtering criteria and also match the category designated in the PROCESSIONLY parameter.

Defaults: None.

LIST: Listing data sets not backed up due to the SKIP parameter

Explanation: LIST (SKIPPED) is an optional parameter that specifies whether DFSMSHsm should list, in the ABARS activity log, those data sets not backed up because the SKIP(PPRC) or SKIP(XRC) parameter is also specified.

The LIST parameter has no effect if you specify VERIFY.

Defaults: None.

MOVE: Removing the data sets

Explanation: MOVE is an optional parameter that specifies whether to delete, uncatalog, or both delete and uncatalog all selected data sets after a successful aggregate backup.

After a successful backup of an application, the MOVE option causes ABACKUP to delete all of the data sets in the application. The data set types in the application are affected in the following manner:

- DASD data sets are deleted
- TAPE data sets are uncataloged
- ACCOMPANY data sets are uncataloged
- ALLOCATE data sets are deleted from DASD (except ICF user catalogs in the ALLOCATE list)
- MIGRATED data sets are deleted

Defaults: None.

OPTIMIZE: Overriding the SETSYS ABARSOPTIMIZE command

Explanation: **OPTIMIZE(option)** is used to override the usage of the SETSYS ABARSOPTIMIZE command. The OPTIMIZE parameter is passed to and used by DFSMSdss when backing up level 0 DASD data sets.

The valid specifications for the *option* value are:

- If value is 1, then DFSMSdss reads one track at a time
- If value is 2, then DFSMSdss reads two tracks at a time
- If value is 3, then DFSMSdss reads five tracks at a time
- If value is 4, then DFSMSdss reads one cylinder at a time

Defaults: The default value for *option* is the value specified on the SETSYS ABARSOPTIMIZE command. If SETSYS ABARSOPTIMIZE was not specified, then this parameter defaults to 3.

PROCESSONLY: Designating the categories to process

Explanation: **PROCESSONLY** is an optional parameter that directs DFSMSHsm to process only the specified subcategories. The subcategories can be specified in any order. Specification of all four subcategories is accepted but is equivalent to not specifying the PROCESSONLY parameter.

Note: Migration copies of data sets that reside in the cloud are not supported. If you select any migration copies, the ABACKUP command fails.

The subparameters in the following table are mutually exclusive required choices.

Subparameter	Explanation
LEVEL0	PROCESSONLY(LEVEL0) specifies that only data sets residing on user DASD are processed.
MIGRATIONLEVEL1	PROCESSONLY(MIGRATIONLEVEL1) specifies that only data sets residing on DFSMSHsm migration level 1 DASD are processed.
MIGRATIONLEVEL2	PROCESSONLY(MIGRATIONLEVEL2) specifies that only data sets residing on DFSMSHsm migration level 2 tape or DASD are processed.
USERTAPE	PROCESSONLY(USERTAPE) specifies that only data sets residing on user tape are processed.

Defaults: None.

SKIP: Skipping data sets on volumes protected by PPRC or XRC

Explanation: SKIP is an optional parameter that specifies whether DFSMSHsm should skip data sets on volumes protected by PPRC or XRC.

Subparameter	Explanation
The following subparameters are mutually exclusive required choices.	
PPRC	SKIP(PPRC) If a data set in the aggregate group is on a volume or volumes protected by peer-to-peer remote copy, specifying SKIP(PPRC) causes ABACKUP to not back up the data set.
NOPPRC	SKIP(NOPPRC) If you specify SKIP(NOPPRC), this causes ABACKUP to back up the corresponding data set.
The following subparameters are mutually exclusive required choices.	
XRC	SKIP(XRC) If a data set is on a volume or volumes that are protected by extended remote copy, specifying SKIP(XRC) causes ABACKUP to not back up the data set.

Subparameter	Explanation
NOXRC	SKIP(NOXRC) If you specify SKIP(NOXRC), this causes ABACKUP to back up the corresponding data set.

Defaults: Each option for this parameter defaults to the corresponding setting of the SETSYS ABARSKIP parameter.

STACK and NOSTACK: Overriding the SETSYS ABARSTAPES setting

Explanation: **STACK** | **NOSTACK** is an optional parameter used to override the SETSYS ABARSTAPES setting.

STACK specifies that DFSMSHsm will stack the ABACKUP output files onto a minimum number of tape volumes during ABACKUP processing.

NOSTACK specifies that DFSMSHsm will not stack the ABACKUP output files during ABACKUP processing.

Defaults: This parameter defaults to the setting specified on the SETSYS ABARSTAPES command. If SETSYS ABARSTAPES was not specified, then this parameter defaults to STACK.

UNIT: Designating the output device

Explanation: **UNIT(unittype)** is an optional parameter specifying the type of tape unit where aggregate backup output files can be allocated. For *unittype*, substitute the type of tape unit. If this parameter is not specified on the ABACKUP command, the UNIT defaults to the UNIT specified in the ABARSUNITNAME parameter of the SETSYS command.

The following are valid types of tape units:

- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. If you specify:

- 3480 or 3480X, you use all the functions of the 3480 Magnetic Tape Subsystem.
- 3490, you use all the functions of the 3490 Magnetic Tape Subsystem.
- 3590-1, you use all the functions of the 3590 Magnetic Tape Subsystem.

If you define both 3592-E05 and 3592-J1A to the same esoteric group, the E05 must use the EFMT1 recording format.

You can define a 3592-E05 or 3592-J1A to the 3590-1 generic group. If so, you must ensure drive compatibility or segmentation when using the 3590-1 in a non-SMS environment.

When you are specifying esoteric names and dealing with units that can use the 3480X or 3490 Improved Data Recording Capability (IDRC), all devices (or none of them) associated with an esoteric name are required to have the 3480X or 3490 IDRC.

Note: In a tape library environment, SMS ACS routines override this parameter at allocation time.

Defaults: None.

Examples of how to code the ABACKUP command

The following examples present different ways to code the ABACKUP command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Displaying the application data sets to be backed up and verifying application backup

In this example, the data sets to be backed up, as indicated in the aggregate group PAY1, are displayed. The ABACKUP command is verified to ensure that it can complete successfully with no errors. No data sets are backed up.

```
ABACKUP PAY1 VERIFY
```

Backing up the application data sets

In this example, the data sets to be backed up, as indicated in the aggregate group PAY1, are backed up. A 3590-1 is specified as the tape unit type for the control and data files.

```
ABACKUP PAY1 EXECUTE UNIT(3590-1)
```

Backing up the application data sets to a minimum number of tapes

In this example, the data sets to be backed up, as indicated in the aggregate group PAY1, are backed up and the output files are stacked onto a minimum number of tape volumes.

```
ABACKUP PAY1 EXECUTE STACK
```

Sample activity log created when you specify ABACKUP with the VERIFY parameter

Figure 91 on page 696 is a sample activity log created when you specify ABACKUP with the VERIFY parameter. The processing has been successful. The data sets listed in the sample activity log would actually be backed up if you issued the ABACKUP command with the EXECUTE parameter.

```
PAGE 0001 Z/OS DFSMSHSM      DATA FACILITY HIERARCHICAL STORAGE MANAGER 05.081 9:29
ARC6054I AGGREGATE BACKUP STARTING FOR AGGREGATE GROUP PAY1, AT 10:56:12, STARTED TASK = DFHSMABR.ABAR0151
ARC6030I ACTIVITY LOG FOR AGGREGATE GROUP PAY1 WILL BE ROUTED TO SYSOUT=A
ARC6166I THE FOLLOWING DATA SETS WERE SELECTED TO BE PROCESSED BY AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 FOR THE
INCLUDE/EXCLUDE SPECIFICATION:
PAY2.PY002.SOCSEC
PAY2.PY002.STATUS.G0002V00
PAY2.PY002.RETIRE
PAY1.INSTRUCT
PAY1.PY001.CHECK
PAY1.PY001.NAMES
PAY1.PY001.PGMLIB
PAY1.PY002.IRA
PAY1.PY002.LADDER
ARC6166I THE FOLLOWING DATA SETS WERE SELECTED TO BE PROCESSED BY AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 FOR THE
ACCOMPANY/ACCOMPANYEXCLUDE SPECIFICATION:
PAY2.PY001.TAXES
ARC6166I THE FOLLOWING DATA SETS WERE SELECTED TO BE PROCESSED BY AGGREGATE BACKUP FOR AGGREGATE GROUP PAY1 FOR THE
ALLOCATE/ALLOCATEEXCLUDE SPECIFICATION:
PAY2.PY001.BENEFIT
ARC6250I THE FOLLOWING TAPE VOLSERS WILL BE REQUIRED DURING ABACKUP FOR AGGREGATE GROUP PAY1
A00002
A00003
ARC6055I AGGREGATE BACKUP HAS COMPLETED FOR AGGREGATE GROUP PAY1, AT 10:56:13, RETCODE = 000
```

Figure 91. Sample Activity Log Created When You Specify ABACKUP with the VERIFY Parameter

Chapter 26. ADDVOL command: Adding or changing DFSMSHsm volumes

The DFSMSHsm ADDVOL command adds new non-SMS-managed volumes to the list of volumes that DFSMSHsm manages or owns and defines the attributes of those volumes. You must issue the ADDVOL command for each non-SMS-managed volume that you want DFSMSHsm to manage or own, including those volumes defined for aggregate recovery with the DEFINE ARPOOL command. In addition, the initialization procedure for DFSMSHsm must issue an ADDVOL command for each primary and migration level 1 volume each time you start DFSMSHsm.



Attention: Do not issue simultaneous ADDVOL commands for backup volumes from multiple systems because this may, in rare cases, create duplicate BVR entries in DFSMSHsm control data sets records.

Use the ADDVOL command for the following purposes:

- Add to DFSMSHsm control a non-SMS-managed volume that has never been added before.
- Add to DFSMSHsm control a primary or migration level 1 volume that has been added previously, but not since the current DFSMSHsm startup.

Note: You do not need to specify migration level 2, backup, or dump volumes each time you start DFSMSHsm.

- Change the attributes specified for a volume previously added.
- Prevent a volume from being selected for migration output.
- Add all aggregate recovery volumes to DFSMSHsm as soon as possible after an ARECOVER command has successfully completed processing if the volumes had not been previously added.

Note:

1. In a JES3 environment, you cannot add a primary volume to DFSMSHsm after initialization.
2. If you try to issue an ADDVOL command for an SMS-managed volume, a volume whose management cannot be determined, or for a volume that is not online, the ADDVOL command is rejected.
3. The ADDVOL command must be issued on each system in the DFSMSHsm plex where this volume is to be used.

You can change attributes within each volume category, but you cannot change the volume category. Volume categories are:

- PRIMARY
- BACKUP(DAILY)
- BACKUP(SPILL)
- MIGRATIONLEVEL1
- MIGRATIONLEVEL2
- DUMP

For example, if a volume has been added to DFSMSHsm as a primary volume, you cannot use the ADDVOL command to change it to a migration or backup volume.

You can change any of the following space management techniques to be used for a primary volume:

- Migrate
- Delete-by-age
- Delete-if-backed-up

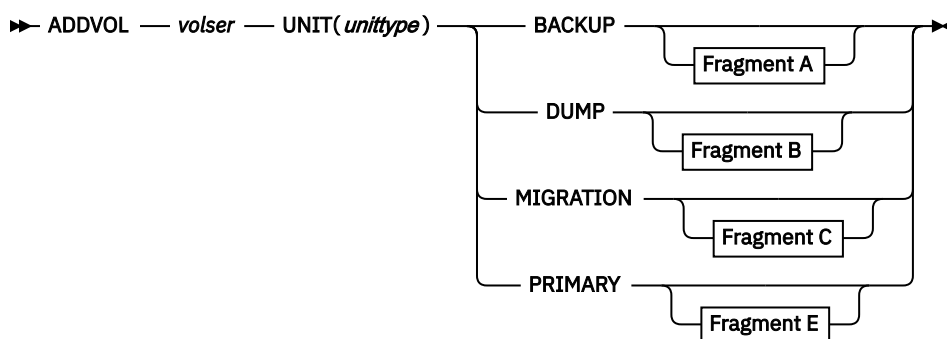
Use caution when you change the attributes of a primary volume if the volume is using migration as the space management technique. If you change the primary volume attributes so that no volume has the same attributes that existed when DFSMSHsm first migrated the data set from the primary volume, an undirected recall may fail. The undirected recall *will* fail if *all* of the following are true:

- You are using the LIKE subparameter of the SETSYS RECALL PRIVATEVOLUME or ANYSTORAGEVOLUME command.
- No available volume has a space management technique of MIGRATE and a recall technique of AUTORECALL.
- Pools are not defined.

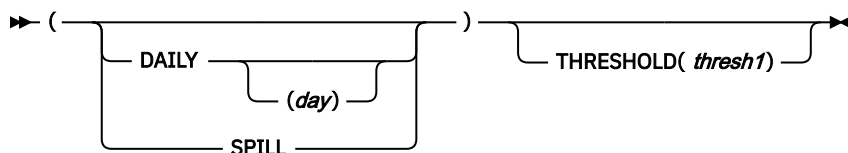
To avoid the problem, you can specify the following UNLIKE subparameters of the SETSYS RECALL command to allow the recall to occur when the attributes do not match:

- AUTOMIGRATION
- AUTOBACKUP
- NOAUTOMIGRATION
- NOAUTOBACKUP
- BACKUPDEVICECATEGORY

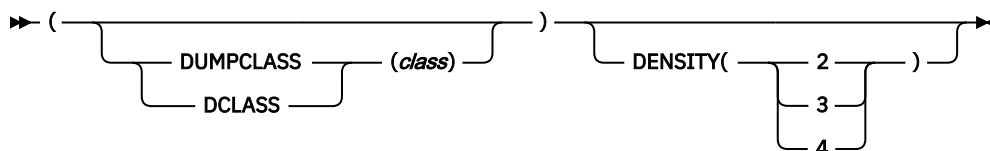
Syntax of the ADDVOL command



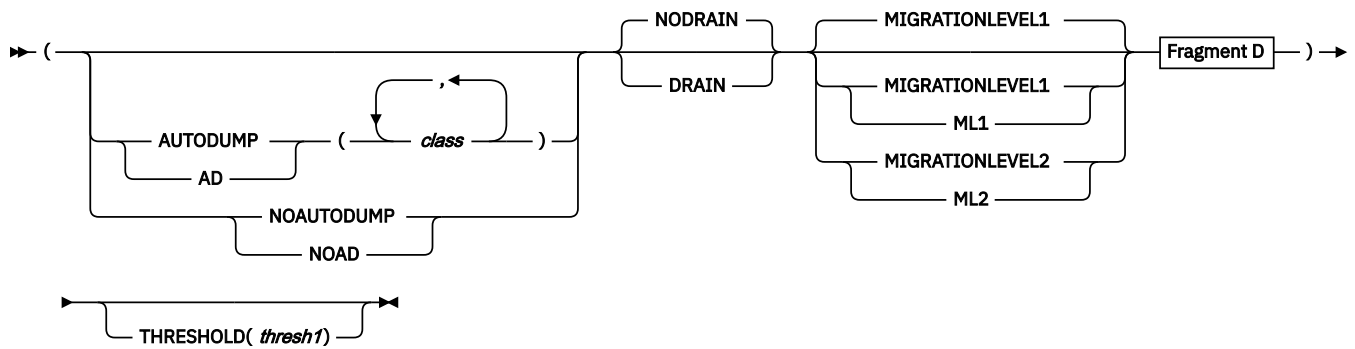
A: BACKUP optional parameters



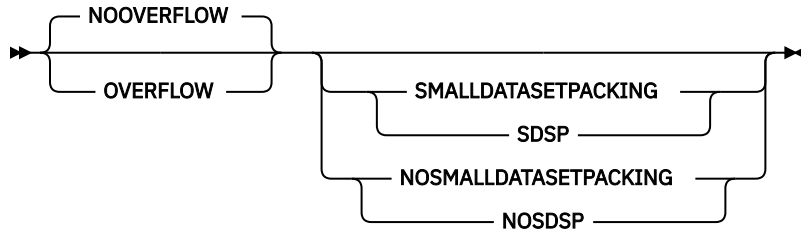
B: DUMP optional parameters



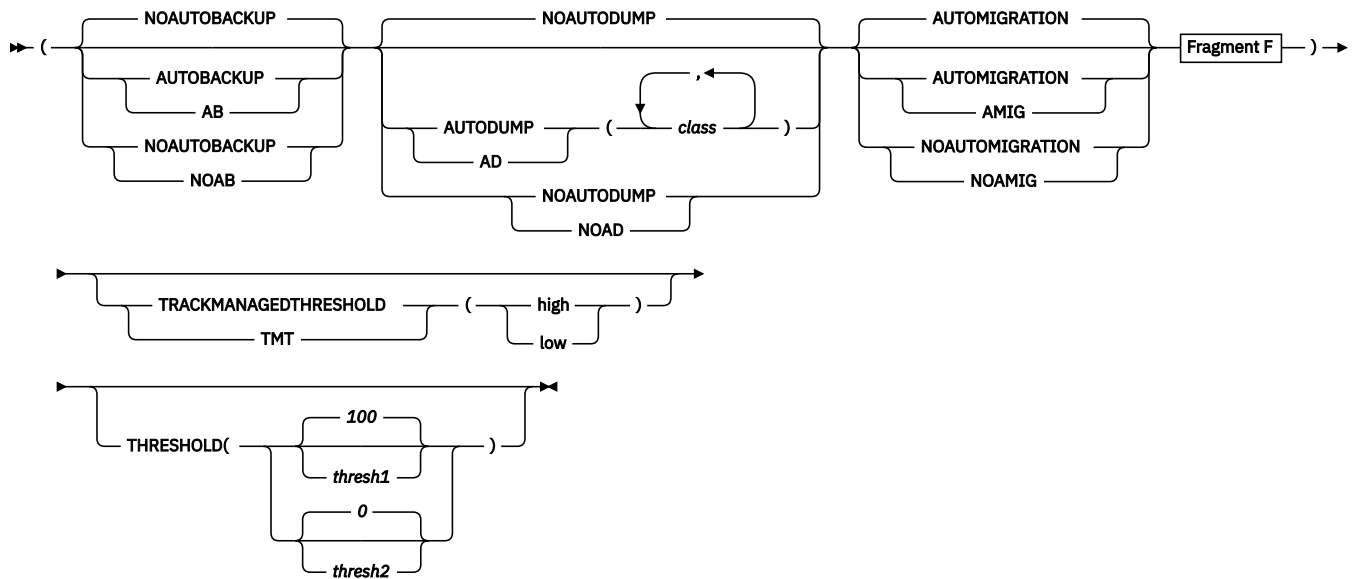
C: MIGRATION optional parameters



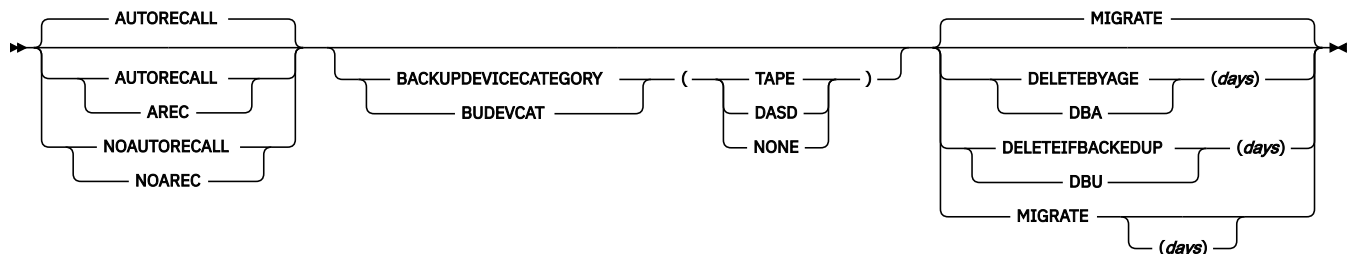
D: MIGRATION optional parameters



E: PRIMARY optional parameters



F: PRIMARY optional parameters



Note:

1. The required parameters BACKUP, DUMP, MIGRATION, and PRIMARY are volume-type parameters; that is, they specify the type of volume to be managed by DFSMSHsm.
2. The ADDVOL command cannot be used with SMS-managed volumes.

3. If multiple optional subparameters are used in one ADDVOL command, they are placed within one set of parentheses with a space between each subparameter, as shown below:

```
ADDVOL PRIM01 UNIT(3390) +
PRIMARY(AUTOBACKUP MIGRATE(8) AUTORECALL) THRESHOLD(90 70)
```

Required parameters of the ADDVOL command

This topic describes the required parameters of the ADDVOL command.

volser: Specifying the volume to be added or changed

Explanation: *volser* is a required positional parameter specifying the serial number of (1) a non-SMS-managed volume to be added to DFSMSHsm control, (2) a volume currently managed or owned by DFSMSHsm, and (3) a volume previously defined for aggregate recovery.

For *volser*, substitute the serial number of the volume being added or changed.

Defaults: None.

Note:

1. Because *volser* is a required positional parameter, you must specify it immediately after the command name.
2. When an attempt is made to add an SMS-managed volume using the ADDVOL command, the command is rejected and a message is issued.
3. A volume previously defined to aggregate recovery with the LOVOLS parameter of the DEFINE ARPOOL command can be subsequently added to DFSMSHsm only as a PRIMARY volume.
4. A volume previously defined to aggregate recovery with the ML1VOLS parameter of the DEFINE ARPOOL command can be subsequently added to DFSMSHsm only as an ML1 volume.
5. If a specific scratch volume managed by RMM is added to DFSMSHsm, RMM rejects it with message EDG4036I when DFSMSHsm attempts to use it.

BACKUP, DUMP, MIGRATION, and PRIMARY: Identifying the type of volume

Explanation: BACKUP, DUMP, MIGRATION, and PRIMARY are mutually exclusive, required parameters that identify for DFSMSHsm the type of volume you are adding or changing.

BACKUP

identifies a volume to be used as a backup volume.

DUMP

identifies a volume to be used as the output of a full-volume dump. DFSMSHsm supports only tape units for the DUMP function. The *unittype* must be a valid tape unit or the ADDVOL command will fail.

MIGRATION

identifies a volume to be used as a migration volume. Migration level 1 volumes must be DASD. Migration level 2 volumes can be either DASD or tape.

PRIMARY

identifies a volume to be used as a primary volume. Primary volumes must be DASD.

Defaults: None.

Note: If a volume record already exists for the specified *volser*, you cannot change the type of volume.

UNIT: Specifying the type of device

Explanation: UNIT(*unittype*) is a required parameter specifying the type of unit where the volume can be allocated.

For *unittype*, substitute the type of DASD or tape unit on which the volume can be allocated.

The following are valid types of DASD units:

- 3380
- 3390
- 9345

The BACKUP, DUMP, and MIGRATE functions use the same types of tape units.

The following are valid types of tape units:

- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. If you specify:

- 3480 or 3480X, you use all the functions of the 3480 Magnetic Tape Subsystem.
- 3490, you use all the functions of the 3490 Magnetic Tape Subsystem.
- 3590-1, you use all the functions of the 3590 Magnetic Tape Subsystem.

When you want to add a tape volume to DFSMSHsm's control while using the Improved Data Recording Capability (IDRC) of the 3480X, specify 3480X. If output is restricted to 3480X, this may change the unit name. It will not allow an incompatible change, such as an attempt to change a tape volume from 3480X to 3480, when that tape volume already contains compacted data. See the SETSYS TAPEHARDWARECOMPACT parameter for more information.

When you specify esoteric unit names and when you deal with units that can use the 3480X IDRC, all devices associated with an esoteric name must have the IDRC or none of them can.

Defaults: None.

Note:

1. If you specify a tape unit for a primary or migration level 1 volume, the ADDVOL command fails.
2. If you specify a DASD unit for a dump volume, the ADDVOL command fails.
3. If a DASD volume record already exists for the specified *volser*, you cannot change the *unittype*.

You can re-ADDVOL a tape with a different unit type as long as the new one is compatible with the existing device type and the unit type (such as an esoteric) is defined to MVS and known to DFSMSHsm. To make the unit type known to DFSMSHsm, use the SETSYS USERUNITTABLE command.

Optional parameters of the ADDVOL command

This topic describes the optional parameters of the ADDVOL command.

AUTOBACKUP and NOAUTOBACKUP: Specifying whether the primary volume is eligible for automatic backup

Explanation: AUTOBACKUP and NOAUTOBACKUP are mutually exclusive , optional subparameters of the PRIMARY parameter specifying whether the data sets on the primary volume are eligible for automatic backup.

AUTOBACKUP

specifies that the data sets on the primary volume are eligible for automatic backup.

NOAUTOBACKUP

specifies that the data sets on the primary volume are not eligible for automatic backup.

Defaults: If you are adding a primary volume to DFSMSHsm, the default is NOAUTOBACKUP. If you are changing the attributes of a primary volume and do not specify either subparameter, the backup status is not changed.

Note: In a multiple-DFSMSHsm-host environment, you can add the same volume to different DFSMSHsm hosts. When you do, you can also assign to that volume a different automatic backup attribute in each DFSMSHsm host.

AUTODUMP and NOAUTODUMP: Specifying whether a volume is eligible for automatic dumping to tape or cloud

Explanation: AUTODUMP(*class*) and NOAUTODUMP are mutually exclusive, optional subparameters of the PRIMARY and MIGRATION parameters specifying whether the volume is eligible for a full-volume automatic dump.

AUTODUMP

Specifies that the primary or migration volume is eligible for a full-volume automatic dump and selects the dump classes that contain the dump copies. For *class*, substitute one previously defined dump class for each of the output copies wanted for the dump. You must have a minimum of one class that is selected, but you can select a maximum of five classes. Each dump class, *class*, must be predefined using the DUMPCLASS parameter of the DEFINE command before you attempt to use the AUTODUMP parameter of the ADDVOL command.

The FREQUENCY and DAY subparameters of the DUMPCLASS parameter of the DEFINE command determine how often a primary volume is dumped to a given dump class.

NOAUTODUMP

Specifies that the primary or migration volume is not eligible for a full-volume automatic dump.

Defaults: If you are adding a primary or migration level 1 volume to DFSMSHsm, the default is NOAUTODUMP.

Note:

1. In a multiple-DFSMSHsm-host environment, the AUTODUMP parameter can be different in the different DFSMSHsm hosts. If you add a volume to more than one DFSMSHsm host with different dump class values, the dump classes are that of the last ADDVOL command processed.
2. The migration volume eligibility is applicable only to migration level 1 volumes and is ignored for migration level 2 volumes.
3. Autodump processing creates up to five dump copies (classes) if none of them designates cloud object storage.
4. You can select a maximum of five dump classes for AUTODUMP specification, including a mixture of TAPE and CLOUD classes with wanted FREQUENCY and DAY values to determine how often a primary volume is dumped to a given dump class. Your AUTODUMP and DUMPCLASS specifications should avoid having multiple CLOUD or mixed TAPE/CLOUD dump classes be eligible to dump on the same day in the autodump cycle because during an automatic dump session, if an eligible dump class designates cloud object storage, it is restricted to dumping to a single class.
5. Do not select a dump class with CLOUD attribute for migration volumes. Dumping migration volumes to cloud object store is not supported.

AUTOMIGRATION and NOAUTOMIGRATION: Specifying whether the primary volume is eligible for automatic volume space management

Explanation: AUTOMIGRATION and NOAUTOMIGRATION are mutually exclusive, optional subparameters of the PRIMARY parameter specifying whether the data sets on the primary volume are eligible for automatic volume space management.

AUTOMIGRATION

specifies that the data sets on the primary volume are eligible for automatic volume space management.

NOAUTOMIGRATION

specifies that the data sets on the primary volume are not eligible for automatic volume space management.

Defaults: If you are adding a volume to DFSMSHsm, the default is AUTOMIGRATION. If you are changing the attributes of a volume and do not specify either subparameter, the volume space management status is not changed.

Note: In a multiple-DFSMSHsm-host environment, you can add the same volume to different DFSMSHsm hosts. When you do that, you can also assign to that volume a different automatic space management attribute in each DFSMSHsm host.

AUTORECALL and NOAUTORECALL: Specifying whether the primary volume is eligible for automatic recall

Explanation: AUTORECALL and NOAUTORECALL are mutually exclusive, optional subparameters of the PRIMARY parameter specifying whether the primary volume is eligible to receive data sets that are automatically recalled.

AUTORECALL

specifies that the primary volume is eligible to receive automatically recalled data sets.

NOAUTORECALL

specifies that the primary volume is not eligible to receive automatically recalled data sets.

Defaults: If you are adding a primary volume to DFSMSHsm, the default is AUTORECALL. If you are changing the attributes of a volume and do not specify either subparameter, the recall status is not changed.

Note:

1. DFSMSHsm can override the AUTORECALL and NOAUTORECALL subparameter in certain circumstances. If you use the DEFINE command to associate the same volume to a special recall pool, the volume is always eligible to receive automatically recalled data sets.
2. If you also specify the DELETEBYAGE or DELETEIFBACKEDUP subparameter, DFSMSHsm ignores the AUTORECALL or NOAUTORECALL subparameter.

BACKUPDEVICECATEGORY: Specifying where backup versions reside

Explanation: BACKUPDEVICECATEGORY is an optional subparameter of the PRIMARY parameter specifying the type of device that receives the backup versions of the data sets when DFSMSHsm backs up the primary volume. The parameters of the BACKUPDEVICECATEGORY parameter are explained:

TAPE

specifies that the backup version of a data set on the primary volume resides on a tape daily backup volume.

DASD

specifies that the backup version of a data set on the primary volume resides on a DASD daily backup volume.

NONE

specifies that DFSMSHsm selects the first available, nonfull tape or DASD daily backup volume where the backup version of a data set resides.

Defaults: If you are adding a primary volume to DFSMSHsm and do not specify BACKUPDEVICECATEGORY, the default is NONE. If you are changing the attributes of a volume and do not specify BACKUPDEVICECATEGORY, the previous backup device category remains in effect.

Note:

1. If you change the backup device category for a volume with a subsequent ADDVOL command, DFSMSHsm could fail an undirected recall of a migrated data set when the LIKE subparameter of the SETSYS RECALL command is in effect. With the LIKE subparameter specified on the SETSYS

RECALL command, DFSMSHsm automatically recalls a data set to a volume only if that volume has the same backup device category as the primary volume that the data set used to be on. Therefore, if DFSMSHsm cannot find a primary volume with the same backup device category as the original primary volume, DFSMSHsm cannot automatically recall the migrated data set. To avoid this problem, you can specify the UNLIKE subparameter with the SETSYS RECALL command.

2. In a multiple-DFSMSHsm-host environment, you can add the same volume to different DFSMSHsm hosts. When you do that, the backup device category for that volume should be the same in all DFSMSHsm hosts or the last one specified with the ADDVOL command is the one in effect.

DAILY and SPILL: Identifying the type of backup volume

Explanation: DAILY(*day*) and SPILL are mutually exclusive, optional subparameters of the BACKUP parameter to specify the use of the backup volume.

DAILY

specifies that the volume receive backup versions of data sets from primary volumes or migration volumes. For *day*, substitute a decimal number from 1 to 31 to represent the day in the backup cycle that you want this volume assigned to. You define the backup cycle with the DEFINE command.

SPILL

specifies that the volume receive valid backup versions from DASD daily backup volumes. A tape spill backup volume can also receive all valid backup versions of data sets from a recycled tape backup volume.

Defaults: If you are adding a backup volume to DFSMSHsm and do not specify DAILY or SPILL, DFSMSHsm considers the volume to be either a daily or a spill backup volume. DFSMSHsm then determines which type of backup volume it needs when it selects a backup volume. If you are changing the attributes of a volume and do not specify either subparameter, the type of backup volume remains in effect.

If you do not specify a day with the DAILY parameter and DFSMSHsm needs another daily backup volume, DFSMSHsm assigns the volume to a day in the backup cycle.

Note: When you specify BACKUP, you can specify DAILY or SPILL, but not both. If you want to use tape for backup for a given day, you must assign at least one tape to that day in the backup cycle or on that day you must back up a volume whose backup device category is tape. Otherwise, you cannot use tape for that day in the backup cycle.

DELETEBYAGE, DELETEIFBACKEDUP, and MIGRATE: Specifying the type of space management on a primary volume

Explanation: DELETEBYAGE, DELETEIFBACKEDUP, and MIGRATE are mutually exclusive, optional subparameters of the PRIMARY parameter specifying the space management type on the primary volume.

DELETEBYAGE (*days*)

specifies that cataloged and uncataloged data sets that reside on a primary volume are scratched if they have been inactive for a specified number of days and the current expiration date has been reached. If you specify this subparameter, a current backup version of the data sets on the primary volume is not required. For *days*, if you are in a single-DFSMSHsm-host environment, substitute a number from 1 to 999. In a multiple-DFSMSHsm-host environment, you can specify a number from 2 to 999. For example, if you specify *days* as 20, DFSMSHsm scratches those data sets on a primary volume that have been inactive for at least 20 days.

DELETEIFBACKEDUP(*days*)

specifies that the data sets on a primary volume can be scratched if they have been inactive for a specified number of days and if the data sets on the primary volume have current backup versions. For *days*, in a single-DFSMSHsm-host environment, substitute a decimal number from 1 to 999. In a multiple-DFSMSHsm-host environment, you can specify a decimal number from 2 to 999. For example, if you specify *days* as 30, DFSMSHsm scratches those data sets that have been inactive for 30 days and that have a current backup version.

MIGRATE

specifies that the data sets that reside on a primary volume are migrated according to the number of days they have been inactive.

MIGRATE(days)

depends on whether you have defined valid thresholds of occupancy. If you do not define thresholds of occupancy, migration causes all data sets not accessed in the last *days* days to migrate. If you define thresholds of occupancy, DFSMSHsm migrates data sets until the low threshold of occupancy is met or until all the data sets inactive for the number of *days* specified have migrated. For *days*, you can specify a decimal number from 0 to 999. For example, if *days* is specified as 25, DFSMSHsm migrates those data sets on the primary volume that have been inactive for at least 25 days until the low threshold of occupancy is met (if you have defined thresholds of occupancy) or until all data sets inactive for 25 days or longer have been migrated.

Defaults: If you are adding a primary volume to DFSMSHsm, the default is MIGRATE. If you are changing the attributes of a volume and do not respecify a subparameter, the space management technique does not change.

Note:

1. If you do not specify *days* with the MIGRATE subparameter, or you specify 0, or you specify 1 in a multiple-DFSMSHsm-host environment when USERDATASETSERIALIZATION is not specified, the number of days the data set can remain inactive before it is allowed to migrate is the value specified with the DAYS parameter of the SETSYS command.
2. In a multiple-DFSMSHsm-host environment, you can add the same volume to different DFSMSHsm hosts. When you do that, the space management technique for that volume must be the same in all DFSMSHsm hosts, or the last one specified with the ADDVOL command is the one that is in effect.
3. DFSMSHsm automatically recalls a data set to a primary volume whose space management technique is MIGRATE even if the data set migrated from a volume whose type of space management is DELETEBYAGE or DELETEIFBACKEDUP. This is true even if you specified SETSYS RECALL(LIKE).

DENSITY: Specifying the tape density

Explanation: DENSITY is an optional parameter specifying the density of the tape dump volume. The values of the DENSITY parameter are explained:

2

specifies a tape density of 32 bytes per millimeter (800 bytes per inch).

3

specifies a tape density of 63 bytes per millimeter (1600 bytes per inch).

4

specifies a tape density of 246 bytes per millimeter (6250 bytes per inch)

Defaults: If you are adding a tape dump volume to DFSMSHsm, the default is the highest density the device can support. If you are changing the attributes of a tape dump volume and do not specify a subparameter, the density does not change.

Note:

1. The use of the DENSITY parameter is no longer supported for tape backup or ML2 volumes. The DENSITY parameter is still supported for dump volumes and for existing volumes already defined to DFSMSHsm.
2. If you specify DENSITY and UNIT, the density must match the density capability of that type of unit or the command fails. If you specify a density of 2, 3, or 4 for the 3480 Magnetic Tape Subsystem, DFSMSHsm ignores it.

DRAIN and NODRAIN: Specifying whether a volume should be selected for migration output

Explanation: DRAIN and NODRAIN are mutually exclusive, optional subparameters of the MIGRATION parameter that specify whether a volume can be selected for migration output.

DRAIN

specifies that the DASD volume be emptied. The DRAIN attribute prevents a migration level 1 volume from being selected for migration output. It prevents a migration level 2 volume from being associated with a keyrange.

NODRAIN

specifies that a migration level 1 volume can be selected for migration output or that a migration level 2 volume can be associated with a keyrange.

Defaults: When adding a DASD migration volume to DFSMSHsm, the default is NODRAIN.

If you are changing the attributes of a migration volume and do not specify either subparameter, the previous drain status is not changed.

Note:

1. DRAIN and NODRAIN pertain only to DASD volumes. The ADDVOL command fails if either parameter is specified for tape volumes.
2. The selection of a migration level 1 volume is performed as each data set is being migrated. If a migration level 1 volume has the DRAIN attribute, it will not be selected for migration output.
3. The selection of a DASD migration level 2 volume differs greatly from that of a migration level 1 volume. Migration level 2 selection is the process of associating the volume with a given keyrange of user data sets. The association of a migration level 2 volume with a particular keyrange is recorded in the migration level 2 control record (L2CR). This association takes place during:
 - Processing of the DEFINE MIGRATIONLEVEL2 command.
 - Migration of a data set whose keyrange is not associated with any volume in the L2CR.
 - Migration of a data set when an out-of-space condition occurs on the volume associated with the keyrange for that data set. In this case, another migration level 2 volume is associated with the appropriate keyrange.

The DRAIN attribute does not allow any association of a migration level 2 volume with a keyrange, and thus makes the volume not eligible for migration output. If, however, you use the ADDVOL command with the DRAIN attribute to add a migration level 2 volume that is already recorded in the L2CR, that volume remains eligible for migration output. If you want to prevent migration output eligibility for such a volume, you must (1) issue the DELVOL command with the UNASSIGN subparameter for that volume before (2) adding that volume with the ADDVOL command and the DRAIN attribute. To determine which DASD migration level 2 volumes are associated with a keyrange, use the QUERY MIGRATIONLEVEL2 command.

4. A volume can be reassigned or removed at any time and added again with the same or a different DRAIN attribute.
5. When changing the DRAIN | NODRAIN option for an ML1 volume in a multiple-DFSMSHsm-host environment, the target volume selection is affected immediately on only that host where the ADDVOL is entered. Therefore, you need to enter the ML1 DRAIN | NODRAIN option on each system that performs migration to that ML1 volume before that system performs any automatic space management.

DUMPCCLASS: Restricting the specified volume to use a specific dump class

Explanation: DUMPCCLASS(*class*) is the subparameter of the DUMP parameter restricting the specified volume to use a specific dump class. For *class*, substitute the name of a previously defined dump class for the output copy wanted for the dump.

If *class* is not specified, the dump volume can be used by any dump class. The FREQUENCY and DAY subparameters of the DUMPCLASS parameter of the DEFINE command determine how often a primary volume is dumped to a given dump class.

Defaults: None.

Note:

1. When you assign a dump volume to a specific dump class, the unit that is specified by the UNIT parameter of the ADDVOL command is not checked for compatibility with the unit that is currently defined for the dump class. This allows volumes of multiple unit types to be assigned to the same dump class at the same time. You can switch between using different unit types for the same dump class by redefining the unit for the dump class.
2. You cannot assign a dump volume to a dump class with a CLOUD attribute. The ADDVOL command fails if the specified dump class currently has a CLOUD(cloud_network_connection_name) attribute defined.

MIGRATIONLEVEL1 and MIGRATIONLEVEL2: Specifying the migration level

Explanation: MIGRATIONLEVEL1 and MIGRATIONLEVEL2 are mutually exclusive, optional subparameters of the MIGRATION parameter that you use to specify the level of the migration volumes.

MIGRATIONLEVEL1

specifies that the migration volume receives migrated data sets from primary volumes. A migration volume specified with this subparameter is called a migration level 1 volume. If the migration level 1 volume you are adding or changing is not mounted when you issue the ADDVOL command, DFSMSHsm allocates the volume, which sends a mount request to the operator.

MIGRATIONLEVEL2

specifies that the migration volume receives data sets migrating automatically from migration level 1 volumes and by command from primary volumes or migration level 1 volumes. A migration volume specified with this subparameter is called a migration level 2 volume.

Defaults: If you are adding a migration volume to DFSMSHsm, the default is MIGRATIONLEVEL1. If you are changing the attributes of a migration volume, it remains at the same level as it was when it was added to the list of volumes owned or managed by DFSMSHsm. If you are changing the attributes of a migration volume and you specify the incorrect level, the ADDVOL command fails.

Note: If DFSMSHsm migrates data sets from primary volumes directly to a tape migration level 2 volume, you can decide not to add additional migration level 1 volumes to this DFSMSHsm host. However, you must have at least one migration level 1 volume added to this DFSMSHsm host if:

- SMS data sets require migration level 1 volumes as specified by their management class attributes
- You allow backup versions generated using the BACKDS or HBACKDS commands to be written to migration level 1 volumes.
- You want DFSMSHsm to do extent reduction
- You want DFSMSHsm to do any type of volume backup and volume dump when you are keeping dump VTOC copy data sets

OVERFLOW and NOOVERFLOW: Specifying the usage of a level 1 volume for data set migration and backup versions

Explanation: OVERFLOW and NOOVERFLOW are mutually exclusive optional subparameters of the MIGRATION parameter that you use to specify how a level 1 volume is considered during selection for placement of a data set migration or backup version.

OVERFLOW

specifies that the volume is considered if either of the following are true:

1. The data you are migrating or backing up is larger than a given size, as specified on the SETSYS ML1OVERFLOW(DATASETSIZE(dssize)) command.

2. DFSMSHsm cannot allocate enough space on a NOOVERFLOW volume.

NOOVERFLOW

specifies that the volume is considered with other level 1 volumes for migration data and backup versions of any size.

Defaults: If you are adding a migration volume to DFSMSHsm, the default is NOOVERFLOW. If you are changing the attributes of a volume and do not specify either subparameter, the overflow attribute is not changed.

Note: The OVERFLOW and NOOVERFLOW subparameters do not apply to the MIGRATIONLEVEL2 parameter. If you specify the OVERFLOW or NOOVERFLOW subparameter when it does not apply, DFSMSHsm ignores it.

SMALLDATASETPACKING and NOSMALLDATASETPACKING: Specifying small data set packing on the level 1 volume

Explanation: SMALLDATASETPACKING and NOSMALLDATASETPACKING are mutually exclusive, optional subparameters of the MIGRATION parameter specifying whether small data set packing can be done on the migration level 1 volume.

SMALLDATASETPACKING

specifies that small data set packing can be done on the migration level 1 volume.

NOSMALLDATASETPACKING

specifies that small data set packing cannot be done on the migration level 1 volume.

Defaults: If you are adding a migration level 1 volume to DFSMSHsm, the default is NOSMALLDATASETPACKING. If you are changing the attributes of a volume and do not specify either subparameter, the small data set packing status does not change.

Note:

1. The SMALLDATASETPACKING and NOSMALLDATASETPACKING subparameters do not apply to the MIGRATIONLEVEL2 subparameter. If you specify the SMALLDATASETPACKING or NOSMALLDATASETPACKING parameter when it does not apply, DFSMSHsm ignores it.
2. Before small data set packing can be done on a migration level 1 volume, you must define a VSAM key-sequenced data set as the small-data-set-packing data set on the migration level 1 volume. Refer to the [z/OS DFSMSHsm Implementation and Customization Guide](#) for details about how to allocate the small-data-set-packing data set. In addition, the SMALLDATASETPACKING parameter must be specified with the SETSYS command. Small-data-set-packing data sets are not required on all migration level 1 volumes.
3. The number of SDSP data sets defined must be at least equal to the maximum number of concurrent volume migration tasks that could be executing in your complex. Additional SDSPs are recommended for RECALL processing and ABARS processing and if some SDSPs should become full during migration.

THRESHOLD: Specifying thresholds of occupancy

Explanation: THRESHOLD is an optional parameter specifying the limits for the percentages of space used on the volume.

The values you specify with THRESHOLD are used differently for the primary volumes, the migration level 1 volumes, and the DASD backup volumes that you define with ADDVOL. To be valid, threshold values must be specified from 0 to 100. A threshold value of 100, however, is interpreted to mean the volume has no thresholds defined.

THRESHOLD(thresh1)

specifies the limit (high threshold) for the percentage of occupied space you want on a volume. When *thresh1* is reached or exceeded, it causes the volume to be eligible for one of the following to occur:

- Interval migration of data sets from a primary volume
- Migration of data sets from level 1 volumes to level 2 volumes during secondary space management

DFSMSHsm uses spill or cleanup processing to reduce the amount of occupied space on a full DASD daily backup volume. When the spill process completes on a full DASD daily backup volume or the cleanup process completes on a full DASD spill backup volume and the percentage of occupied space is less than or equal to the percentage specified by *thresh1*, DFSMSHsm no longer considers the DASD volume to be full and uses it for further backup or spill processing.

THRESHOLD(*thresh2*)

specifies the limit (low threshold) for the percentage of space you want to remain occupied on a primary volume after interval migration or volume migration of data sets from the volume that has been completed.

The threshold values as they apply to each volume type are explained:

Volume Type	Threshold Values
Primary volumes	Both high and low thresholds apply. If the first limit (high threshold) is specified as 100 or if it is not greater than the second limit (low threshold), or if the low threshold is equal to zero, no thresholds are defined for the volume.
Migration level 1 volumes	Only one threshold applies to NOOVERFLOW volumes (this is the ADDVOL default). If you specify a <i>thresh2</i> values for NOOVERFLOW migration level 1 volumes then DFSMSHsm ignores it. Both thresholds are ignored for OVERFLOW volumes. You can specify THRESHOLD for the OVERFLOW volume pool by using the SETSYS ML1OVERFLOW command. See “ML1OVERFLOW: Manage ML1 OVERFLOW volumes” on page 1160 in Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.
DASD backup volumes	Only one threshold applies. If you specify a <i>thresh2</i> value for DASD backup volumes, DFSMSHsm ignores it.
Migration level 2 volumes	No threshold values applies. If you specify the THRESHOLD parameter for migration level 2 volumes, DFSMSHsm ignores it.
Tape backup volumes	No threshold values applies. If you specify the THRESHOLD parameter for tape backup volumes, DFSMSHsm ignores it.
Dump volumes	No threshold values applies. If you specify the THRESHOLD parameter for dump volumes, DFSMSHsm ignores it.

Defaults: If you are adding a primary volume to DFSMSHsm, the default value for *thresh1* is 100, and the default value for *thresh2* is 0. Therefore, a primary volume has no valid default thresholds. If you are adding a migration level 1 volume to DFSMSHsm, the default value for *thresh1* is 100. Therefore, a migration level 1 volume has no valid default threshold. If you are adding a DASD backup volume to DFSMSHsm, the default value for *thresh1* is 90.

If you are changing the attributes of a volume and do not specify threshold values, the thresholds are not changed. For a primary volume, if you specify only the high threshold, the low threshold is not changed. If, however, the new high threshold is less than or equal to the low threshold or is equal to 100, the volume no longer has valid thresholds.

Note:

1. If you do not define valid thresholds of occupancy for a primary volume, automatic migration occurs only during automatic primary space management (no interval migration occurs) and is based on the minimum migration age.
2. The *thresh1* and *thresh2* values are positional. For a primary volume, you must specify both a high threshold and a low threshold.
3. In specifying the threshold parameter for migration level 1 volumes, you want to maintain equal free space on all of your migration level 1 volumes. If you use different device types for migration level 1 volumes, you must calculate the appropriate percentages that will make the same amount of free space available on each device type. For example, if you have a mixture of 3390 models 1 and 2, you might specify 88% for model 1 (92M) and 94% for model 2 (96M).

TRACKMANAGEDTHRESHOLD: Specifying the high and low threshold for the track-managed space of a non-SMS level 0 extended address volume

Explanation: TRACKMANAGEDTHRESHOLD is an optional parameter, only valid with the PRIMARY keyword, specifying the required high and low threshold for the track-managed space of a non-SMS level-0 extended address volume (EAV).

TRACKMANAGEDTHRESHOLD is used to ensure that enough free space is maintained in the track-managed space of a volume separate from the free space on the entire volume. This is done to ensure that the track-managed space of an EAV has enough free space for data sets to be allocated on it. Even if a volume's low THRESHOLD has been met, space management will continue for data sets that reside in the track-managed space if the low TRACKMANAGEDTHRESHOLD has not been met.

If TRACKMANAGEDTHRESHOLD is not specified, then the corresponding values for THRESHOLD will be used. If no threshold values are specified, then all threshold values will use the default values. If TRACKMANAGEDTHRESHOLD is specified, then both high and low are required subparameters. If only one value is specified for TRACKMANAGEDTHRESHOLD, then the command will fail.

The following table indicates the threshold values as they apply to each volume type:

Primary Volume Type	Threshold Values
EAV volume	Both high and low thresholds apply. If the first limit (high threshold) is specified as 100 or if it is not greater than the second limit (low threshold), or if the low threshold is equal to zero, no thresholds are defined for the volume.
Non-EAV volume	If you specify the TRACKMANAGEDTHRESHOLD parameter for a non-EAV volume, the ADDVOL command will fail.

Selecting thresholds for volumes

When you choose thresholds for your volumes, consider the following:

- Size of data sets
- Rate that users require data space
- Frequency of interval migration

Follow this guideline when you set the thresholds of occupancy: The larger the data set is, the lower the threshold. Set the thresholds so users have space for their data sets but data set thrashing is minimized. *Thrashing* is a condition in which the system can do little useful work because of excessive data movement between primary and migration volumes. Data set thrashing can be caused by any of the following reasons:

- Not enough primary volume space
- The high threshold is set too low, causing DFSMSHsm to migrate data sets more often than is necessary during interval migration.

- The low threshold is set too low, causing DFSMSHsm to migrate too many data sets from primary volumes.
- The minimum migration age is set too low. DFSMSHsm uses the minimum migration age for volume processing as a lower limit to determine which data sets are eligible for migration. If the minimum migration age is too low, automatic primary space management forces the migration of data sets that were recently referred to and are likely to be referred to again soon.

Examples of how to code the ADDVOL command

The following examples present different ways to code the ADDVOL command:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Adding a primary volume

In this example, a primary non-SMS-managed volume is added to the control of DFSMSHsm. The volume is added with the characteristics of automatic volume space management, automatic backup, and automatic recall; and thresholds of occupancy are specified.

```
ADDVOL VOL004 UNIT(3390) PRIMARY(AUTOMIGRATION +
    AUTOBACKUP AUTORECALL) THRESHOLD(90 65)
```

Adding a primary extended addressable volume

In this example, a primary non-SMS-managed extended address volume is added to the control of DFSMSHsm. The extended address volume will have track-managed occupancy thresholds of 80 and 40 respectively and thresholds of occupancy for the entire volume are 85 and 60 respectively.

```
ADDVOL PRIM01 UNIT(3390) PRIMARY(AB AM) TRACKMANAGEDTHRESHOLD(80 40) THRESHOLD(85 60)
```

Adding a migration level 1 volume

In this example, a migration level 1 volume is added to the control of DFSMSHsm. The volume is added with the characteristics of no small data set packing, and a threshold of occupancy is specified.

```
ADDVOL VOL123 UNIT(3390) MIGRATION(MIGRATIONLEVEL1 +
    NOSMALLDATASETPACKING) THRESHOLD(90)
```

Adding a dump volume

In this example, a dump volume is added to the control of DFSMSHsm. The volume is specified as a dump volume with a dump class of DAILY.

```
ADDVOL DMP001 UNIT(3590-1) DUMP(DUMPCCLASS(DAILY))
```

Adding a tape spill backup volume

In this example, a backup volume is added to the control of DFSMSHsm. The volume is specified as a spill volume, and a threshold of occupancy is specified.

```
ADDVOL BKP002 UNIT(3590-1) BACKUP(SPILL) THRESHOLD(85)
```


Adding a tape daily backup volume

In this example, a tape backup volume is added. The volume is specified as a daily backup volume assigned to day 7 in the backup cycle.

```
ADDVOL BKP006 UNIT(3590-1) BACKUP(DAILY(7))
```

Adding a tape migration level 2 volume

In this example, a tape migration level 2 volume is added.

```
ADDVOL M2TP01 UNIT(3590-1) MIGRATION(MIGRATIONLEVEL2)
```

Changing the characteristics of a primary volume

In this example, the backup characteristics of a primary volume under control of DFSMSHsm are changed from no automatic backup to automatic backup, and from a low threshold of 65% to a low threshold of 75%. Because you are changing the low threshold, you must specify the high threshold again to correctly position the new value for the low threshold. You do not have to specify any parameter again for a characteristic that is not to be changed, but the volume must be specified as being in the same volume category.

```
ADDVOL VOL003 UNIT(3390) PRIMARY(AUTOBACKUP) THRESHOLD(90 75)
```

Assigning the space management attribute to a primary volume

In this example, the delete-if-backed-up space management attribute is assigned to a primary volume. A backup device category of tape is specified.

```
ADDVOL VOL005 UNIT(3390) PRIMARY(DELETEIFBACKEDUP(10) +  
    BACKUPDEVICECATEGORY(TAPE))
```

Preventing a volume from being selected for migration output

In this example, DFSMSHsm stops targeting a migration volume for any further migration or backup activity.

```
ADDVOL MIG113 U(3390) MIGRATION(ML1 DRAIN)
```

Specifying a primary volume to be automatically dumped

In this example, a primary volume is automatically dumped to specified classes (DAILY and WEEKLY) during automatic dump processing.

```
ADDVOL PRIM14 UNIT(3390) PRIMARY(AUTODUMP(DAILY,WEEKLY))
```

Specifying a migration volume to be automatically dumped

In this example, a migration volume is to be automatically dumped to specified classes (WEEKLY and MONTHLY) during automatic dump processing.

```
ADDVOL MIG019 UNIT(3390) MIGRATION(ML1 AUTODUMP(WEEKLY,MONTHLY))
```


Adding a migration volume previously defined for aggregate recovery

In this example, a volume that had previously been defined as a migration level 1 volume for aggregate recovery with the DEFINE ARPOOL command is added to DFSMSHsm.

```
ADDVOL MIG123 UNIT(3390) MIGRATION(ML1)
```

Specifying a migration volume to hold large backup or migration versions

In this example, a ML1 volume becomes an overflow volume, a repository for large data set backup or migration versions. The size of data sets that will backup or migrate to this volume will be determined by the SETSYS ML1OVERFLOW command. See [“ML1OVERFLOW: Manage ML1 OVERFLOW volumes” on page 1160 in Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057.](#)

```
ADDVOL MIG201 UNIT(3390) MIGRATION(ML1 OVERFLOW)
```


Chapter 27. ALTERDS command: Changing the backups for a non-SMS-managed data set

The DFSMSHsm ALTERDS command alters the number of backup versions that are maintained for a non-SMS-managed data set or UNIX file and the frequency of creating the backup versions for the data set or UNIX file.

If an ALTERDS command is issued for an SMS-managed data set or UNIX file that is contained in an SMS-managed zFS, the command fails and a message is issued. The number of versions and frequency of backup for an SMS-managed data set is determined from the data set's management class attributes. The number of versions of backup for a UNIX file that is contained in an SMS-managed zFS is determined from the management class of the zFS data set.

You can either specify new values with the VERSIONS and FREQUENCY parameters or revert to the DFSMSHsm default values with the SYSVERSIONS or SYSFREQUENCY parameters.

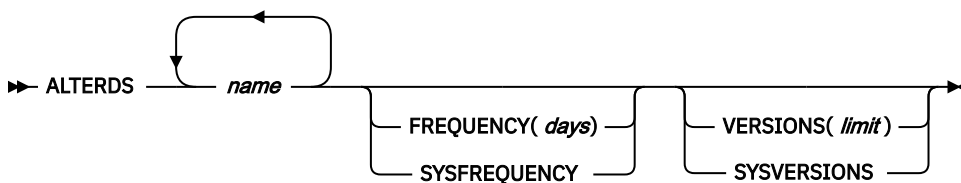
- When you specify a frequency of backup for your data set, the value overrides any frequency value that you specified with the BACKVOL command or the current DFSMSHsm value for FREQUENCY.
- When you specify a value for the VERSIONS parameter for your data set or UNIX file, the value overrides the current DFSMSHsm value for VERSIONS.
- When you specify SYSVERSIONS or SYSFREQUENCY, DFSMSHsm uses the value that you specified with the VERSIONS parameter or the FREQUENCY parameter of the SETSYS command.

If you did not use the SETSYS command to specify these values, DFSMSHsm uses the current DFSMSHsm default values for the number of backup versions or the frequency of creation of the backup versions.

You do not have to repeat any of these parameters each time you start DFSMSHsm because the values are stored in the BCDS data set record. The record is not deleted even if all of the backup versions for the data set or UNIX file are deleted. To delete this record from the BCDS, along with the record of each backup version, use the ALTERDS command with the SYSVERSIONS and SYSFREQUENCY parameters (reverts to DFSMSHsm default values). Then use the BDELETE command to delete the backup versions.

If you specify the ALTERDS command without any parameters, you receive a message that the command has completed successfully. DFSMSHsm, however, does not change any of the data set or UNIX file parameters that affect backup.

Syntax of the ALTERDS command



Note:

1. You can specify VERSIONS or SYSVERSIONS, FREQUENCY or SYSFREQUENCY, or a combination of these parameters with the ALTERDS command, but you must specify at least one parameter with the ALTERDS command to change the backup parameters.
2. Due to the order of processing to determine eligibility for backup, DFSMSHsm checks certain conditions prior to processing the ALTERDS parameters. For example, DFSMSHsm fails a data set with an unsupported data set organization prior to checking the ALTERDS parameters.

Required parameters of the ALTERDS command

This topic describes the required parameter of the ALTERDS command.

name: Specifying the name of the data set or UNIX file being altered

Explanation: *name* ... is a required positional parameter specifying the fully qualified data set or absolute UNIX path name or list of fully qualified data sets of the non-SMS-managed data sets or UNIX files whose backup attributes are being changed.

Defaults: None.

Note:

1. Because *name* ... is a required positional parameter, you must specify it immediately after the command name.
2. DFSMSHsm does not handle partitioned data set members individually. If you specify a member name of a partitioned data set, DFSMSHsm changes the backup parameters for the entire partitioned data set.
3. The UNIX path name must be enclosed in single quotation marks and be an absolute path that starts from root (/).

Optional parameters of the ALTERDS command

This topic describes the optional parameters of the ALTERDS command.

FREQUENCY and SYSFREQUENCY: Specifying the frequency of backup versions

Explanation: **FREQUENCY** and **SYSFREQUENCY** are mutually exclusive, optional parameters specifying the number of days between backup versions of data sets. If the data set has changed since the last time it was backed up and the specified number of days (frequency) has elapsed since the last backup version was created, a new backup version of the data set is created during incremental volume backup. If, however, the specified number of days has elapsed but the data set was not changed, a new backup version of the data set is not created during volume backup unless you specified the TOTAL parameter of the BACKVOL command.

FREQUENCY(days) is specified to change the number of days between backup versions of the specified data set. For *days*, substitute a decimal number from 0 to 999. For example, if you specify *days* as 5, DFSMSHsm backs up the data set if the data set has changed since the last time DFSMSHsm backed it up and the latest backup version is at least five days old. If you specify 0, DFSMSHsm creates a backup version of a changed data set every day that automatic backup runs.

SYSFREQUENCY is specified to revert to the value you specified with the SETSYS command or to the DFSMSHsm default for the frequency of data set backup.

Defaults: None.

Note: Specify the SYSFREQUENCY parameter only if a previous ALTERDS command has changed the frequency of backup versions for this data set and if you want to return to the frequency specified on the SYSFREQUENCY setting. If you specify the FREQUENCY parameter of the BACKVOL command and also specify the SYSFREQUENCY parameter of the ALTERDS command, DFSMSHsm uses the frequency set in the BACKVOL command. DFSMSHsm determines the frequency of backup in descending order:

- ALTERDS
- BACKVOL
- SETSYS
- DFSMSHsm default

VERSIONS and SYSVERSIONS: Specifying the number of backup versions to maintain

VERSIONS and SYSVERSIONS are mutually exclusive, optional parameters specifying the maximum number of backup versions to be kept when additional versions are made for the data set or UNIX file.

VERSIONS(*limit*) is specified to change the number of backup versions to be kept for the specified data set or UNIX file. The maximum number of backup versions is limited to 29 versions or 100 versions based on the BCDS record length. The following values are the valid maximum allowable number of backup versions for different BCDS record lengths:

- Record length of 2040 to 6543 — 29 versions maximum
- Record length of 6544 or more — 100 versions maximum

For *limit*, substitute a decimal number from 0 to 100. If you specify 0, DFSMSHsm does not back up the data set.

SYSVERSIONS is specified to revert to the value you specified with the SETSYS command or to the DFSMSHsm default for the number of backup versions to be kept for backup data sets.

Defaults: None.

Note:

1. Specify the SYSVERSIONS parameter only if a previous ALTERDS command has changed the number of backup versions for this data set or UNIX file and if you want to return to the number of backup versions specified on the SYSVERSIONS setting. DFSMSHsm determines the number of backup versions in the following order:

- ALTERDS
- SETSYS
- DFSMSHsm default

2. If you reduce the number of backup versions with this command, DFSMSHsm does not delete existing backup versions when the ALTERDS command is processed.

The existing excess backup versions will be deleted to match the specified limit at the time a new backup version is created for the data set or UNIX file. The EXPIREBV command can be used to delete unwanted backup versions of data sets or UNIX files without creating any new backup versions.

3. DFSMSHsm can keep a maximum of 29 versions or 100 versions, depending on the BCDS record length. Some versions may be kept for a cataloged data set with that name, and some versions kept for uncataloged data sets with that name.

The VERSIONS or SYSVERSIONS parameter indicates the number of backup versions kept for (1) the cataloged data set and (2) the uncataloged data sets. UNIX files are always considered to be cataloged. DFSMSHsm gives priority, if necessary, to versions of the cataloged data set. For example, if you specify VERSIONS(15), when the 15th backup version of the cataloged data set is made, no more than 14 versions will be kept for the uncataloged data sets.

4. If the file was backed up by a NEWNAME request, and the directory did not exist at the time of backup, DFSMSHsm was unable to store the name of the containing zFS. Thus DFSMSHsm is unable to use that zFS data set name for SMS-management checking. DFSMSHsm will then check for current existence of the file, and use the current containing zFS data set for the SMS-management check. If the file also does not exist, DFSMSHsm cannot determine whether the file is SMS-managed or not and the ALTERDS will fail.

Examples of how to code the ALTERDS command

The following examples present different ways to code the ALTERDS command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Changing the number of elapsed days between backup versions

In this example, the number of elapsed days between creation of backup versions of the specified data set is changed to 3.

```
ALTERDS PAC4485.INPUT.LOAD FREQUENCY(3)
```

Changing the maximum number of backup versions that are kept

In this example, the maximum number of backup versions kept for the specified data set is changed to twenty-one.

```
ALTERDS JAB3364.REPORT.TEXT VERSIONS(21)
```

Specifying zero number of backup versions that are kept

In this example, you do not want DFSMSHsm to keep any backup versions of the specified data set.

```
ALTERDS RTJ2219.REPORT.TEXT VERSIONS(0)
```


Chapter 28. ALTERPRI command: Altering the priority of queued requests

The ALTERPRI command allows you to alter the priority of queued requests for some DFSMSHsm functions. You can alter the priority of queued requests by the request number, by the user ID that issued the request, or by the data set, or UNIX file named in the request. You can alter the priority to make the request either the highest or lowest priority in its queue.

You can use the ALTERPRI command to alter the queueing priority for the following types of requests:

- ABACKUP
- ARECOVER
- BACKDS
- BACKVOL
- DELETE
- FRBACKUP
- FREEVOL
- FRRECOV
- MIGRATE
- RECALL
- RECOVER

You cannot alter the queueing priority of BACKVOL CDS commands and requests that have already been selected for processing.

Use the ALTERPRI command on an as-needed basis only. The ALTERPRI command only works for queued requests and will have no effect on any new requests.

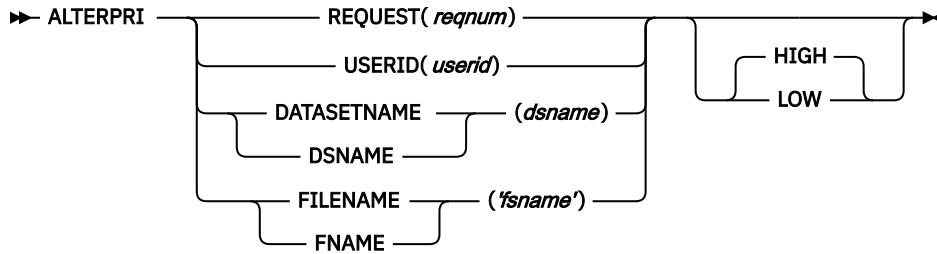
Note:

1. To prioritize a request on the Common Recall Queue, enter the ALTERPRI command on the same host that originated the recall request.
2. The BACKVOL STORAGEGROUP, BACKVOL VOLUMES, FRBACKUP DUMP, FRBACKUP DUMPONLY, and FRRECOV DSNAME commands, and requests for single volume recovery from incremental backups, each generate multiple requests (one request per volume), all with the same request number. Therefore, an ALTERPRI REQUEST command entered for one of these requests will alter the priority of all requests that are generated for the command.
3. This command is security that is protected, so a security profile must be defined and users must be given access to it before the ALTERPRI command can be issued. For information about defining FACILITY class profiles, see [z/OS Security Server RACF Security Administrator's Guide](#).
4. Changing the priority of queued requests running in the Common Dump Queue or the Common Recover Queue will be completed for queued requests that originated on the host issuing the ALTERPRI command for a request number or user ID that matches the parameters on the ALTERPRI command. A single ALTERPRI by USERID affects requests in both common queues.

The QUERY COMMONQUEUE(RECOVER) command or the QUERY COMMONQUEUE(DUMP) command can be used to determine the requests running in their respective common queues. This will identify the HOST id of the originating host for the request and can be used to determine the host from which a corresponding ALTERPRI should be issued.

For a host that is connected to a Common Dump or Recover Queue and is not the Master Scheduler, the local queue will be searched and the Master Scheduler member's group queue will be searched. For a host that is the Master Scheduler, only the local queue will be searched.

Syntax of the ALTERPRI command



Required parameters of the ALTERPRI command

This topic describes the required parameter of the ALTERPRI command.

REQUEST, USERID, and DATASETNAME, and FILENAME: Specifying the requests to be reprioritized

Explanation

REQUEST, USERID, DATASETNAME, and FILENAME are mutually exclusive, required parameters that identify the requests to be reprioritized.

REQUEST(*reqnum*)

Requests having the specified request number *reqnum* are to be altered in queuing priority.

USERID(*userid*)

Requests issued by the specified user *userid* are to be altered in queuing priority.

userid is an 8-character TSO/E user ID. To alter the priority for operator-initiated recalls, specify a value of **OPER*.

DATASETNAME(*dsname*)

Requests for the specified data set *dsname* are to be altered in queuing priority.

FILENAME(*fname*)

Requests for the specified UNIX file *fname* are to be altered in queuing priority. The specified UNIX file name must be enclosed in single quotation marks and begin with the forward slash character.

Defaults

None.

Usage notes:

1. You can use the QUERY REQUEST command to determine the request number of a request for which you want to alter the queuing priority.

Optional parameters of the ALTERPRI command

This topic describes the optional parameter of the ALTERPRI command.

HIGH and LOW: Specifying the new priority of a request

Explanation

HIGH and LOW are mutually exclusive, optional parameters that indicate how to alter the priority of the specified requests.

HIGH

The specified request is to be given the highest priority on its respective queue.

LOW

The specified request is to be given the lowest priority on its respective queue.

Defaults

If neither the HIGH or LOW parameter is specified, the default is HIGH.

Examples of how to code the ALTERPRI command

The following examples present different ways to code the ALTERPRI command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Altering the queuing priority of all requests for a specific data set

To alter the queuing priority of all requests for a particular data set, enter the ALTERPRI command with the DATASETNAME parameter specified. In this example, all requests for data set USERA.DSET01 are updated so that they have the highest priority on their respective queues.

```
ALTERPRI DATASETNAME(USERA.DSET01)
```

Altering the queuing priority of all requests with a specific request number

To alter the queueing priority of all requests with a particular request number, enter the ALTERPRI command with the request number specified. In this example, all requests with request number 1078 are updated so that they have the highest priority on their respective queues.

```
ALTERPRI REQUEST(1078) HIGH
```

Altering the queueing priority of all requests issued by a specific user

To alter the queueing priority of all requests issued by a particular user, enter the ALTERPRI command with the userid specified. In this example, all of the requests issued by user USERA are updated so that they have the lowest priority on their respective queues.

```
ALTERPRI USERID(USERA) LOW
```

Altering the queueing priority of all operator-initiated requests

To alter the queueing priority of all operator-initiated requests, enter the ALTERPRI command with a value of **OPER* for USERID. In this example, all operator-initiated requests are updated so that they have the lowest priority on their respective queues.

```
ALTERPRI USERID(**OPER*) LOW
```

Chapter 29. ARECOVER command: Recovering a specified aggregate group

The ARECOVER command performs a recovery of specified aggregated data sets previously backed up with the ABACKUP command. A BCDS must be defined at the aggregate recovery site prior to issuing an ARECOVER command; otherwise, the command fails.

Note: In some topics, you will find the term *Application Backup and Recovery* used rather than *Aggregate Backup and Recovery*. The two terms are interchangeable.

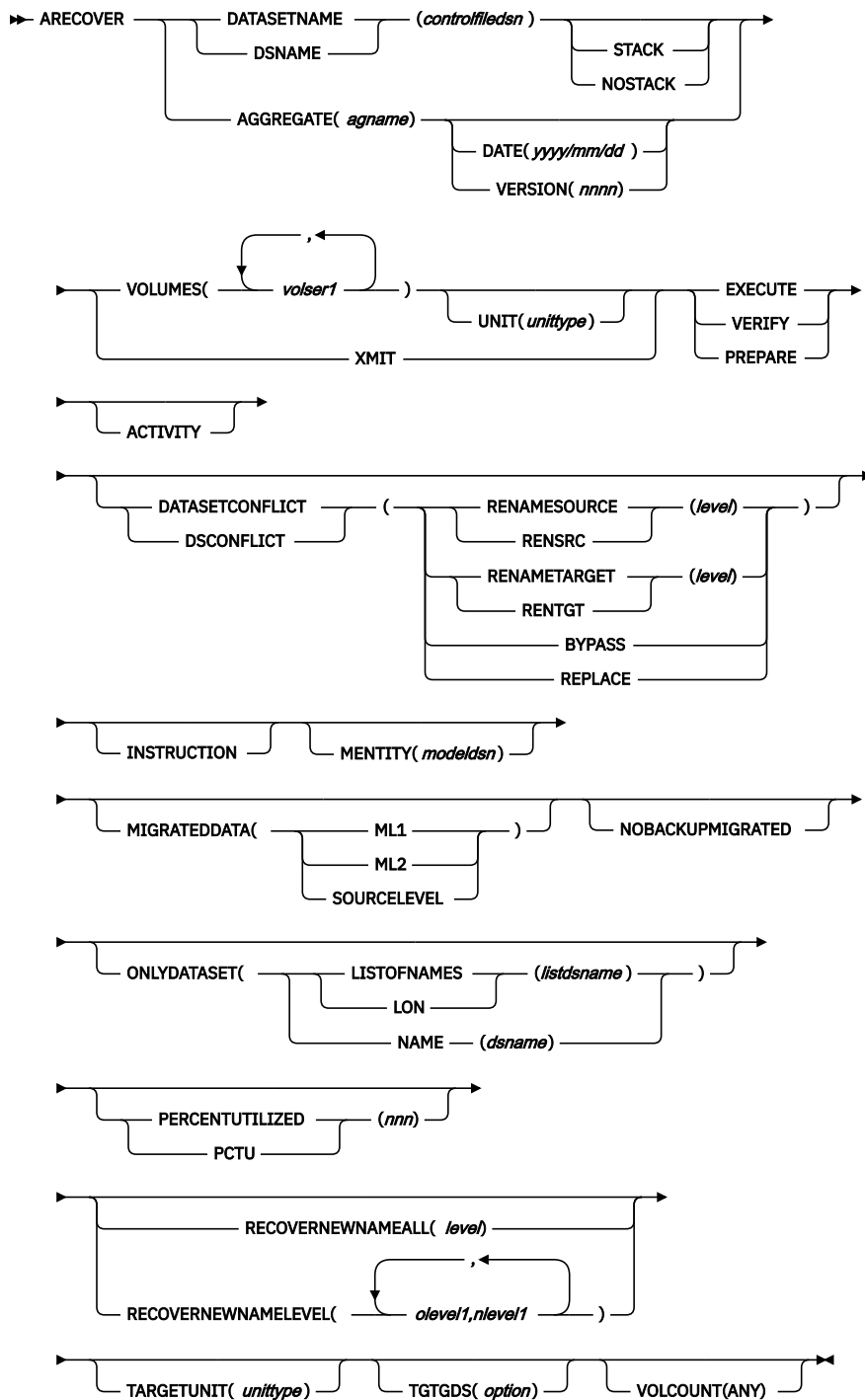
The ABR record contains the information necessary to maintain an inventory of aggregate backup and recovery activities for a particular aggregate group. To create an ABR record, issue the ARECOVER command with the PREPARE option, catalog the aggregate backup files, and add the ABACKUP output volumes to the HSMABR RACF tape volume set if requested. This allows you to issue subsequent ARECOVER commands using the aggregate group name. If the ABR record has not been created prior to issuing the ARECOVER command, the command must include the DATASETNAME and VOLUMES parameters. Once the ARECOVER command is issued and it is determined that an ABR record does not exist, one is automatically created.

If you issue the ARECOVER command with the AGGREGATE parameter, an aggregate backup and recovery support (ABARS) activity record (ABR record for the specific aggregate) must exist in the BCDS at the recovery site in order to perform a successful recovery of the previously backed-up aggregate data sets. To create an ABR record, see [Chapter 8, “Aggregate backup and recovery support \(ABARS\),” on page 319](#) for a description of the information contained in the ABR record.

You can also use the ARECOVER command to recover individual data sets from an aggregate backup version. You can specify a single fully qualified data set name on the ARECOVER command to recover a particular data set. In order to recover multiple data sets, you can supply the name of a data set that contains a list of data set names to be recovered from a specific aggregate backup version.

Note: Issuing the ABACKUP and ARECOVER commands while SETSYS DEBUG is in effect causes the commands to be processed as if the VERIFY parameter were in effect. For additional information on SETSYS DEBUG, see [“Using DEBUG to determine the results of DFSMSHsm processing without moving data sets” on page 587](#).

Syntax of the ARECOVER command



Note:

1. The SETSYS MAXABARSADDRESSSPACE command determines the number of ARECOVER commands that can run at one time. The maximum number is 64.
2. DFSMSHsm can process only one ARECOVER command for each control file data set name or aggregate group at one time.
3. The volumes must be listed in the order that they are to be mounted or the command fails.
4. Issuing ARECOVER with AGGREGATE requires that ARECOVER DATASETNAME be issued previously using the PREPARE parameter.

5. In a multiple address space for DFSMSHsm environment, ABARS commands, like ARECOVER, can only be issued by the host identified as the main host. DFSMSHsm ignores the ABARS commands if directed to a host identified as HOSTMODE=AUX.

Required parameters of the ARECOVER command

This topic describes the required parameters of the ARECOVER command.

AGGREGATE: Specifying the aggregate group to be recovered

Explanation: **AGGREGATE(*agname*)** is a required parameter that identifies the aggregate group to be recovered.

When you specify **AGGREGATE(*agname*)**, the most current version of the backup output is used for recovery unless **DATE** or **VERSION** is specified.

An ABR record must exist at the recovery site to support the **AGGREGATE** parameter. During **AGGREGATE** processing, the ABR record is read for the aggregate group. All files associated with the ABR record must be cataloged. If the files are not cataloged and ABR records exist for multiple copies, the process is repeated until an ABR record is found for one of the copies, with the proper files cataloged. If no ABR record is found with the proper files cataloged, a message is issued and the ARECOVER **AGGREGATE** fails. If this occurs, you must use **ARECOVER DATASETNAME(*controlfiledsn*)** with the **PREPARE** parameter to create the ABR record and catalog the proper files. See [“How to create the ABR record” on page 733](#) to create the ABR record.

Note:

1. Do not specify **AGGREGATE** when **DATASETNAME** is specified.
2. You must specify either **EXECUTE**, **VERIFY**, or **PREPARE** when **AGGREGATE** is specified.
3. If **PREPARE** is specified with the **AGGREGATE** parameter, **INSTRUCTION**, **ACTIVITY**, or both parameters must also be specified.

Defaults: None.

DATASETNAME: Specifying the control file

Explanation: **DATASETNAME(*controlfiledsn*)** is a required parameter used to identify the control file that contains information about the aggregated data sets previously backed up with the **ABACKUP** command.

controlfiledsn specifies the aggregate backup control file to be recovered by aggregate recovery. For *controlfiledsn*, substitute the name of the control file you want recovered. The control file name is stored in the volume header label in the format *outputdatasetprefix.C.CccVnnnn*. This is the *outputdatasetprefix* previously specified in the aggregate group.

If the control file is not catalogued at your recovery site, you can use **ARECOVER DATASETNAME(*controlfiledsn*)** with the **PREPARE** parameter to create the ABR record, which contains the control file names. See [“How to create the ABR record” on page 733](#) to create the ABR record.

Note:

1. You must specify either **VOLUMES(*volser1...*,*volsern*)** or **XMIT** when **DATASETNAME** is specified.
2. You must specify either **EXECUTE**, **VERIFY**, or **PREPARE** when **DATASETNAME** is specified.
3. Do not specify **AGGREGATE** when **DATASETNAME** is specified.

Defaults: None.

VOLUMES and XMIT: Specifying the volume type

Explanation: **VOLUMES(*volser1...*,*volsern*)** | **XMIT** are mutually exclusive, required parameters identifying the volumes associated with aggregate recovery processing.

Note: **VOLUMES** and **XMIT** are valid only when **DATASETNAME** is specified.

VOLUMES(*volser1...volsern*) specifies from 1 to 15 volumes where the control file to be recovered resides. For (*volser1...volsern*), substitute the serial numbers of the volumes where the control file resides. The serial numbers must be specified in the same order as they are listed in message ARC6061I issued at the conclusion of the ABACKUP processing.

XMIT tells DFSMSHsm that the ABACKUP output has been transmitted, and it also tells DFSMSHsm *not* to use the *volser* and *unit* information in the control file to drive the recovery. Instead, ARECOVER searches the catalog for the control and data files and, if found, uses the *unit* and *volser* from the catalog when allocating these data sets. UNIT and VOLUMES are invalid if the XMIT parameter is present, and if they are specified, ARC6002E reports an error.

Defaults: None.

EXECUTE, VERIFY, and PREPARE: Specifying whether data movement or verification should occur or whether an ABR record should be created

Explanation: **EXECUTE** | **VERIFY** | **PREPARE** are mutually exclusive, required parameters specifying whether aggregate recovery or error checking is performed or an ABR record is created.

EXECUTE specifies that DFSMSHsm perform aggregate recovery.

VERIFY specifies that DFSMSHsm should only verify the aggregate recovery configuration. No data sets will be recovered. Verification determines if errors or data set conflicts exist.

PREPARE specifies that an ABR record should be created without further verification processing or data movement if it does not already exist at the recovery site.

When PREPARE is specified, and an ABR record already exists for the specified aggregate group version and all the associated ARECOVER input files are properly cataloged, a message is issued and the PREPARE function ends. If the ABR record already exists but one or more associated files are not cataloged, a message is issued for each file that is not cataloged, the ABR record is deleted (with an accompanying message), and the PREPARE function continues.

During ARECOVER processing, when VERIFY or PREPARE is specified, messages are issued indicating the amount of storage required to perform a successful aggregate recovery. The amounts of L0, ML1, and ML2 storage required are retrieved from the ABR record and displayed separately along with the combined total storage requirement.

Note:

1. Issue the ARECOVER VERIFY command to determine the storage requirements for data movement prior to issuing the ARECOVER EXECUTE command, and ensure that sufficient storage space is available. Refer to the text below for storage requirement information when using the VERIFY parameter.
2. If the ABR record does not exist at the recovery site when EXECUTE or VERIFY is specified on an ARECOVER DATASETNAME(controlfiledsn) command, the ABR record is created at the recovery site and EXECUTE or VERIFY processing continues.
3. You must specify either EXECUTE, VERIFY, or PREPARE when DATASETNAME or AGGREGATE is specified.
4. If PREPARE is specified with the AGGREGATE parameter, INSTRUCTION, ACTIVITY, or both parameters must also be specified.

Defaults: None.

Optional parameters of the ARECOVER command

This topic describes the optional parameters of the ARECOVER command.

ACTIVITY: Recovering the activity log

Explanation: **ACTIVITY** is an optional parameter specifying the recovery of the activity log data set if one exists on the output tape for the specified aggregate group.

Specify the **ACTIVITY** parameter the first time **ARECOVER** is executed. Specifying **ACTIVITY** with the **EXECUTE**, **VERIFY**, or **PREPARE** parameters causes recovery of the activity log, if one exists. If the activity log data set does not exist on the **ABACKUP** output tapes and **ACTIVITY** is specified, an error message is issued and **ARECOVER** processing continues.

Refer to the [“Aggregate recovery output” on page 411](#) for the information contained in the activity log data set.

Defaults: None.

DATASETCONFLICT: Resolving data set naming conflicts

Explanation: **DATASETCONFLICT** is an optional parameter used to resolve any data set naming conflicts.

Note: **DATASETCONFLICT** may be specified in conjunction with the **RECOVERNEWNAMEALL** or **RECOVERNEWNAMELEVEL** parameters.

Subparameter	Explanation
RENAMESOURCE	DATASETCONFLICT(RENAMESOURCE(<i>level</i>)) specifies that data sets from the aggregate backup output files are recovered with the new high-level qualifier specified by the <i>level</i> option.
RENAMETARGET	RENAMETARGET(<i>level</i>) specifies that existing data sets at the recovery site are renamed with the high-level qualifier specified by <i>level</i> , and data sets from the aggregate backup output files are recovered with their original names. An ALTER NEWNAME is performed for the existing data set name. For a detailed explanation of the ALTER NEWNAME function, refer to z/OS DFSMS Access Method Services Commands .
BYPASS	DATASETCONFLICT(BYPASS) specifies that the data sets from the aggregate backup output files are skipped and not recovered.
REPLACE	DATASETCONFLICT(REPLACE) specifies that the existing data set at the recovery site is deleted and replaced with the data set from the aggregate backup output files.

Special care should be taken when changing the high-level qualifier so that the existing data set does not become uncataloged. This may occur when no catalog alias exists for the new high-level qualifier.

Note: When using **REPLACE**, the data set is deleted prior to attempting the recovery of the data set from the aggregate backup output files. If the recovery fails for any reason, the data set remains deleted. Reasons for failure include, among others:

- I/O errors on the input tape
- Failure to allocate the new data set
- Recovering at a site that does not support the type of data set being recovered.

Not all actions performed by **DATASETCONFLICT** subparameters are valid for all data sets. If conflicts still exist, see [“Performing aggregate recovery” on page 391](#) for more information on data set naming resolution.

Defaults: None.

DATE and VERSION: Specifying the specific aggregate group to be recovered

Explanation: **DATE** | **VERSION** are optional parameters you can use to specify the specific aggregate group to recover.

Note: DATE or VERSION are valid only when you specify AGGREGATE(*agname*).

DATE(*yyyy/mm/dd*) indicates the date of the aggregate group version you want recovered.

- *yyyy* is the 4-digit year
- *mm* is the month (01-12)
- *dd* is the day of the month (01-31)

Note: If more than one aggregate version has been created on the date you specify, the most current version on the specified date is selected.

VERSION(*nnnn*) indicates the specific version of the aggregate group to be recovered.

Defaults: If neither DATE nor VERSION is specified, the most current aggregate group version is recovered.

INSTRUCTION: Recovering the instruction data set

Explanation: **INSTRUCTION** is an optional parameter to recover the instruction data set if one exists on the output tape for the specified aggregate group at the backup site.

Specify the INSTRUCTION parameter the first time ARECOVER is executed for a given aggregate group. Specifying INSTRUCTION with the EXECUTE, VERIFY, or PREPARE parameters causes recovery of the instruction data set, if one exists. If the instruction data set does not exist on the ABACKUP output tapes and INSTRUCTION is specified, an error message is issued and ARECOVER processing continues.

Refer to the [Chapter 8, “Aggregate backup and recovery support \(ABARS\),” on page 319](#) for the information contained in the instruction data set.

Defaults: None.

MENTITY: Specifying a model entity

Explanation: **MENTITY**(*modeldsn*) is an optional parameter specifying the predefined model entity. It is used to define discrete profiles to RACF for data sets to be recovered. For *modeldsn*, substitute the data set name of the predefined model entity. Only those data sets previously protected by discrete profiles and defined to RACF before aggregate backup are protected by discrete profiles using the model entity. If the MENTITY parameter is specified in the ARECOVER command but the model entity is not predefined, aggregate recovery fails.

If MENTITY is not specified and a discrete profile is not predefined, the data set is recovered with a default discrete profile. The profile owner is set to the high level qualifier of the data set, and UACC is set to NONE. If a RACF profile has been predefined for the data sets, they are recovered with this predefined profile.

For more information on RACF discrete profiles and model entities, see [z/OS Security Server RACF Security Administrator's Guide](#).

Defaults: None.

MIGRATEDDATA: Specifying the migration level for recovering migrated data sets

Explanation: **MIGRATEDDATA** is an optional parameter specifying the migration level when recovering migrated data sets.

Note: Since ABACKUP of data sets migrated to the cloud is not supported, none will be included in the recovery.

Subparameter	Explanation
ML1	MIGRATEDDATA(ML1) specifies that all migrated data sets are recovered to migration level 1 DASD. This is the default if you do not specify MIGRATEDDATA.
ML2	MIGRATEDDATA(ML2) specifies that all migrated data sets are recovered to nonspecific tape volumes which are then managed by DFSMSHsm as migration level 2 volumes.
SOURCELEVEL	MIGRATEDDATA(SOURCELEVEL) specifies that all ML1 migrated data sets at the ABACKUP site are recovered to migration level 1 DASD; all ML2 DASD or ML2 tape migrated data sets are recovered to nonspecific tape volumes and then ADDVOLed as migration level 2 volumes.

Defaults: None.

Note: You must specify the tape unit name when recovering migrated data sets to nonspecific volumes. Use the SETSYS ARECOVERML2UNIT command to specify the tape unit name in a nonspecific volume request. Valid tape unit names or esoteric unit names specified in the SETSYS USERUNITTABLE command are also accepted. If the unit name is not specified, the default tape unit name is 3590-1. You must specify a machine type specific esoteric name when in a mixed tape machine-type environment.

NOBACKUPMIGRATED: Specifying that automatic backup does not occur for recovered migrated data sets

Explanation: **NOBACKUPMIGRATED** is an optional parameter to specify that recovered migrated data sets are *not* backed up during automatic backup. If you do not specify the parameter, DFSMSHsm backs up recovered migrated data sets residing on ML1 DASD the next time automatic backup occurs.

Defaults: None.

ONLYDATASET: Recover a single data set or a group of data sets from an aggregate group backup

Explanation: **ONLYDATASET** is an optional parameter to specify that a particular data set or a group of data sets are to be recovered from the backup copy of an aggregate group. ONLYDATASET parameter has two subparameters: NAME and LISTOFNAMES. Use NAME if you wish to only restore one fully qualified data set name from aggregate group backup tapes or LISTOFNAMES if you wish to create a file containing a list of multiple fully qualified data set names you wish to restore from the aggregate group backup tapes.

ONLYDATASET can be abbreviated as ONLYDS.

Subparameter	Explanation
NAME	ONLYDATASET(NAME(dsname)) specifies the fully qualified name of the data set to be recovered from the aggregate group backup.
LISTOFNAMES	ONLYDATASET(LISTOFNAMES(listdsname)) specifies that data sets listed in a separate file are to be recovered from the backup copy of an aggregate group. For the list file (<i>listdsname</i>), specify a single fully qualified data set name.

If you specify a data set that is not found in the aggregate group backup, the system issues an error message and the ARECOVER command ends with a non-zero return code.

To recover more than one data set from the aggregate group, use the LISTOFNAMES subparameter.

Defaults: None.

LISTOFNAMES: Recover multiple data sets from an aggregate group backup

Explanation: **LISTOFNAMES**(*listdsname*) is an optional parameter to specify that data sets listed in a separate file are to be recovered from the backup copy of an aggregate group. For the list file (*listdsname*), specify a single fully qualified data set name.

LISTOFNAMES can be abbreviated as LON.

If one or more of the data sets in the list specified on the LISTOFNAMES parameter are not found on the tapes for that aggregate version, ARECOVER will recover the data sets that are found, fail those data sets that are not found, and end with a non-zero return code.

The LISTOFNAMES data set must be a sequential data set, fixed-block, a logical record length (LRECL) of 80 and must also be cataloged. If the LISTOFNAMES data set is not cataloged or properly allocated, ARECOVER processing fails with a non-zero return code.

In the LISTOFNAMES data set, specify only one fully qualified data set name per record. Specifying multiple data set names per record, or using partially qualified data set names will cause ARECOVER processing to fail with a non-zero return code. Each record in the LISTOFNAMES data set should only contain 1 fully qualified data set name per 80 character record. There is no limit to the number of records that could be contained in the file. The LISTOFNAMES parameter should be used when you want to recover more than 1 data set from the aggregate group, but you do not wish to recover all data sets in the aggregate group.

To recover just one data set from the aggregate group, use the NAME subparameter of the ONLYDATASET parameter.

Defaults: None.

PERCENTUTILIZED: Specifying the percentage of DASD space used during ARECOVER of data sets to non-SMS volumes

Explanation: **PERCENTUTILIZED**(*nnn*) is an optional parameter that allows DFSMSdss to fill the L0 volumes in the associated ARPOOL up to 100% (100) for this aggregate.

For *nnn*, substitute an integer between 1 and 100.

Defaults: If you do not specify a PERCENTUTILIZED value, the default value is specified by the SETSYS ARECOVERPERCENTUTILIZED parameter. The SETSYS ARECOVERPERCENTUTILIZED default value is 80% (080).

RECOVERNEWNAMEALL and RECOVERNEWNAMELEVEL: Specifying a high-level qualifier for recovered data sets

Explanation: **RECOVERNEWNAMEALL** is an optional parameters that you use to specify a new high-level qualifier for recovered data sets.

RECOVERNEWNAMEALL(*level*): You use this parameter to rename all data sets that you want to recover with the high-level qualifier *level* option.

RECOVERNEWNAMELEVEL(*olevel1,nlevel1, ...*): You use this parameter to specify that all data sets having a high-level qualifier of *olevelx* are recovered with a new high-level qualifier of *nlevelx*. The maximum number of *olevelx,nlevelx* qualifier pairs is 30.

Note:

1. The RECOVERNEWNAMEALL and RECOVERNEWNAMELEVEL parameters allow data sets to be renamed before verification checks for like-named conflicts. These rename parameters apply to data sets in the INCLUDE and ALLOCATE lists (except for ICF user catalogs in the ALLOCATE list), but do not apply to ACCOMPANY data sets. In addition, migrated source VSAM data sets may not be renamed. If a rename criterion applies to a migrated source VSAM data set, a message is issued and the data set is not renamed.

2. If a rename criterion applies to a nonmigrated source VSAM data set, the new component names are not generated by ARECOVER processing but are generated by DFSMSdss if the data set name is from the INCLUDE list or by SMS VSAM allocation if the data set is from the ALLOCATE list.
3. If RECOVERNEWNAMEALL is specified, a message indicates that all data sets will be renamed with the specified level. If RECOVERNEWNAMELEVEL is specified, a message lists the level pairs specified, indicating that each data set matching the old level will be renamed to the corresponding new level. Each data set meeting a rename criterion is recovered with the new name during data movement. Like-named conflicts created by the new name are evaluated during the conflict resolution process. See [Chapter 8, “Aggregate backup and recovery support \(ABARS\),” on page 319](#) for more information about data set name conflict resolution.

Defaults: None.

STACK and NOSTACK: Specifying whether or not the ABACKUP output is STACKED

Explanation: **STACK** | **NOSTACK** are optional parameters you can use to specify to the ARECOVER DATASETNAME processing whether or not the ABACKUP output was stacked. When the ABACKUP output files are stacked, the ARECOVER processing will allocate file sequence number four.

STACK specifies that DFSMSHsm stacked the ABACKUP output files onto a minimum number of tape volumes during ABACKUP processing.

NOSTACK specifies that DFSMSHsm did not stack the ABACKUP output files during ABACKUP processing. The NOSTACK parameter allows an installation the capability to ARECOVER backups performed on a down-level version of DFSMSHsm.

TARGETUNIT: Specifying unit types to be used for ARECOVER

Explanation: **TARGETUNIT(*unittype*)** is an optional parameter specifying the tape unit type for data sets being recovered to tape devices during an aggregate recovery.

For *unittype*, substitute the type of tape unit.

The following are valid types of tape units:

- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. If you specify:

- 3480 or 3480X, you use all the functions of the 3480 Magnetic Tape Subsystem.
- 3490, you use all the functions of the 3490 Magnetic Tape Subsystem.
- 3590-1, you use all the functions of the 3590 Magnetic Tape Subsystem.

If you define both 3592-E05 and 3592-J1A to the same esoteric group, the E05 must use the EFMT1 recording format.

You can define a 3592-E05 or 3592-J1A to the 3590-1 generic group. If so, you must ensure drive compatibility or segmentation when using the 3590-1 in a non-SMS environment.

Note: When you perform an ABACKUP function with the 3480X or 3490 IDRC, you must use the same tape unit (3480X or 3490 IDRC) to do a successful RECOVER. Otherwise, the data cannot be read.

Defaults: If you do not specify this parameter, DFSMSHsm uses the unit type specified with the SETSYS ARECOVERUNITNAME parameter.

TGTGDS: Overriding the SETSYS ARECOVERTGTGDS command

Explanation: **TGTGDS(option)** is used to override the specification of the SETSYS ARECOVERTGTGDS command. The TGTGDS parameter is passed to DFSMSDss and provides greater flexibility managing SMS-managed generation data sets that are being restored to level 0 DASD.

The valid values for *option* are as follows:

Option	Explanation
DEFERRED	TGTGDS(DEFERRED) specifies that the target data set is to be assigned the DEFERRED status.
ACTIVE	TGTGDS(ACTIVE) specifies that the target data set is to be assigned the ACTIVE status, for example, rolled into the GDG base.
ROLLEDOFF	TGTGDS(ROLLEDOFF) specifies that the target data set is to be assigned the ROLLEDOFF status.
SOURCE	TGTGDS(SOURCE) specifies that the target data set is to be assigned the same status as that of the source data set.

Defaults: This parameter defaults to the setting specified on the SETSYS ARECOVERTGTGDS command. If SETSYS ARECOVERTGTGDS was not specified, then this parameter defaults to SOURCE.

UNIT: Specifying unit types used for ARECOVER

Explanation: **UNIT(unittype)** is an optional parameter specifying the tape unit type to be allocated for the recovering of data sets. The VOLUMES parameter is required when you use UNIT.

For *unittype*, substitute the type of tape unit.

The following are valid types of tape units:

- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. If you specify:

- 3480 or 3480X, you use all the functions of the 3480 Magnetic Tape Subsystem.
- 3490, you use all the functions of the 3490 Magnetic Tape Subsystem.
- 3590-1, you use all the functions of the 3590 Magnetic Tape Subsystem.

If you define both 3592-E05 and 3592-J1A to the same esoteric group, the E05 must use the EFMT1 recording format.

You can define a 3592-E05 or 3592-J1A to the 3590-1 generic group. If so, you must ensure drive compatibility or segmentation when using the 3590-1 in a non-SMS environment.

Note: When you perform an ABACKUP with the 3480X or 3490 IDRC, you must use the same tape unit (3480X or 3490 IDRC) to do a successful RECOVER. Otherwise, the data cannot be read.

Defaults: If *unittype* is not specified, the default is the *unittype* specified on the SETSYS ABARSUNITNAME command. If UNIT has not been specified with the ARECOVER command nor the SETSYS ABARSUNITNAME command, the default is 3590-1.

VOLCOUNT: Allowing data sets to expand to multivolume as necessary during ARECOVER

Explanation: **VOLCOUNT(ANY)** is an optional parameter that affects how DFSMSHsm will invoke DFSMSdss during the ARECOVER of data sets that were dumped by DFSMSdss.

ANY specifies for DFSMSHsm to pass the VOLCOUNT(ANY) parameter to DFSMSdss during the ARECOVER of data dumped by DFSMSdss (L0 data sets). This parameter affects the way SMS-managed data set allocations are performed by DFSMSdss for the L0 data sets dumped from primary volumes. Refer to [z/OS DFSMSdss Storage Administration](#) for more specific information about the restrictions and limitations of the DFSMSdss VOLCOUNT parameter. The result is the allocation of the target data sets on as many volumes as required, to a maximum of 59.

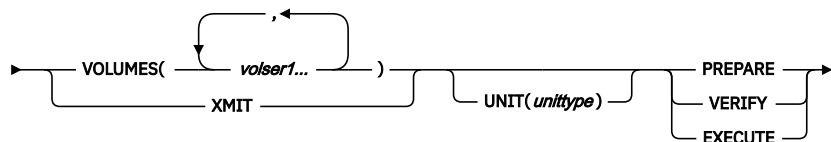
The VOLCOUNT parameter is used to override the global SETSYS ABARSVOLCOUNT environment. Your current environment can be determined by issuing the QUERY ABARS command and examining the ARC6036I messages.

Defaults: None.

How to create the ABR record

If the ABR record does not exist for any reason, one must be created. The ARECOVER command syntax to create the ABR record is shown below. Only the relevant command parameters are listed.

► ARECOVER — DATASETNAME — (— *controlfiledsn* —) ►



Note:

1. DATASETNAME is the keyword parameter used to specify the control file data set name.
2. Either VOLUMES or XMIT is required. When VOLUMES is specified (UNIT is optional), the volume list is used to allocate the control file to read it. When XMIT is specified, the required files must already be cataloged, and the system supplies the volume list.
3. PREPARE, VERIFY, and EXECUTE are mutually exclusive. Use the PREPARE parameter with ARECOVER to create the ABR record without the overhead of verification processing or data movement. If you specify VERIFY or EXECUTE, the ABR record is created followed by the VERIFY or EXECUTE processing.

Examples of how to code the ARECOVER command

The following examples reflect the simplified ARECOVER commands, which require the user only to know the name of the aggregate group that has been backed up. You do not need to enter the control file data set name (and VOLUMES/UNIT, if XMIT is not specified) each time an aggregate recovery is performed.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Preparing a new ABR record

In this example, the ARECOVER command for the aggregate group, PAY1.C.C01V0001, is prepared. This preparation eliminates the overhead of verification processing or data movement associated with the VERIFY or EXECUTE parameter. TAPE23 is specified as the volume for the aggregate group, and 3590-1 is specified as the tape unit type to be allocated for the aggregate group. INSTRUCTION recovers the instruction data set and ACTIVITY recovers the activity log data set.


```
ARECOVER DATASETNAME(PAY1.C.C01V0001) VOLUMES(TAPE23) +  
UNIT(3590-1) PREPARE INSTRUCTION ACTIVITY
```

Verifying aggregate recovery

In this example, the ARECOVER command for the aggregate group, PAY1.C.G0001V00, is verified. This verification ensures that the command can complete successfully with no errors. The data sets to be recovered, as indicated by the aggregate group, are checked to ensure that no like-named data set conflicts exist. No data sets are recovered.

```
ARECOVER AGGREGATE(PAY1) VERIFY
```

Recovering the aggregated data sets

In this example, the data sets to be recovered, as indicated by the aggregate group, PAY1.C.G0001V00, are recovered. 3590-1 is specified as the tape unit type for receiving data sets recovered to tape devices.

```
ARECOVER AGGREGATE(PAY1) EXECUTE TARGETUNIT(3590-1)
```

Replacing like-named aggregated data sets during recovery

In this example, the data sets specified by the aggregate group, PAY1.C.G0001V00, are recovered. Any like-named data set that exists at the aggregate recovery site is deleted and is replaced with the corresponding aggregate backup data set. DATASETCONFLICT (REPLACE) is the proper parameter which deletes the existing data set at the ARECOVER site and recovers the ABACKUP data set.

```
ARECOVER AGGREGATE(PAY1) EXECUTE DATASETCONFLICT(REPLACE)
```

Recovering RACF-protected aggregated data sets

In this example, the data sets to be recovered as indicated by the aggregate group, PAY1.C.G0001V00, are recovered. When those data sets previously protected by discrete profiles and defined to RACF before aggregate backup are recovered, they are protected by the predefined model entity, USRT004.MODEL.PROF.

```
ARECOVER AGGREGATE(PAY1) EXECUTE MENTITY(USRT004.MODEL.PROF)
```

Specifying the percentage of DASD space for non-SMS volumes

This example shows how to fill NON-SMS Level 0 volumes to 90% of capacity during ARECOVER processing.

```
ARECOVER AGGREGATE(PAY1) EXECUTE PERCENTUTILIZED(090)
```

Specifying the migration level of recovered migrated data sets

This example shows how to recover all migrated data sets to ML2 tape volumes during ARECOVER processing.

```
ARECOVER AGGREGATE(PAY1) EXECUTE MIGRATEDDATA(ML2)
```


Specifying a high-level qualifier

This example shows how to rename all data sets that are recovered from ABACKUP output files with the high level qualifier of PAY1. The existing high level qualifier will be replaced.

```
ARECOVER AGGREGATE(PAY1) EXECUTE RECOVERNEWNAMELEVEL(PAY1)
```

Recovering an aggregate containing L0 data sets in the include list which failed due to inadequate space

In this example, L0 data sets were not recovered successfully by DFSMSdss because an inadequate space condition occurred. The ARECOVER command can be issued again with the VOLCOUNT parameter, allowing the failing data sets to be recovered to as many volumes as needed (maximum of 59) for successful allocation.

```
ARECOVER DSNAME(PAY1.C.C01V0002) VOLUMES(TAPE24) VOLCOUNT(ANY) EXECUTE
```

Sample activity log output for the ARECOVER command

Figure 92 on page 735 is sample activity log output for the ARECOVER command.

```
PAGE 0001 Z/OS DFSMSHSM DATA FACILITY HIERARCHICAL STORAGE MANAGER 06.081 09:34
ARC6102I AGGREGATE RECOVERY STARTING USING CONTROL FILE DATA SET PAY1.C.C01V0002, AT 18:16:00,
STARTED TASK = DFSMSMR.ABAR0154
ARC6030I ACTIVITY LOG FOR CONTROL FILE DATA SET PAY1.C.C01V0002 WILL BE ROUTED TO SYSOUT=A
ARC6115I AGGREGATE RECOVERY USING CONTROL FILE DATA SET PAY1.C.C01V0002 WILL USE VOLUMES
A00020
A00016
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY001.NAMES DURING AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.RETIRE DURING AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.SOCSEC DURING AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY2.PY002.STATUS.G0002V00 DURING AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.INSTRUCT DURING AGGREGATE RECOVERY
ARC6120I CONFLICT RESOLUTION ACTION 'REPLACE' WILL BE USED FOR DATA SET PAY1.PY002.IRA DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY001.NAMES HAS BEEN UNCATALOGED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.RETIRE HAS BEEN UNCATALOGED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.SOCSEC HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.INSTRUCT HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6108I DATA SET PAY1.PY002.IRA HAS BEEN DELETED DURING AGGREGATE RECOVERY
ARC6004I 000 ARECOVER PAGE 0001 5695-DF175 DFSMSdss V1R1.0 Data Set Services 92.238 18:16
ARC6004I 000 ARECOVER ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS CHK DEFAULT TO YES
ARC6004I 000 ARECOVER RESTORE DATASET(FILTERDD(SYS00016)) -
ARC6004I 000 ARECOVER INDDNAME(SYS00015) -
ARC6004I 000 ARECOVER OUTDYNAM( -
ARC6004I 000 ARECOVER (DUMPS) -
ARC6004I 000 ARECOVER ) -
ARC6004I 000 ARECOVER PERCENTUTILIZED( -
ARC6004I 000 ARECOVER 080 -
ARC6004I 000 ARECOVER ) -
ARC6004I 000 ARECOVER ) -
ARC6004I 000 ARECOVER SPHERE -
ARC6004I 000 ARECOVER TGTGDS(SOURCE) -
PAGE 0002 Z/OS DFSMSHSM 1.9.0 DATA FACILITY HIERARCHICAL STORAGE MANAGER 06.081 09:34
ARC6004I 000 ARECOVER CATALOG FORCE
ARC6004I 000 ARECOVER ADR109I RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
ARC6004I 000 ARECOVER ADR109I RI01 (01), 92238 18:16:41 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ARC6004I 000 ARECOVER ADR006I (001)-SETUP(01), 92238 18:16:42 EXECUTION BEGINS
ARC6004I 000 ARECOVER ADR700I (001)-TDOS (01), THE INPUT DUMP DATA SET BEING PROCESSED IS IN LOGICAL DATA SET FORMAT
ARC6004I 000 ARECOVER ADR396I (001)-NEWS(01), DATA SET PAY2.PY002.SOCSEC ALLOCATED, ON VOLUME(S): DUMPS
ARC6004I 000 ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY2.PY002.SOCSEC HAS BEEN CATALOGED IN CATALOG TPCMAST1
ARC6004I 000 ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY2.PY002.SOCSEC WAS RESTORED
ARC6004I 000 ARECOVER ADR396I (001)-NEWS(01), DATA SET PAY2.PY002.STATUS.G0002V00 ALLOCATED, ON VOLUME(S): DUMPS
ARC6004I 000 ARECOVER ADR465I (001)-DALOC(01)
ARC6004I 000 ARECOVER DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN CATALOGED IN CATALOG TPCMAST1
ARC6004I 000 ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY2.PY002.STATUS.G0002V00 WAS RESTORED
ARC6004I 000 ARECOVER ADR396I (001)-NEWS(01), DATA SET PAY1.INSTRUCT ALLOCATED, ON VOLUME(S): DUMPS
ARC6004I 000 ARECOVER ADR465I (001)-DALOC(01), DATA SET PAY1.INSTRUCT HAS BEEN CATALOGED IN CATALOG TPCMAST1
ARC6004I 000 ARECOVER ADR489I (001)-TDLOG(01), DATA SET PAY1.INSTRUCT WAS RESTORED
ARC6004I 000 ARECOVER ADR454I (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
ARC6004I 000 ARECOVER PAY2.PY002.SOCSEC
ARC6004I 000 ARECOVER PAY1.INSTRUCT
ARC6004I 000 ARECOVER PAGE 0002 5695-DF175 DFSMSdss V1R1.0 Data Set Services 92.238 18:16
ARC6004I 000 ARECOVER ADR013I (001)-CLTSK(01), 92238 18:17:00 TASK COMPLETED WITH RETURN CODE 0000
ARC6004I 000 ARECOVER ADR012I (SCH)-DSSU (01), 92238 18:17:00 DFSMSdss PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC6109I REPLACE OPTION SPECIFIED,
GDG DATA SET PAY2.PY002.STATUS.G0002V00 HAS BEEN RESTORED USING EXISTING GDG ATTRIBUTES
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY RECOVERED USING AGGREGATE GROUP PAY1:
PAY1.PY001.NAMES
PAY2.PY002.RETIRE
PAY2.PY002.SOCSEC
PAY2.PY002.STATUS.G0002V00
PAY1.INSTRUCT
PAY1.PY002.IRA
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY ALLOCATED USING AGGREGATE GROUP PAY1:
PAY2.PY001.BENEFIT
ARC6116I THE FOLLOWING DATA SETS WERE SUCCESSFULLY CATALOGED USING AGGREGATE GROUP PAY1:
PAY2.PY001.TAXES
ARC6103I AGGREGATE RECOVERY HAS COMPLETED FOR AGGREGATE GROUP PAY1, USING CONTROL FILE DATA SET PAY1.C.C01V0002,
AT 18:17:31, RETCODE = 000
```

Figure 92. ARECOVER sample activity log output

Chapter 30. AUDIT command: Auditing DFSMSHsm

The AUDIT command operates differently depending on whether it is issued for a control data set or for a common queue. The following two topics describe how the AUDIT command works with control data sets and common queues.

Auditing control data sets

The AUDIT command detects, reports, diagnoses, and often provides repairs for discrepancies between control data sets.

The purpose of DFSMSHsm is to manage your inactive data and to make DASD space available where it is needed. It does this by automating tasks and providing interactive tools. Automated tasks help you to move from a *user-managed* to a *system-managed* environment. The most important goal of DFSMSHsm, however, is to provide data integrity.

To ensure that you have data integrity, DFSMSHsm uses numerous data set records to track individual data sets. These records are contained in:

- Master catalog, which is a list of data sets for the entire system
- User catalog, which is a list of data sets accessible from that catalog
- Journal, which keeps a running record of backup and migration transactions
- SDSP data sets contained on migration volumes
- MCDS, which is an inventory of migrated data sets
- BCDS, which is an inventory of backed up data sets, UNIX files and volumes, dumped volumes, and backed up aggregates
- OCDS, which contains a tape table of contents (TTOC) inventory of migration and backup tape volumes

In normal operation, these records stay in synchronization. However, because of data errors, hardware failures, or human errors, it is possible for these records to become unsynchronized. The AUDIT command allows the system to cross-check the various records concerning data sets and DFSMSHsm resources. AUDIT can list errors and propose diagnostic actions or, at your option, complete most repairs itself.

Consider using the AUDIT command for the following reasons:

- After any CDS restore (highly recommended)
- Periodic checks
- After an ARC184I message (error when reading/writing DFSMSHsm CDS records)
- Errors on the RECALL or DELETE of migrated data sets
- Errors on BDELETE or RECOVER of backup data sets and UNIX files
- DFSMSHsm tape-selection problems
- RACF messages (mismatches)
- Power or hardware failure
- Check the consistency of fast replication CDS record relationships

You can use AUDIT to cross-check the following sources of control information:

- MCDS or individual migration data set records
- BCDS or individual backup data set records or UNIX files records or ABARS records or fast replication records
- OCDS or individual DFSMSHsm-owned tapes
- DFSMSHsm-owned DASD volumes

- Migration-volume records
- Backup-volume records
- Recoverable-volume records (from dump or incremental backup)
- Contents of SDSP data sets
- Fast Replication dump, volume pair, version, and volume copy pool association records
- DFSMSHsm-owned cloud containers

Auditing common queues

Common queues have interrelated entries that may become corrupted due to abends and unexpected losses of connectivity. A corrupted common queue may cause certain requests to not be processed. DFSMSHsm automatically corrects some inconsistencies, but for others, it is necessary to issue the AUDIT COMMONQUEUE command. The AUDIT command enables DFSMSHsm to dynamically correct inconsistencies with minimal impact on processing.

Consider using AUDIT COMMONQUEUE in the following situations:

- After receiving an ARC1506E message
- After receiving an ARC1187E message
- When recall requests are unexpectedly not being selected for processing

The structure of the common queues is not externalized. Message ARC1544I is the only output that AUDIT COMMONQUEUE returns. It does not return a specific message for each error because individual error messages are of no value. The OUTDATASET, SYSOUT, and TERMINAL parameters are not used with AUDIT COMMONQUEUE.

Unlike other AUDIT functions, auditing a common queue is not time intensive and may be performed at any time.

Using the AUDIT command

Before using the AUDIT command, read the descriptions of the available options in Chapter 75, “Using the AUDIT command,” on page 1255. Appendix A discusses which conditions are detected by each option.

It is best to use AUDIT at times of low system activity, as some audit processes can run for quite some time. However, the AUDIT process can be used at any time.

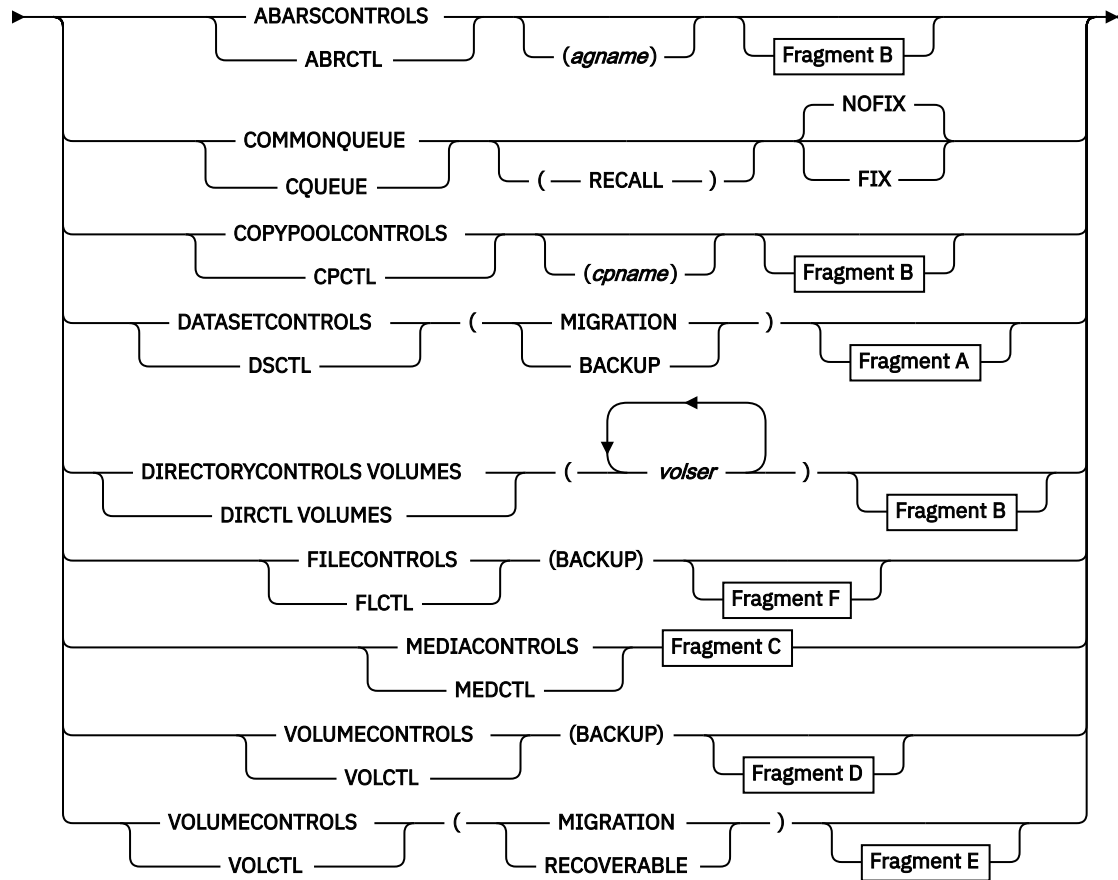
SMS-managed *volumes* cannot be audited with the AUDIT command and the process for auditing SMS-managed *data sets* is similar to that for auditing non-SMS-managed data sets.

Note: If AUDIT is executing and backup of the DFSMSHsm control data sets is started (via BACKVOL CDS command or AUTOBACKUP), all DFSMSHsm functions on the host that started this backup are halted until the AUDIT function and the backup of the DFSMSHsm CDS's have completed.

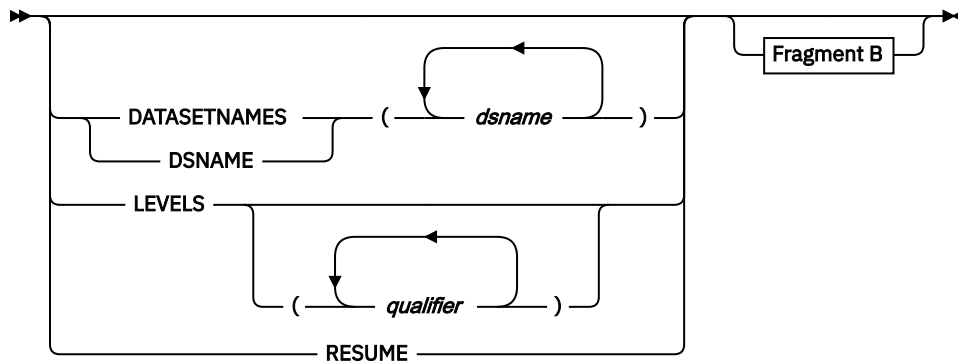
Syntax of the enhanced AUDIT commands

When you issue the AUDIT command, you must specify one of the required parameters, to indicate a primary category of control information, which will “drive” that invocation of AUDIT. The optional parameters qualify what to audit, the output destination, and other qualifying information.

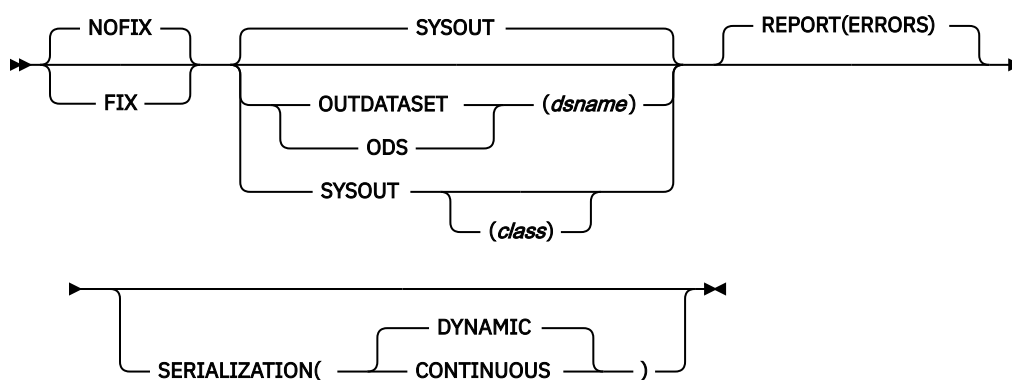
➤ AUDIT ➤



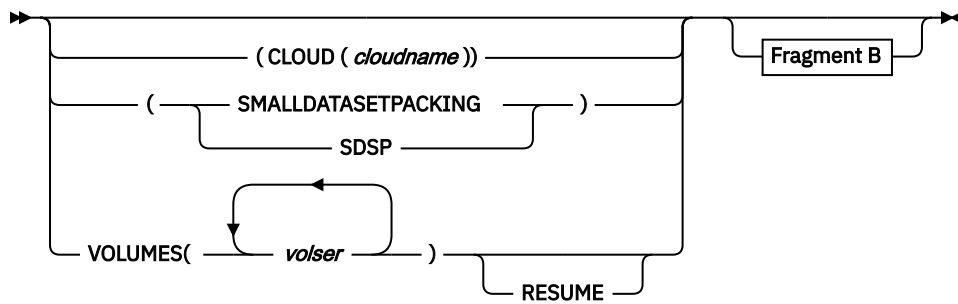
A: DATASETCONTROLS optional parameters



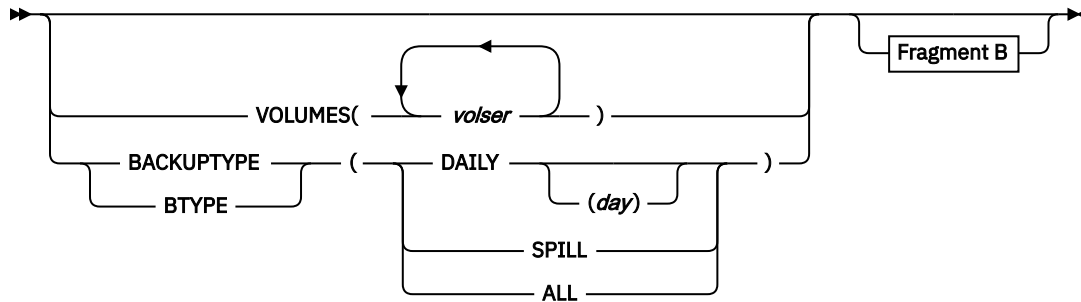
B: Additional optional parameters



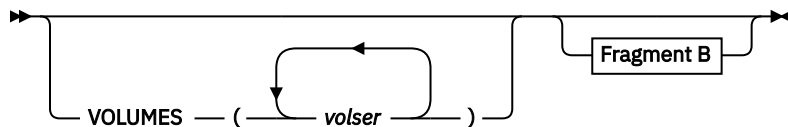
C: MEDIACONTROLS parameters



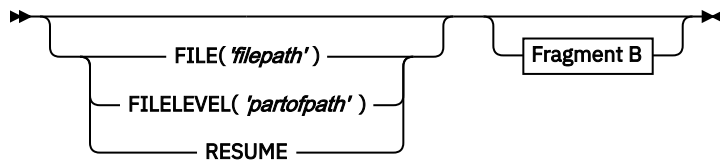
D: VOLUMECONTROLS (BACKUP) optional parameters



E: VOLUMECONTROLS (MIGRATION and RECOVERABLE) optional parameters



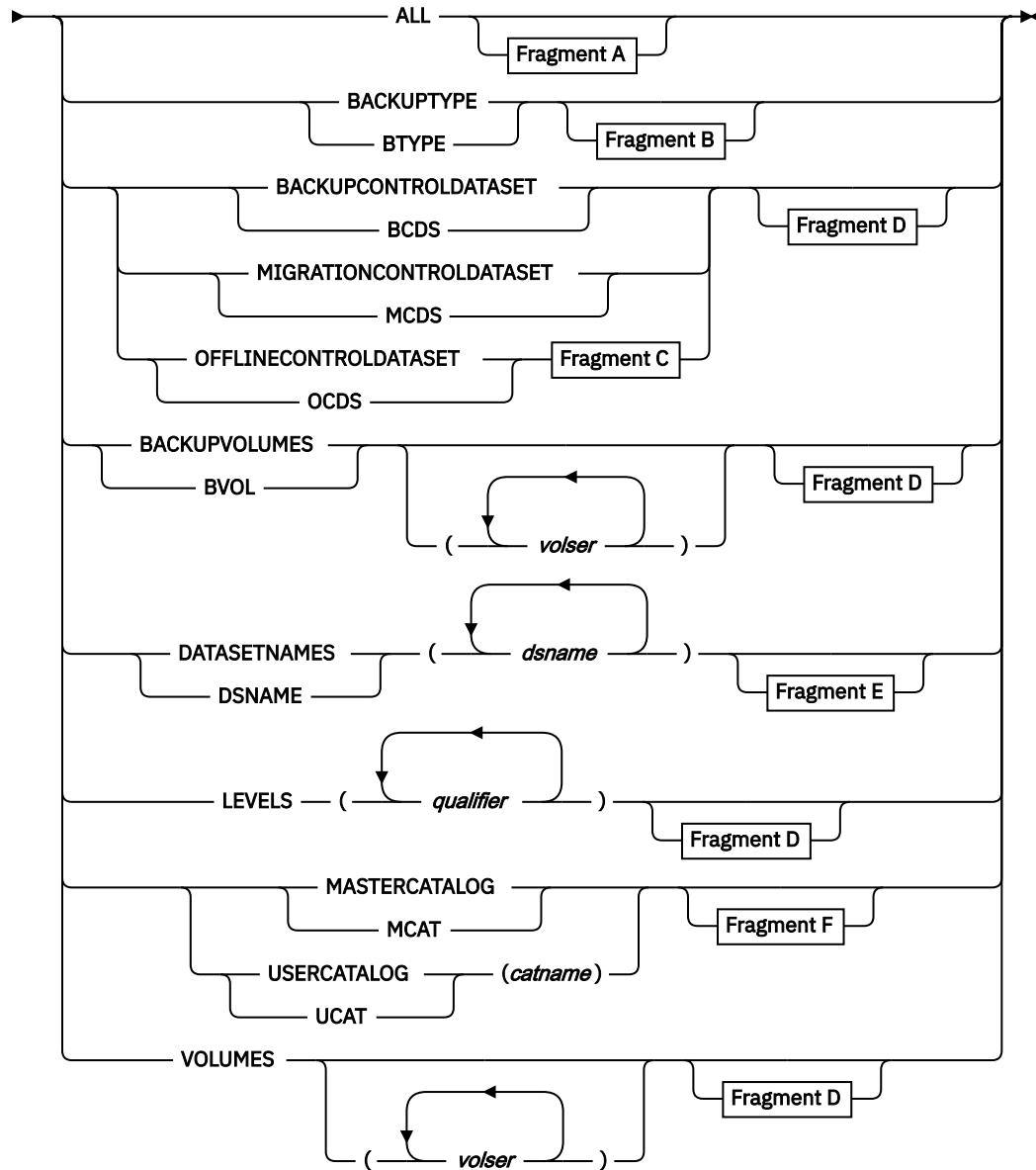
F: FILECONTROLS optional parameters



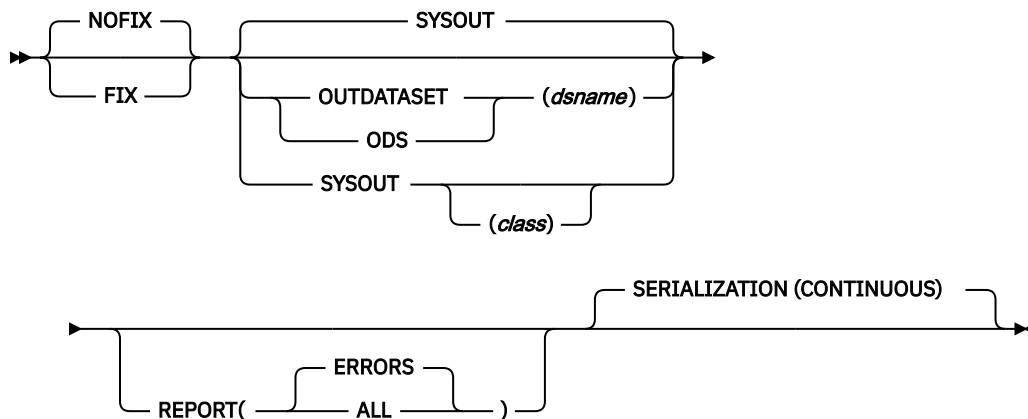
Syntax of the original AUDIT commands

The following options provide less comprehensive auditing, and somewhat less flexible output, than the previous options, but are available for compatibility with CLISTs or job streams developed for use with older versions of DFSMSHsm.

➡ AUDIT ➡

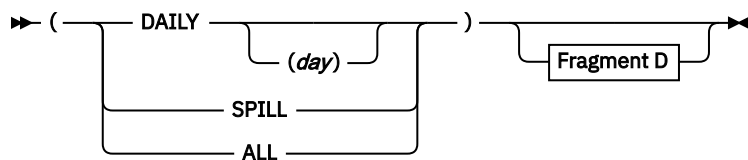


A: ALL optional parameters

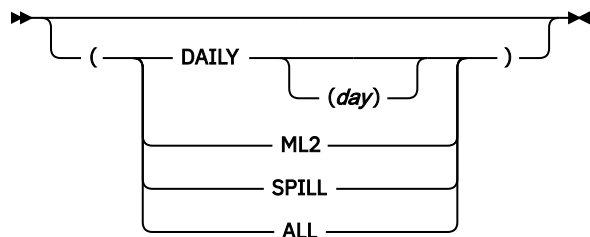


B: BACKUPTYPE parameters

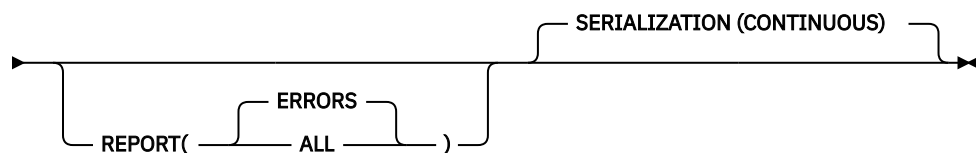
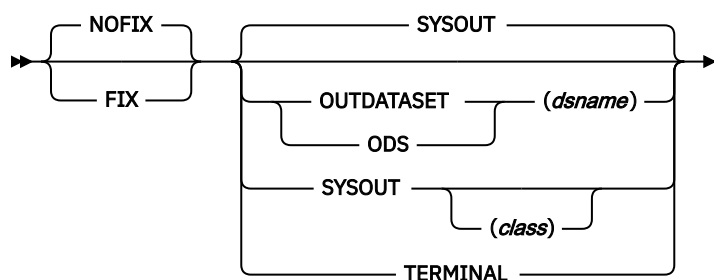
AUDIT command for DFSMSHsm



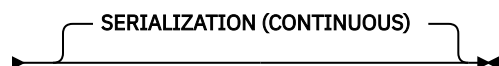
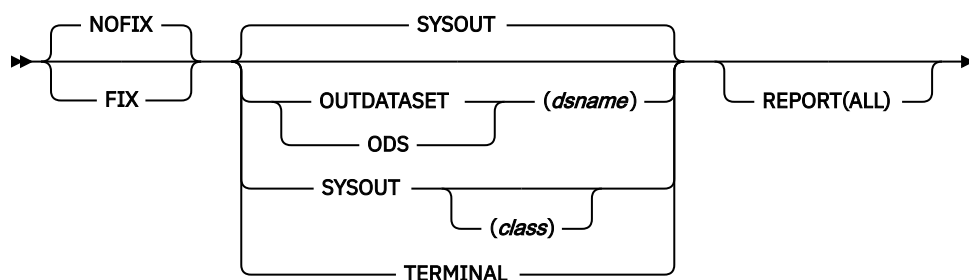
C: OFFLINECONTROLDATASET parameters



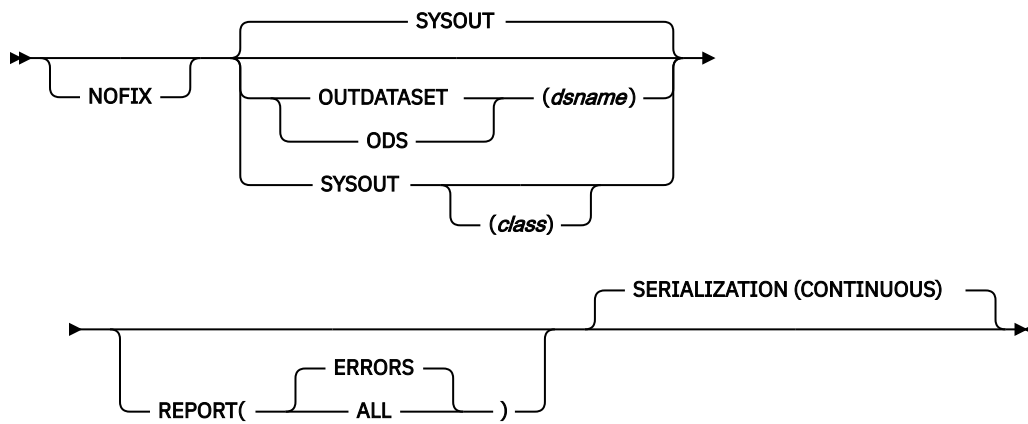
D: Additional optional parameters



E: DATASETNAMES optional parameters



F: MASTERCATALOG and USERCATALOG optional parameters



Required and optional parameters of the AUDIT command

This topic describes the required parameters of the AUDIT command. Some of the parameters listed in this topic are used as both required and optional parameters in the syntax diagrams. They are described in this topic and identified as such.

ABARSCONTROLS: Requesting an audit of ABR records in the BCDS

Explanation: **ABARSCONTROLS** is specified to audit the ABR records in the BCDS.

ABARSCONTROLS(*agname*) is an optional parameter that allows you to perform an audit of all ABR records matching the specified aggregate-group name.

IF AUDIT ABARSCONTROLS is issued and no ABR records are found, or if AUDIT ABARSCONTROLS(*agname*) is issued and no ABR records matching the *agname* specified are found, an error message is issued.

Defaults: None.

ALL, BACKUPCONTROLDATASET, MIGRATIONCONTROLDATASET, and OFFLINECONTROLDATASET: Requesting an audit of data sets

Explanation: **ALL**, **BACKUPCONTROLDATASET**, **MIGRATIONCONTROLDATASET**, and **OFFLINECONTROLDATASET** are mutually exclusive parameters that you use to audit sources of data set names, such as all control data sets or a specific control data set. Each of these represents a source of data set names.

ALL is specified to audit the migration control data set, the backup control data set, and the offline control data set. **ALL** is not used to audit the common queues.

BACKUPCONTROLDATASET is specified to audit the DFSMSHsm backup control data set.

MIGRATIONCONTROLDATASET is specified to audit the DFSMSHsm migration control data set.

OFFLINECONTROLDATASET is specified to audit the DFSMSHsm offline control data set. The TTOC records are the source of information to determine which data sets are audited. **DAILY** | **ML2** | **SPILL** | **ALL** are mutually exclusive, optional subparameters of the **OFFLINECONTROLDATASET** parameter used to specify which tape volumes to audit.

Subparameter	Explanation
DAILY	<p>OFFLINECONTROLDATASET(DAILY) is specified to audit tape daily backup volumes. For DAILY(day), substitute a decimal number from 1 to 31 to represent the day in the backup cycle. If you do not specify a particular day, the audit output occurs in the following order:</p> <ul style="list-style-type: none"> • All volumes assigned to day 1 • All volumes assigned to day 2 • And so forth, until all tape daily backup volumes have been audited <p>For each day, the order of the audit output is in ascending volume serial number sequence.</p>
ML2	<p>OFFLINECONTROLDATASET(ML2) is specified to audit all tape migration level 2 volumes. The order of audit output occurs in ascending volume serial number sequence for migration level 2 volumes.</p>
SPILL	<p>OFFLINECONTROLDATASET(SPILL) is specified to audit all tape spill backup volumes. The order of audit output occurs in ascending volume serial number sequence for spill volumes.</p>
ALL	<p>OFFLINECONTROLDATASET(ALL) is specified to audit all tape volumes. The audit output occurs in the following order:</p> <ul style="list-style-type: none"> • All daily backup volumes. The daily backup volumes are listed for each day, and the volumes assigned to each day are listed in ascending volume serial number sequence. • All spill backup volumes. The spill backup volumes are listed in ascending volume serial number sequence. • All unassigned backup volumes. The unassigned backup volumes are listed in ascending volume serial number sequence. • All migration level 2 volumes. The migration level 2 volumes are listed in ascending volume serial number sequence.

Note:

1. When you request an audit of the MCDS or BCDS, DFSMSHsm uses the VTOC of the primary volumes to ensure that the volume specified in the catalog contains the data set referred to. Therefore, those primary volumes (including those managed by SMS) must be mounted.
2. If you specify the OFFLINECONTROLDATASET parameter without a subparameter, the AUDIT command fails.
3. Dump volumes are not audited.

BACKUPTYPE, BACKUPVOLUMES, and VOLUMES: Requesting an audit of primary, migration, or backup volumes

The VOLUMES parameter can be used either as a primary option or as an optional parameter. Therefore, it is listed both ways in the syntax tables at the beginning of this topic.

Explanation: BACKUPTYPE, BACKUPVOLUMES, and VOLUMES are mutually exclusive parameters that you use to audit data set names; such as, one or more backup volumes, non-SMS-managed primary volumes, or migration volumes. Each of these represents a source of data set names.

BACKUPTYPE is specified to audit a type of backup volume. **DAILY | SPILL | ALL** are mutually exclusive, optional subparameters of the BACKUPTYPE parameter, specifying the type of backup volume DFSMSHsm is to audit.

Subparameter	Explanation
DAILY	<p>BACKUPTYPE(DAILY) is specified to audit daily backup volumes. For DAILY(day), substitute a decimal number from 1 to 31 to represent the day in the backup cycle. If <i>day</i> is specified, DFSMSHsm audits all backup volumes assigned to the specified day in the backup cycle. The order of the audit output is by entries in the daily backup cycle volume record (BVR) for the specified day. If you do not specify a particular day, the audit output occurs in the following order:</p> <ul style="list-style-type: none"> • All volumes assigned to day 1 • All volumes assigned to day 2 • And so forth, until all tape daily backup volumes have been audited <p>For each day, the order of the audit output is by entries in the backup cycle volume record (BVR).</p>
SPILL	<p>BACKUPTYPE(SPILL) is specified to audit all spill backup volumes. The order of the audit output is by entries in the spill backup cycle volume record (BVR).</p>
ALL	<p>BACKUPTYPE(ALL) is specified to audit all daily and spill backup volumes. The order of the audit output is all daily backup volumes and then all spill backup volumes.</p>

BACKUPVOLUMES and **BACKUPVOLUMES(volser...)** is specified to audit specific backup volumes or all backup volumes. For *(volser...)*, substitute the volume serial numbers of the backup volumes you want DFSMSHsm to audit. If you specify particular volumes, the order of the audit output is the same as the order of the volumes you specified. If you do not specify a particular volume, the order of the audit output occurs in ascending volume serial number order.

VOLUMES and **VOLUMES(volser...)** is specified to audit one or more non-SMS-managed volumes. For *(volser...)*, substitute a volume serial number or list of volume serial numbers for volumes that you want in the audit. SMS-managed volumes are not included in the audit. If a volume serial number for an SMS-managed volume is specified, DFSMSHsm does not audit that volume. **VOLUMES(volser...)** can be used with the **VOLUMECONTROLS** primary option and is required with the **DIRECTORYCONTROLS** and **MEDIACONTROLS** primary options.

Defaults: If you specify **VOLUMES** without *(volser...)*, DFSMSHsm audits all non-SMS-managed primary volumes, migration level 1 volumes, and all migration level 2 DASD volumes.

If you specify **BACKUPVOLUMES** without *(volser...)*, DFSMSHsm audits all backup volumes with a volume record in the BCDS.

There is no default for **BACKUPTYPE**. If you do not specify a subparameter with the **BACKUPTYPE** parameter, the **AUDIT** command fails.

Note: If the **VOLUMES** parameter specifies a migration level 1 volume, be aware of the following items:

- The **MEDIACONTROLS (SDSP)** primary option must be used to audit the control information in a small-data-set-packing data set.
- When you use the **BACKDS** command to back up a data set, the backup version temporarily resides on a migration level 1 volume. DFSMSHsm audits these data sets.

COMMONQUEUE: Requesting an audit of common queues

Explanation: **COMMONQUEUE** is a parameter you specify to audit all of the DFSMSHsm common queues. The data in a common queue can be corrupted by abends and unexpected losses of connectivity. A corrupted common queue may cause certain requests to not be processed. Specify **COMMONQUEUE** to enable DFSMSHsm to dynamically correct inconsistencies with minimal impact on processing.

Specify **COMMONQUEUE(RECALL)** to audit just the common recall queue.

If possible, issue AUDIT COMMONQUEUE(RECALL) when all the typical hosts are connected to the common queue.

Defaults: None.

COPYPOOLCONTROLS: Requesting an audit of fast replication CDS record relationships

Explanation: **COPYPOOLCONTROLS** is a parameter you specify to confirm that the fast replication CDS record relationships are correct for the copy pools that are managed by DFSMSHsm. The AUDIT function identifies any discrepancies that are discovered.

COPYPOOLCONTROLS

audits all copy pool records.

COPYPOOLCONTROLS(*cpname*)

audits the copy pool records for a specific copy pool.

Defaults: None.

DATASETCONTROLS: Requesting an audit of data set control records

Explanation: DATASETCONTROLS audits either the records in the MCDS or the records in the BCDS. It checks all (or specified) records for synchronization with other CDS records.

Subparameter	Explanation
MIGRATION	DATASETCONTROLS (MIGRATION) specifies that the data set records in the MCDS are cross-checked and a list of errors is produced.
BACKUP	DATASETCONTROLS (BACKUP) specifies that the data set records in the BCDS are cross-checked and a list of errors is produced.

Along with this parameter, other useful optional parameters are DATASETNAMES(*dsname...*), LEVELS(*qualifier...*), or RESUME.

Defaults

- When using the DATASETCONTROLS command, you must specify either MIGRATION or BACKUP. If you do not specify one or the other, you will get an error message.
- If you specify AUDIT DATASETCONTROLS(MIGRATION) without optional parameters, you will get an audit of all data set records in the MCDS.
- If you specify AUDIT DATASETCONTROLS(BACKUP) without optional parameters, you will get an audit of all data set records in the BCDS.

DATASETNAMES, LEVELS, and RESUME: Requesting an audit of data sets

The DATASETNAMES and LEVELS parameter can be used either as a primary option or as an optional parameter. Therefore, each is listed both ways in the syntax tables at the beginning of this topic.

Explanation: **DATASETNAMES(*dsname...*)** | **LEVELS(*qualifier...*)** | **RESUME** are mutually exclusive parameters to audit a data set, a list of data sets, a data set group, or a list of data set groups.

DATASETNAMES(*dsname...*) is specified to audit a data set or list of data sets. For (*dsname...*) substitute the fully qualified name or a list of fully qualified names for the data sets you want DFSMSHsm to audit. You can specify a data set name of up to 44 characters.

LEVELS(*qualifier...*) is specified to audit a group or list of groups of data sets with the same set of initial characters in the data set name. If you specify this parameter, DFSMSHsm audits all MCDS records for data sets that begin with the supplied qualifiers. If a qualifying data set found in the MCDS also has a BCDS entry, the BCDS entry will also be audited. For (*qualifier...*), substitute the set of initial characters in

the data set name or list of the set of initial characters for the data sets you want DFSMSHsm to audit. You can specify a qualifier of up to 44 characters.

RESUME applies only to the DATASETCONTROLS(MIGRATION) and DATASETCONTROLS(BACKUP) parameters of the AUDIT command. The RESUME function enables you to continue a previous audit of DATASETCONTROLS that for some reason has not been completed.

AUDIT processing does not run to completion when an operator enters a HOLD command. AUDIT processing might not complete if DFSMSHsm ends abnormally. If you entered the original command without the DATASETNAMES or LEVELS parameters, you can re-enter the command, using the RESUME parameter, and the audit will resume.

Note: The HOLD command informs DFSMSHsm to stop the audit as soon as AUDIT finishes processing the current data set. All AUDIT commands that have not been processed remain in the queue. The RELEASE command allows the execution of AUDIT commands that are in the queue, but will not cause an interrupted command to be restarted; thus you must issue another AUDIT command that includes the RESUME parameter.

Defaults: None.

Note:

1. When you specify the DATASETNAMES parameter, DFSMSHsm prints all audit information for each audited data set, whether or not errors are found for that data set.
2. When you specify the LEVELS parameter, DFSMSHsm prints MCDS and BCDS entries for each group.
3. DFSMSHsm does not handle partitioned data set members individually. If you specify a partitioned data set with a member name, DFSMSHsm audits the entire partitioned data set.

DIRECTORYCONTROLS VOLUMES: Comparing a VTOC or a TTOC to CDS records

Explanation: **DIRECTORYCONTROLS VOLUMES(volser...)** is specified to compare information in the DASD VTOC or the TTOC with that of the control records in the MCDS or BCDS.

VOLUMES(volser...) is a required parameter. For (volser...), substitute a volume serial number or a list of volume serial numbers. Each serial number must identify a DFSMSHsm-owned tape backup or migration volume or a DFSMSHsm-owned DASD migration volume.

Defaults: None.

FILECONTROLS: Requesting an audit of UNIX file control records

FILECONTROLS audits the records in the BCDS. It checks all (or specified) records for synchronization with other CDS records.

Subparameter	Explanation
BACKUP	FILECONTROLS (BACKUP) specifies that the UNIX file records in the BCDS are cross-checked and a list of errors is produced.

Along with this parameter, other useful optional parameters are FILE(filepath...), FILELEVELS(partofpath...), or RESUME.

Defaults

- When using the FILECONTROLS command, you must specify BACKUP. If it is not specified, you will get an error message.
- If you specify AUDIT FILECONTROLS(BACKUP) without optional parameters, you will get an audit of all UNIX file records in the BCDS.

FILE, FILELEVEL, and RESUME: Requesting an audit of UNIX files

The FILE and FILELEVEL parameter can be used either as a primary option or as an optional parameter. Therefore, each is listed both ways in the syntax tables at the beginning of this topic

FILE(filepath...) | **FILELEVEL(partofpath...)** | **RESUME** are mutually exclusive parameters to audit a UNIX file.

FILE(filepath...) is specified to audit a UNIX file. For (filepath...) substitute a single absolute UNIX file path name that you want DFSMSHsm to audit. The UNIX file path name must be enclosed in single quotation marks.

FILELEVEL(partofpath...) is specified to audit a group of files that begin with the specified partofpath partial path. Only a single absolute UNIX path to a directory may be specified. partofpath name must be enclosed in single quotation marks.

RESUME applies only to the FILECONTROLS(BACKUP) parameter of the AUDIT command. The RESUME function enables you to continue a previous audit of FILECONTROLS that for some reason has not been completed. AUDIT processing does not run to completion when an operator enters a HOLD command.

AUDIT processing might not complete if DFSMSHsm ends abnormally. If you entered the original command without the FILE or FILELEVEL parameters, you can reenter the command, using the RESUME parameter, and the audit will resume.

Defaults: None.

Note:

1. When you specify the FILE parameter, DFSMSHsm prints all audit information for each UNIX file, regardless if errors are found for that UNIX file.
2. When you specify the FILELEVEL parameter, DFSMSHsm prints BCDS entries for each directory.

MASTERCATALOG and USERCATALOG: Requesting an audit of a catalog

Explanation: **MASTERCATALOG** and **USERCATALOG** are mutually exclusive parameters that you use to audit data set entries in a catalog, which is a source of data set names.

MASTERCATALOG is specified to audit the master catalog.

USERCATALOG(catname) is specified to audit a particular user catalog. For (catname), substitute the name of the user catalog you want DFSMSHsm to audit.

Defaults: None.

Note: You cannot specify the TERMINAL parameter with the MASTERCATALOG or USERCATALOG parameters.

MEDIACONTROLS VOLUMES, MEDIACONTROLS(SDSP) VOLUMES and MEDIACONTROLS(CLOUD): Requesting an audit of control information

Explanation: **MEDIACONTROLS VOLUMES** and **MEDIACONTROLS(SDSP) VOLUMES(volser...)** are mutually exclusive parameters that are specified to audit control records as follows:

MEDIACONTROLS audits control information contained in migration copies and backup versions.

MEDIACONTROLS(SMALLDATASETPACKING) audits only the SDSP (small data set packing) data sets on the specified volumes.

In either case, **VOLUMES(volser...)** is a required parameter. For (volser...) substitute a volume serial number or a list of volume serial numbers. Each serial number must identify a DFSMSHsm-owned, single-file format, tape backup or migration volume or a DASD migration volume.

MEDIACONTROLS(CLOUD(cloud_network_connection_name)) audits control information in migration copies and full volume dump copies in the cloud. The cloud_network_connection_name refers to a defined cloud_network_connection_name. MEDIACONTROLS(CLOUD(cloud_network_connection_name))

examines containers created using the current HSMplex name. The CLOUD parameter is mutually exclusive with the VOLUMES and SMALLDATASETPACKING parameters. The RESUME parameter does not apply.

Defaults: None.

Note:

1. If DFSMSHsm is unable to complete output processing to backup or migration tapes, an incomplete TTOC is created. When an audit is performed against these tapes, the AUDIT function positions to the last data set described by that portion of the TTOC that was written successfully and then proceeds to audit the tape from that point forward.

RESUME can be specified on the AUDIT MEDCTL VOLUMES(tape volser) FIX Command in order to resume a previously interrupted audit in the case where the AUDIT was held, DFSMSHsm was stopped, or SETSYS EMERGENCY was specified.

RESUME is only valid for migration and backup tapes. It is not valid when specified with SDSP or when the volser specified is for a DISK volume.

RESUME is only valid when FIX is also specified. If the initial AUDIT is issued without specifying FIX and the AUDIT is held, and then the generated FIXCDS commands are manually issued, the results of AUDIT MEDCTL with RESUME are unpredictable. Manually entering patches generated by NOFIX is discouraged by IBM.

AUDIT will not be able to RESUME after abends or experiencing an I/O error or a point or note failure. If you CANCEL the AUDIT TASK using the CANCEL TCADDRESS Command, be sure to first HOLD AUDIT in order for RESUME to execute.

If RESUME is validly specified, but the required information and records are not available, AUDIT will start at the beginning of the tape.

If RESUME is specified on the command, message ARC0877I will be issued stating the volser and extension on which AUDIT is starting.

Note: When audit is held or stopped, the function terminates. After audit terminates and before the AUDIT MEDCTL RESUME command is issued, the volume is available for all functions. Following are some situations that will affect the resumed audit:

- If the volume is RECYCLED and then DELVOLED, the TTOC is deleted and the audit cannot be resumed.
- If RECALLS or RECONNECTIONS occur for data sets already audited, the valid block count will not reflect this when the resumed audit completes.
- If additional entries are added to the volume by migration or backup, the audit will resume at the interrupt point and continue on through the new entries as usual.
- However, if the additional entries result in a FAILEDCREATE situation where the TTOC is not consistent with the contents of the tape media and TTCFAIST is ON, the RESUME parameter will be ignored and audit will add the missing TTOC entries. On completion of this audit, the AUDIT MEDCTL RESUME command can be issued again and audit will then resume with the last data set being processed when audit was held.

VOLUMECONTROLS (BACKUP, MIGRATION, and RECOVERABLE): Requesting an audit of control information

Explanation: **VOLUMECONTROLS (BACKUP, MIGRATION, and RECOVERABLE)** are mutually exclusive parameters that are specified to audit control records as follows:

Subparameter	Explanation
BACKUP	BCDS control records for backup volumes are audited.
MIGRATION	MCDS control records for migration volumes are audited.

Subparameter	Explanation
RECOVERABLE	BCDS control records that concern recoverable volumes (volumes which have been backed up, dumped, or both) are audited.

Note: The optional parameter VOLUMES(*volser...*) can be used to specify a volume serial number or a list of volume serial numbers for any of the VOLUMECONTROLS options listed above.

Defaults: None.

Optional parameters of the AUDIT command

This topic describes the optional parameters of the AUDIT command.

FIX and NOFIX: Specifying whether auditing repairs errors

Explanation

FIX and NOFIX are mutually exclusive, optional parameters specifying whether auditing repairs any error it can when an error is found.

FIX

Auditing repairs any error it can.

NOFIX

Auditing reports errors, but does not fix them. For COMMONQUEUE, the number of errors only is reported.

Tip: The output data set contains executable FIXCDS commands to create or correct CDS records, but those CDS changes are only a subset of the corrective actions taken by audit processing. In addition, the CDS records can be read multiple times during the course of an audit. If the NOFIX option is specified, the changes to a record generated on its initial read are not reflected in that record when it is read a subsequent time, this might result in incorrect action being taken. For these reasons, you should not later submit output from an AUDIT NOFIX command as a substitute for an AUDIT FIX command.

Defaults

The default is NOFIX.

Usage notes:

1. The number of errors reported by AUDIT COMMONQUEUE FIX may be lower than the number of errors reported by a previous AUDIT COMMONQUEUE NOFIX. This is acceptable since the errors in the common queues are interrelated. Hence, fixing one error may actually fix several other errors.
2. If you specify the FIX parameter and the parameter applies, DFSMSHsm issues a message indicating whether the fix was successful or unsuccessful.
3. The FIX parameter can degrade performance. When you specify it with the MASTERCATALOG or USERCATALOG parameter, DFSMSHsm does not fix the audited information. If you specified the FIX parameter with the MASTERCATALOG or USERCATALOG parameter, DFSMSHsm allows the operator to cancel an audit by replying N to the following message:

```
ARC0803A  WARNING:  AUDIT OF CATALOG MAY DEGRADE PERFORMANCE,
           REPLY 'Y' TO START AUDIT OR 'N' TO CANCEL
           AUDIT COMMAND
```

4. You must use the FIXCDS command to correct any discrepancies between the catalog and the MCDS.
5. If you specify the FIX parameter when you audit a primary or migration volume, DFSMSHsm scratches and uncatalogs all utility data sets on the volume even if the utility data sets have expiration dates.

6. If you use the CDSR=YES startup parameter, DFSMSHsm issues the RESERVE macro to keep the other z/OS images from accessing the three volumes containing the MCDS, BCDS, and OCDS under the following conditions:
 - You are in a multiple-image environment.
 - You have specified the FIX command.
 - You have specified SERIALIZATION(CONTINUOUS).
 - When the primary audit category is one of the following:
 - ALL | BACKUPCONTROLDATASET | MIGRATIONCONTROLDATASET | OFFLINECONTROLDATASET(DAILY(*day*) | ML2 | SPILL | ALL)
 - MASTERCATALOG | USERCATALOG(*catname*)
 - BACKUPTYPE(DAILY(*day*) | SPILL | ALL) | BACKUPVOLUMES | BACKUPVOLUMES(*volser...*) | VOLUMES | VOLUMES(*volser...*)
 - DATASETNAMES(*dsname...*) | LEVELS(*qualifier...*)
7. The reserve applies to the three volumes that contain the control data sets; therefore, no other data sets on those volumes can be accessed from another z/OS image.
8. If you specify the FIX parameter when you audit a volume, DFSMSHsm catalogs the uncataloged data sets (excluding rolled-off GDG data sets) on the volume. Consequently, AUDIT should not be run concurrently with other jobs that are creating uncataloged data sets.

OUTDATASET, SYSOUT, and TERMINAL: Specifying the output location for the report

Explanation: **OUTDATASET(dsname)**, **SYSOUT**, and **TERMINAL** are mutually exclusive, optional parameters that specify the output location for the report you are requesting.

OUTDATASET(dsname) specifies the name of the data set where DFSMSHsm is to write the output data. For (*dsname*), substitute the fully qualified name of the data set to receive the audit report. These parameters are not applicable with AUDIT COMMONQUEUE.

If the data set does not exist, DFSMSHsm dynamically allocates and catalogs an output data set with the following characteristics:

- Data set name specified (*dsname*)
- Record format of fixed-blocked with ANSI control characters (FBA)
- Logical record length of 121
- Data set is system reblockable
- Primary allocation of 20 tracks (see second note)
- Secondary allocation of 50 tracks (see second note)
- Unit of SYSALLDA (see second note)

If the data set already exists, DFSMSHsm will use the data set. The data set must have the following characteristics:

- The data set must be cataloged and on DASD.
- The data set record format must be FBA, and the logical record length must be 121.
- The data set is system reblockable.
- The user can choose the primary space allocation.
- If the data set does not contain data, DFSMSHsm starts writing output data at the beginning of the data set.
- If the data set contains data, DFSMSHsm writes the output data after the existing data.

SYSOUT(class) specifies that the report is to go to the specified output class. For (*class*), substitute one alphanumeric character for the class you want.

TERMINAL specifies that the report is to be printed at the same terminal where the command was issued, and is to be printed to SYSOUT using only the output class specified with the SYSOUT parameter of the SETSYS command or the DFSMSHsm default for SYSOUT.

Defaults: The default is SYSOUT, and the default for (*class*) is the value specified with the SETSYS command. If you do not use the SYSOUT parameter of the SETSYS command to specify the output location, the default is class A.

Note:

1. DFSMSHsm does not handle partitioned data set members individually. If you specify a partitioned data set with a member name as the output data set, the audit report could be written over existing data.
2. If you select the OUTDATASET option, you can use the PATCH command to change the unit name, primary allocation, and secondary allocation. If you select the SYSOUT option, you can use the PATCH command to change whether, and how, DFSMSHsm limits the lines of SYSOUT output. Refer to the [z/OS DFSMSHsm Implementation and Customization Guide](#) for the PATCH command.

REPORT: Specifying how much audit information to print

Explanation: **REPORT** is an optional parameter specifying how much audit information you want printed. These parameters are not applicable with AUDIT COMMONQUEUE nor AUDIT COPYPOOLCONTROLS.

Subparameter	Explanation
ALL	REPORT(ALL) specifies that all audit information, including the error-condition information for the specified data sets, volumes, levels, catalogs, or control data sets is printed.
ERRORS	REPORT(ERRORS) specifies that only the error-condition information for the specified volumes, catalogs, or control data sets is printed.

Defaults: The default is ERRORS, except when using the DATASETNAMES parameter. For DATASETNAMES, the ALL parameter always applies.

Note: The ERRORS subparameter of REPORT does not apply for DATASETNAMES. If you specify ERRORS when it does not apply, DFSMSHsm ignores it and uses the ALL parameter.

SERIALIZATION: Stabilizing control information

Explanation: **SERIALIZATION** is an optional parameter stabilizing the control information. DFSMSHsm control information changes dynamically as DFSMSHsm operates. For AUDIT to detect an out-of-synchronization condition, diagnose the problem, and apply a repair, AUDIT must stabilize (serialize) access to control data sets and records to maintain a stable environment. These parameters are not applicable with AUDIT COMMONQUEUE.

Subparameter	Explanation
DYNAMIC	SERIALIZATION(DYNAMIC) specifies that an enqueue is placed on the CDSs only when the audit function finds a discrepancy in a record relationship. This enqueue is released periodically to allow other functions access to the CDSs. SERIALIZATION(DYNAMIC) is the option of choice; however, if you need to complete an audit function in the shortest time possible, SERIALIZATION(CONTINUOUS) is the better performer.
CONTINUOUS	SERIALIZATION(CONTINUOUS) specifies that an enqueue is placed on the CDSs for the entire audit. Use SERIALIZATION(CONTINUOUS) when it is important for the DFSMSHsm AUDIT command to complete in the shortest period of time, such as just after a CDS RESTORE. Be careful if you select the SERIALIZATION(CONTINUOUS) option, as no other DFSMSHsm functions can process until the audit function is complete.

Note: Continuous is functional only when DFSMSHsm is processing in a multihost environment and the FIX parameter is specified on the AUDIT command.

Defaults: The default for the Enhanced Audit commands is SERIALIZATION(DYNAMIC) and the default for the Original Audit commands is SERIALIZATION(CONTINUOUS).

Using the AUDIT command

For more discussion of the AUDIT function, see [Chapter 75, “Using the AUDIT command,” on page 1255](#).

Chapter 31. AUTH command: Identifying authorized users

There are two methods used to authorize user commands. DFSMSHsm allows an installation to control the authorization of its commands through the use of either RACF FACILITY class profiles or the AUTH command.

If the RACF FACILITY class is active when DFSMSHsm starts, the following processing occurs:

- DFSMSHsm uses RACF FACILITY class checking for all authorized and user commands
- DFSMSHsm honors profiles in the FACILITY class that are added or modified

See "Authorizing and Protecting DFSMSHsm Commands in a FACILITY Class Environment" in the [z/OS DFSMSHsm Implementation and Customization Guide](#) for information on setting up the RACF environment.

If the RACF FACILITY class is not active when DFSMSHsm starts, DFSMSHsm uses the AUTH command to determine authorized DFSMSHsm users and commands.

The AUTH command identifies both the authorized user who can only issue authorized DFSMSHsm commands and the authorized user who cannot only issue authorized DFSMSHsm commands but can also add, delete, and change the authority of other DFSMSHsm users. When DFSMSHsm is installed, the storage administrator with responsibility for DFSMSHsm should be identified as the authorized user who can affect the authority of other DFSMSHsm users.

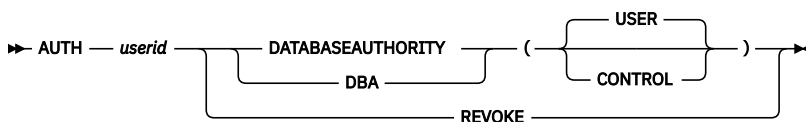
The AUTH command can be submitted only by users who are already authorized users having the database authority control attribute, or the command must be part of the PARMLIB member being processed during DFSMSHsm startup.

You need not repeat the AUTH command each time you start DFSMSHsm. If you do not specify any parameters with the AUTH command, DFSMSHsm accepts the command but does not add or remove any names from the list of authorized users.

Note:

1. Use the AUTH command carefully because anyone who is an authorized user can issue DFSMSHsm user commands without having RACF check the security of the command issuer.
2. The AUTH command cannot be issued from the system console.
3. If you issue the AUTH command when the RACF FACILITY class is active, it completes with a return code of four. The authorized user record (MCU) is still created or updated, but access to the DFSMSHsm storage administrator commands is still controlled by the RACF FACILITY class. Maintaining accurate MCU records is helpful when the RACF FACILITY class is not active or when using StorWatch DFSMSHsm Monitor.

Syntax of the AUTH command



Required parameters of the AUTH command

This topic describes the required parameters of the AUTH command.

userid: Specifying the user whose authorization is to be changed

Explanation: *userid* is a required positional parameter that specifies the identification of the user whose authority to issue any authorized command is to be changed. Authorized user commands are the storage administrator, operator, and system programmer commands.

For *userid*, substitute a string of 1-8 alphanumeric characters for the identification of the user whose authorization is to be changed. The first character of the character string must be alphabetic or #, \$, or @, and the remainder of the string can be alphanumeric and #, \$, or @.

Defaults: None.

Note:

1. Because *userid* is a required positional parameter, you must specify it immediately after the command name.
2. The first authorized user, CONTROL, or USER (preferably CONTROL), must be authorized by the PARMLIB member.

DATABASEAUTHORITY and REVOKE: Giving authority to or revoking authority of a user

Explanation: **DATABASEAUTHORITY** and **REVOKE** are mutually exclusive, required parameters giving or revoking DFSMSHsm authorization of the specified user.

DATABASEAUTHORITY specifies the level of authorization of the specified user.

Subparameter	Explanation
USER	DATABASEAUTHORITY(USER) is an optional parameter that gives authorization to the specified user. The user specified with <i>userid</i> is authorized to use any DFSMSHsm command except the AUTH command.
CONTROL	<p>DATABASEAUTHORITY(CONTROL) is an optional parameter that gives a higher-level of authorization to the specified user. The user specified with <i>userid</i> is authorized to use the AUTH command to add, delete, or change the DFSMSHsm authorization of other users.</p> <p>Note: The CONTROL subparameter should be limited to as few user IDs as possible.</p>

REVOKE specifies that the *userid* identified is no longer authorized to issue all DFSMSHsm commands. The user can still issue DFSMSHsm user commands.

See [Figure 93 on page 757](#) for an illustration of DFSMSHsm authorization.

The DFSMSHsm user commands are described in [z/OS DFSMSHsm Managing Your Own Data](#).

Defaults: If the DATABASEAUTHORITY parameter is specified without a subparameter, the default is USER.

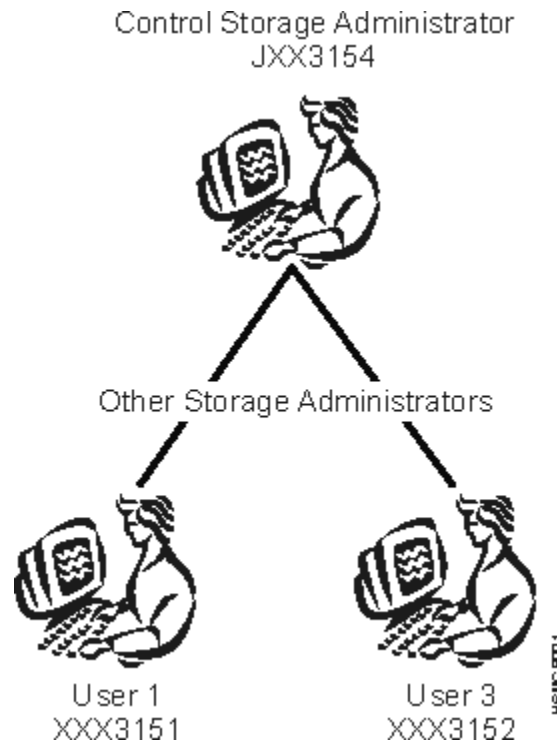


Figure 93. Identifying Authorized DFSMSHsm Users

Examples of how to code the AUTH command

The following examples present different ways to code the AUTH command:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Authorizing a user to issue all DFSMSHsm commands

The following command allows user ID JXX3154 to add, delete, or change the DFSMSHsm authorization of other users.

```
AUTH JXX3154 DATABASEAUTHORITY(CONTROL)
```

Authorizing a user to issue DFSMSHsm commands

The following command allows user ID XXX3151 to issue all authorized user commands except the AUTH command. This command can be issued by user ID JXX3154 or placed in the PARMLIB member.

```
AUTH XXX3151 DATABASEAUTHORITY
```

Revoking authority of a user

The following command cancels the authority to issue authorized DFSMSHsm commands for user ID XXX3152. This command can be issued by JXX3154 or placed in the PARMLIB member.

```
AUTH XXX3152 REVOKE
```


Chapter 32. BACKDS command: Backing up a specific data set

The BACKDS command creates a backup version of a specific data set or UNIX file. When you enter the BACKDS command, DFSMSHsm does not check whether the data set has changed or has met the requirement for frequency of backup. When you enter the BACKDS command for a UNIX file, DFSMSHsm will check whether the file has changed since its last backup. When DFSMSHsm processes a BACKDS command, it stores the backup version on either tape or the ML1 volume with the most available space.

Note: If the size of a data set or UNIX file that is targeted to DASD exceeds the free space on the ML1 volume having the most free space, then the data set backup fails. If there is no overflow ML1 volume available or the data set or UNIX file does not qualify for placement on an overflow ML1 volume, then the data set or UNIX file backup fails.

You can issue a BACKDS command for the following:

- An SMS-managed data set (if allowed by its management class).
- A data set that resides on a level 0 volume.
- A migrated data set currently residing on a migration level 1 volume.
- An uncataloged data set (DFSMSHsm is used as the datamover).
- A user data set that resides on a migration level 1 or a DASD migration level 2 volume. We do not recommend that users directly allocate data sets on migration volumes.
- A UNIX file.
- UNIX files within a directory, or directory sub-tree.

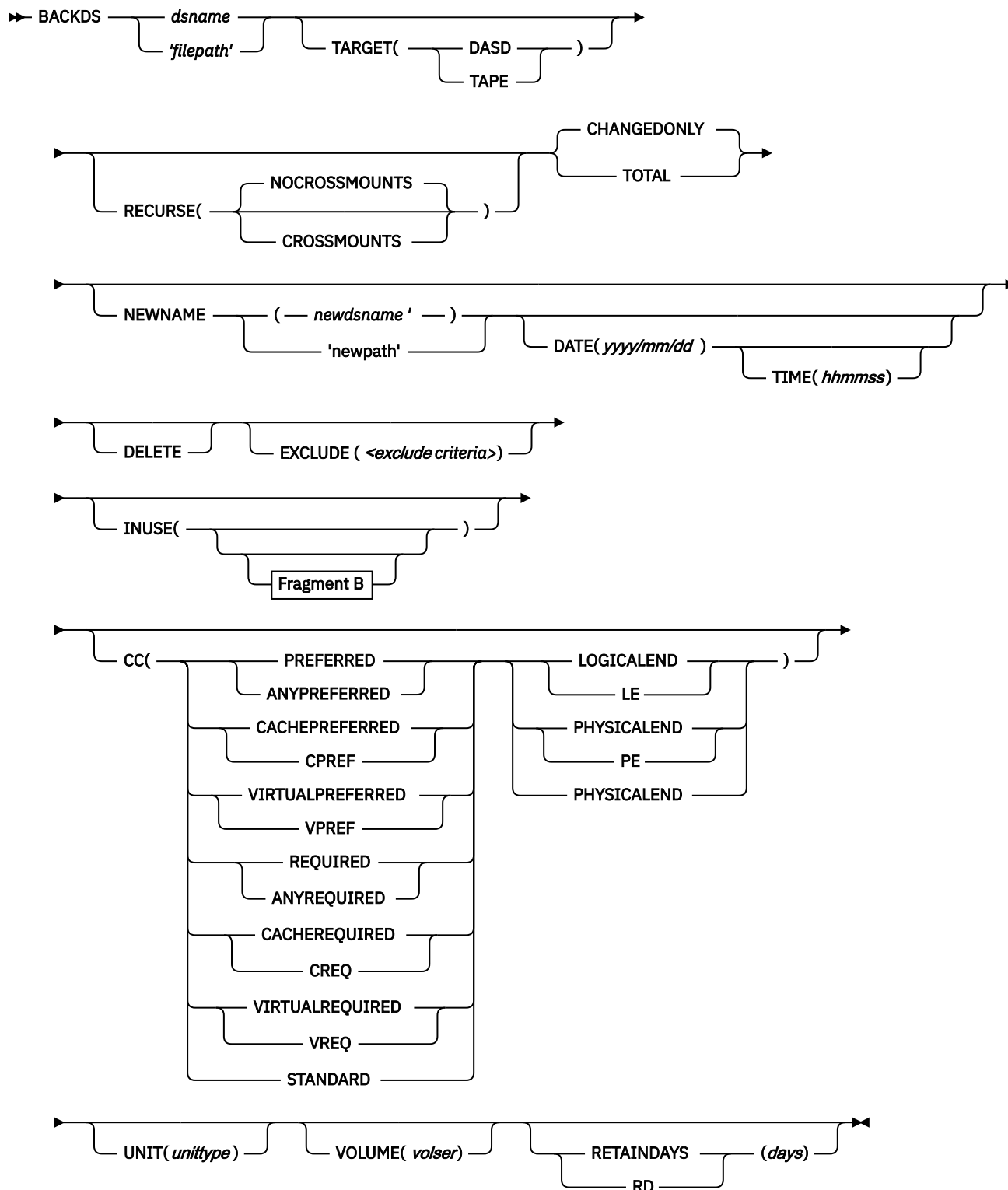
Regardless of the status or location of the data set, the volume containing the data set must be mounted and online, or the command fails.

When you want to back up an uncataloged data set, you must specify the volume serial number of the volume that contains the data set. In addition, you must specify the type of unit of the volume specified.

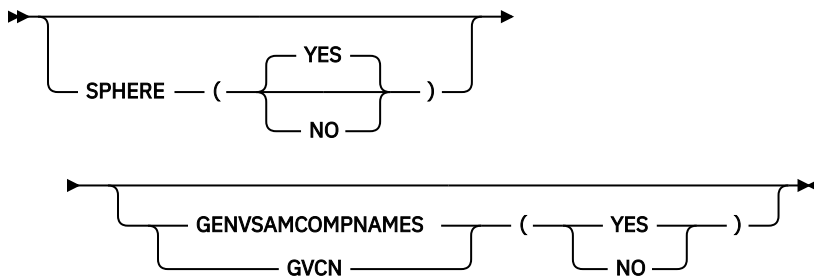
Note: If you specify a data set currently migrated to a migration level 2 volume, DFSMSHsm does not back up the data set.

Note: Any files that reside in an encrypted zFS will be decrypted as they are written to the backup.

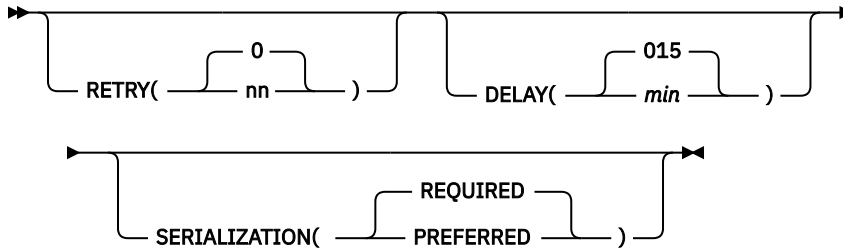
Syntax of the BACKDS command for a UNIX file



A: NEWNAME optional parameters



B: INUSE optional parameters



Note: The UNIT and VOLUME parameters do not apply to cataloged data sets.

Required parameters of the BACKDS command

This topic describes the required parameters of the BACKDS command.

The *dsname* and *filepath* parameters are mutually exclusive, required parameters that identify the data set or UNIX file to be backed up.

***dsname*: Specifying the name of the data set to be backed up**

Explanation: *dsname* is a required positional parameter that specifies the fully qualified name of the data set you want DFSMSHsm to back up. For *dsname*, substitute the fully qualified name of the data set.

Defaults: None.

Note:

1. Because *dsname* is a required positional parameter, you must specify it immediately after the command name. You cannot specify an alias for *dsname*. If you do, DFSMSHsm does not back up the data set.
2. The volume where the data set resides must be mounted before you issue the BACKDS command.
3. DFSMSHsm does not handle partitioned data set members individually. A member name of a partitioned data set, DFSMSHsm backs up the entire partitioned data set.

***filepath*: Specifying the name of the UNIX file to be backed up**

Explanation: *filepath* The specified name must be enclosed in single quotation marks and is considered to be made up of two parts: an absolute directory path that ends in a forward slash, followed by a *pattern* (for example, /directory/path/*pattern*):

Defaults: None.

- The directory path must begin with a forward slash and indicates a directory where DFSMSHsm should begin its search for a file or empty directory to back up.
- The pattern can contain any valid file name characters, and may contain wildcard characters question mark (?) and asterisk (*). If the name of the file should contain the specific ?, *, or \ characters, the backslash character (\) must be used to escape that character, indicating it is not a wildcard. (for example, \?, *, or \\)

- If the pattern doesn't contain any wildcard characters, this is treated as an absolute path to a file – (/u/dir1/file1.txt) That single file will be backed up.
- If the pattern doesn't contain any characters at all, or is only a single asterisk, this is treated as an absolute path to a directory - (for example, /u/dir1/ or /u/dir1/*) All files and empty directories found within the directory will be backed up.
- Wildcard in the pattern - An asterisk (*) can be used to match any number of characters within a file name. An asterisk may be used at any position in the pattern. A second asterisk may also be used at the end of the pattern to indicate any number of characters following the pattern is acceptable. A question mark (?) can be used to represent any single character in the file name. Multiple question mark characters are allowed.

Note: Because the filename is a required positional parameter, you must specify it immediately after the command name.

Optional parameters for the BACKDS command

This topic describes the optional parameters for the BACKDS command.

CC: Specifying concurrent copy for data set backup

CC is an optional parameter that specifies concurrent copy during BACKDS processing of data sets, or UNIX File Snapshot during BACKDS processing of UNIX files to provide a point-in-time backup copy. The parameters of the CC parameter are explained:

STANDARD

Specifies that you want to use standard backup methods without using concurrent copy or UNIX File Snapshot .

PREFERRED

ANYPREFERRED

Specifies the data is to be processed with concurrent copy if possible. Virtual concurrent copy is attempted first, if the storage subsystem on which the data resides is capable of virtual concurrent copy and working-space data sets have been defined. Otherwise, cache-based concurrent copy is attempted if the storage subsystem is capable of it. If both types of concurrent copy are not possible or fail, the data is processed by using standard I/O as if CONCURRENT were not specified.

When processing a UNIX file, DFSMSHsm will attempt to create a file snapshot of the file and take the backup from the clone after releasing serialization on the original. If the file snapshot cannot be successfully created, serialization will be obtained on the file and will be processed as if STANDARD was requested.

CACHEPREFERRED

CPREF

Specifies that data is to be processed with cache-based concurrent copy. Otherwise, the data is processed using standard I/O as if CONCURRENT were not specified. Virtual concurrent copy is not attempted.

When processing a UNIX file, DFSMSHsm will attempt to create a file snapshot of the file and take the backup from the clone after releasing serialization on the original. If the file snapshot cannot be successfully created, serialization will be obtained on the file and will be processed as if STANDARD was requested.

VIRTUALPREFERRED

VPREF

Specifies that data is to be processed with virtual concurrent copy if possible. Otherwise, the data is processed using standard I/O as if CONCURRENT were not specified. Cache-based concurrent copy is not attempted.

When processing a UNIX file, DFSMSHsm will attempt to create a file snapshot of the file and take the backup from the file snapshot after releasing serialization on the original. If the file snapshot cannot be successfully created, serialization will be obtained on the file and will be processed as if STANDARD was requested.

REQUIRED**ANYREQUIRED**

Specifies the data is to be processed with concurrent copy if possible. Virtual concurrent copy is attempted first, if the storage subsystem on which the data resides is capable of it and working-space data sets have been defined. Otherwise, cache-based concurrent copy is attempted if the storage subsystem is capable of it. If both types of concurrent copy are not possible or fail, the data will not be processed.

When processing a UNIX file, DFSMSHsm will attempt to create a file snapshot of the file. If the file snapshot cannot be successfully created, the backup will fail.

Note: Some files such as hard links cannot be processed with Clone. If backing up a directory or many files, the overall request will fail if one of these types is encountered and is unable to be processed.

CACHEREQUIRED**CREQ**

Specifies that data is to be processed with cache-based concurrent copy if possible. If a cache-based concurrent copy session cannot be established, the data set will not be backed up. Virtual concurrent copy and standard I/O are not attempted.

DFSMSHsm will attempt to create a file snapshot of the file. If the file snapshot cannot be successfully created, the backup will fail.

Note: Some files such as hard links cannot be processed with Clone. If backing up a directory or many files, the overall request will fail if one of these types is encountered and is unable to be processed.

VIRTUALREQUIRED**VREQ**

Specifies that data is to be processed with virtual concurrent copy if possible. If a virtual concurrent copy session cannot be established, the data set will not be backed up. Cache-based concurrent copy and standard I/O are not attempted.

DFSMSHsm will attempt to create a file snapshot of the file. If the file snapshot cannot be successfully created, the backup will fail.

Note: Some files such as hardlinks cannot be processed with Clone. If backing up a directory or many files, the overall request will fail if one of these types is encountered and is unable to be processed.

PHYSICALEND**PE**

Specifies that control returns to the application or user only after the backup physically completes.

LOGICALEND**LE**

Specifies that control returns to the application or user when concurrent copy or UNIX File Snapshot initialization completes.

Defaults: If you do not specify any subparameters with the CC parameter, the defaults are STANDARD and PHYSICALEND.

If you do not specify CC, then DFSMSHsm does not use concurrent copy for non-SMS data sets. However, for SMS data sets or UNIX files contained in an SMS-managed zFS, DFSMSHsm uses the values from the management class.

Note:

1. If LOGICALEND is used with PREFERRED, VIRTUALPREFERRED, or CACHEPREFERRED, and concurrent copy is unavailable for the data set, the backup continues and DFSMSHsm ignores the CC options.
2. If LOGICALEND is used with STANDARD, the command fails.
3. UNIX File Snapshot can only be initiated from the owning system for files in a zFS that is mounted NORWSHARE. If the zFS is mounted RWSHARE, UNIX file Snapshot can be used from any system in

the sysplex. Use the z/OS UNIX shell command **zfsadm fsinfo** to see details about the attributes of a file system. For example:

```
zfsadm fsinfo -owner file_system_name
```

CHANGEDONLY/TOTAL: Specifying whether to back up UNIX files that have a last backup date in their metadata

CHANGEDONLY/TOTAL

Specifies whether DFSMSHsm backs up only those UNIX files specified (explicitly or by a filter) that have a last backup date in their metadata that is older than the modification or metadata modification date. Additionally, if the change dates in the file metadata are newer than the most recent version in the DFSMSHsm inventory, or DFSMSHsm does not have a backup of the file, DFSMSHsm will create a backup. A change in the metadata modification date, but not the modification date, can be caused by truncation of the file, as well as a change to the number of hard links to the file. CHANGEDONLY is not valid when a data set name is specified. An error message (parsing) will be issued. The CHANGEDONLY keyword is overridden if specified or defaulted when NEWNAME is specified.

Abbreviations: CHGD

Defaults: CHANGEDONLY

TOTAL is an optional parameter that requests DFSMSHsm take a backup of the UNIX file regardless of the last backup date value.

Note:

This parameter does not apply to backup requests for data sets.

For restrictions please refer to topic "Data set backup using NEWNAME" <https://www.ibm.com/docs/en/zos/3.1.0?topic=backup-data-set-using-newname>

DELETE: Specifying whether to delete the UNIX file after a successful backup

DELETE is an optional parameter that specifies whether DFSMSHsm should remove the UNIX file after a successful backup has been performed. Directories are not removed. If the RETAINDAYS keyword is not specified, a default RETAINDAYS value of 365 (patchable **MCVT_ARCHIVE_RD_DEFAULT** field value) is used as the period to retain the backup of this file regardless of other backup versions that are subsequently taken. If the RETAINDAYS keyword is specified, then the specified value is used as the retention period.

Backup versions are not automatically rolled off by DFSMSHsm, even if retention period is passed. To remove backup versions that have passed the RETAINDAYS criteria the EXPIREBV command should be issued, backup versions will be removed then in compliance with the EXPIREBV logic.

Abbreviations: The TSO abbreviation convention applies for DELETE. There are no additional abbreviations

Defaults: DFSMSHsm does not delete a UNIX file after backup processing has been performed.

EXCLUDE and EXCLUDEF: Specifying files or directories to be excluded from processing

EXCLUDE | EXCLUDEF are mutually exclusive, optional parameters that are used to specify files and directories that are to be excluded during Unix File backup processing.

EXCLUDE(<exclude_criteria>) - specifies a comma-separated list of exclude criteria directly on the command. This criteria is to be applied to the UNIX files examined during the UNIX File backup processing.

The exclude criteria list specified on the command with EXCLUDE keyword has the following restrictions and rules unique to the keyword:

- The *exclude_criteria* list must be specified in single quotes
- The *exclude_criteria* list cannot exceed 1023 characters

EXCLUDEF('<filepath>') - specifies the UNIX path to a file that contains the exclude criteria to be applied to the UNIX files examined during the UNIX File backup processing. The file path must be specified in single-quotes and must be an absolute path. Any imbedded single quotes in the filename must be specified with a second single quote, e.g. the filename exclude'criteria.txt would be specified as: 'exclude"criteria.txt'. (optional)

The exclude criteria file specified using the EXCLUDEF keyword has the following restrictions and rules unique to the keyword:

- Line starting with a blank character is a comment.
- Each line can only contain one criterion.
- A single line cannot be longer than 1023 characters.
- Total length of all exclude criteria with added commas to separate them cannot exceed 32767 characters. For 'n' number of criteria, 'n-1' number of COMMA should also be accounted for to calculate the total size.

Exclude criteria, whether specified on the command in a comma separated list using the EXCLUDE keyword or specified within a file using the EXCLUDEF keyword, must conform to the following rules:

- Absolute directory path, beginning and ending with a forward-slash character.
- Partial directory path, ending with a forward-slash character. If the directory name begins with a forward slash character, then it is considered an absolute path. If the directory name does not begin with a forward slash character, it is considered a relative path that begins at the directory requested on the command. A directory name that is a relative path is considered relative to the directory requested on the command only, even when the command requests recursion.
- Filename pattern (does not contain forward slashes). Can contain * or ? wildcard. An asterisk (*) can match with 0 or more characters in a filename. A question mark (?) can match with a single character. Wildcard characters may be escaped with the backslash character if desired.
- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.
- If a filename contains special characters that are not to be treated as wildcard or escape characters i.e. an asterisk (*), a question mark (?), or backslash (\), these characters must be escaped using a backslash (\). If these characters are not escaped, they will be treated as wildcard or escape characters.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories.
- If a criterion begins with a directory wildcard (/.../) then it is treated as relative path.
- A comma cannot be used with the exclude criterion.
- A single criterion cannot exceed 1023 characters.
- A single forward slash or a single asterisk are treated as invalid patterns.
- A directory path cannot contain consecutive forward slashes.

The example below shows sample criteria in a file specified by the EXCLUDEF keyword. The same criteria could be specified on the command using the EXCLUDE keyword where each criterion is separated by a comma.


```

Useful comment (maybe with commas) <= blank in 1st position - ignored
/a/b/c/                                     <= empty line - ignored
/a/b/c/x1.txt                             <= absolute directory
/a/.../d/.../g/syslog1.txt                <= absolute directory with filename
/a/.../d/.../g/*log?.txt                  <= absolute directory with wildcards
/a/.../d/.../g/*log?.txt                  <= absolute directory and filename with wildcards
e/f/g                                     <= relative directory
e/f/g/mvs.log                             <= relative directory with filename
d/e/f/g/*log?.txt                         <= relative directory with filename wildcard
d/.../g/sys*. *                           <= relative directory and filename with wildcards
/.../d/.../g/sys*. *                     <= relative difrectory and filename with wildcards
*.log                                     <= filename with wildcards
*.log2                                   <= blank in 1st position makes a comment, ignored
\*abc                                    <= escaping wildcards - files with exact name *abc
\\abc                                   <= escaping wildcards - files with exact name \abc

```

Defaults: No files are excluded from processing.

INUSE: Tailoring backup processing of UNIX files in use at the time of backup

The INUSE keyword is an optional parameter that specifies how UNIX files that are in use at the time of backup should be processed.

When the INUSE keyword is specified, UNIX files that are in use at the time of backup will have their backup retried up to a specified maximum number of times, delaying a specified number of minutes in between each retry. On retry an attempt to obtain exclusive serialization to the UNIX file will be made. If on retry exclusive serialization is obtained then a normal backup of the UNIX file will be made. If exclusive serialization cannot be obtained, then additional retries of the backup will be made until the max number of retries have been made. If after the maximum number of retries exclusive serialization cannot be obtained, the backup request will either fail or a backup of the file will be taken anyway, depending on if the command specified serialization was required or preferred.

The INUSE keyword has subkeywords, RETRY, DELAY, and SERIALIZATION that enable the user to specify the maximum number of retries, the time to delay in between retries, and whether exclusive serialization is required or preferred.

INUSE(RETRY(*nn*),DELAY(*min*),SERIALIZATION(REQUIRED|PREFERRED))

- RETRY(*nn*) - DFSMSHsm will retry a maximum of *nn* times to back up a UNIX file after the first attempt fails because the UNIX file is in use. For *nn*, specify a value 0 to 99. The default value is 0.
- DELAY(*min*) - DFSMSHsm delays for *min* minutes before retrying a backup attempt which failed because the UNIX file in use. DFSMSHsm will delay *min* minutes in between each retry. For *min*, specify a value from 0 to 999. The default value is 15.
- SERIALIZATION(PREFERRED|REQUIRED) - Specifies whether exclusive serialization of the UNIX file is required after the max number of retries of the backup have been made. The default value for SERIALIZATION is REQUIRED.
 - PREFERRED - If the UNIX file is still in use after the max number of retries for the backup have been made, the UNIX file will be backed up anyway. A message indicating a fuzzy backup was made will be issued. If the UNIX file is not in use at that time, a normal backup will be made. If RETRY(0) is in effect and exclusive serialization could not be obtained on the first attempt then a fuzzy backup will be made.
 - REQUIRED - If the UNIX file is still in use after the max number of retries for the backup is made, the backup request will fail.

The subkeywords are optional and any that are omitted will have the default value for that subkeyword used.

All supported UNIX file types can be backed up with the INUSE keyword, however only Regular Files will be checked to be in use. Symbolic links, FIFOs, and directories will not be checked to be in use and will not have their backup retried.

zFS File Snapshot is not supported for in use UNIX files. If SERIALIZATION(PREFERRED) and CC(PREF) or CC(REQ) are specified on the command and the UNIX file is in use at the time of backup, the backup request will still fail.

NEWNAME: Specifying a name to assign to the backup version

NEWNAME is an optional parameter. *dsname* is the name of the data set to back up, and *newdsname* is the data set name to assign to the new backup version that is created by specifying BACKDS. Both *dsname* and *newdsname* must be fully qualified and in the standard data set name format. DATE, TIME, SPHERE, and GENVSAMCOMPNames are optional parameters that are only valid with the NEWNAME parameter.

newpath is a new directory path for the UNIX file backups. When processing a UNIX file, this subparameter requests DFSMSHsm to create a backup of the requested UNIX files as if it had come from the specified directory *newpath*. The specified *newpath* must be a subset of the request directory path, and must start and end with a forward slash. When processing a request with the *newpath* parameter, the *newpath* value is checked against the request directory path starting at the end of the path, moving to the higher directory levels to ensure that the specified *newpath* is a subset matching the requirements above. If an intermediate directory of the request directory path is specified as the *newpath*, the remaining lower directory portions of the request directory path are appended (see example 1.4). The resulting new directory should not be equal to the request directory path (see examples 1.5 - 1.7 below which result in ARC1381I REAS=10). If the *newpath* is not a subset of the request directory, then the ARC1334I BACKUP VERSION NOT CREATED with REAS=23 is issued (see example 1.8).

The NEWNAME keyword is intended to assist migrating backups from a previous product into DFSMSHsm. Backups would be recovered into a temporary directory, and then those files would be backed up by DFSMSHsm, specifying the new root directory of the files using the NEWNAME keyword. DFSMSHsm matches the last directory subset of the request. The CHANGEDONLY keyword is overridden if specified or defaulted when NEWNAME is specified.

Examples:

```
BACKDS '/tmp/2017-07-08/prod/webapp1/*' RECURSE NEWNAME('/prod/webapp1/') DATE(2017/07/08)
```

```
BACKDS '/tmp/2017-07-08/*' RECURSE NEWNAME('/') DATE(2017/07/08)
```

Incorrect:

```
BACKDS '/tmp/2017-07-08/prod/webapp1/*' RECURSE NEWNAME('/user/myapp/')
```

Examples showing the resulting backup when using different paths specified in the NEWNAME keyword.

BACKDS '/x/y/z/file.txt' with various *newpath* values:

1.1. NEWNAME('/') - is backed up as /file.txt.

1.2. NEWNAME('/z/') - is backed up as /z/file.txt.

1.3. NEWNAME('/y/z/') - is backed up as /y/z/file.txt.

1.4. NEWNAME('/y/') - is backed up as /y/z/file.txt, since '/z/' of the source path is appended to '/y/', and 'file.txt' is backed up as if it came from '/y/z/' directory.

1.5. NEWNAME('/x/y/z/') - is an error with ARC1381I REAS=10 issued.

1.6. NEWNAME('/x/') - is an error with ARC1381I REAS=10 issued.

1.7. NEWNAME('/x/y/') - is an error with ARC1381I REAS=10 issued.

1.8. NEWNAME('/not_subset/') - is an error with ARC1334I REAS=23 issued.

Tip: It is best if the NEWNAME directory exists at the time of the backup. DFSMSHsm stores the name of the containing zFS data set in the backup record for future SMS-management checks. (This attribute is used for the number of versions to keep for roll-off processing, ALTERDS checking, and whether the Recover processing should end due to a No-Crossmounts condition on a recursive file request. If

the NEWNAME directory does not exist, DFSMSShsm cannot store the containing zFS name, and some functions, such as ALTERDS might be unable to be performed.

The parameters of the NEWNAME parameter are explained:

DATE

Specifies the date to assign to the backup version. If DATE is specified without the NEWNAME parameter, the BACKDS command fails. If DATE is specified without TIME, the current time is assigned to the backup version. The date format is: *yyyy/mm/dd*; where *yyyy* is the 4-digit year, *mm* is the 2-digit month, and *dd* is the 2-digit day.

TIME

Specifies the time to assign to the new backup version. If TIME is specified without DATE, the BACKDS command fails. If DATE is specified without TIME, the current time is assigned to the backup version. If TIME is not specified, and a backup copy with the date specified on the command and the current time already exists, the command fails. The time format is: *hhmmss*; where *hh* is 2-digit hour, *mm* is the 2-digit minute, and *ss* is the 2-digit second.

SPHERE

Specifies whether AIX and PATH components of a VSAM data set is backed up with the base cluster. SPHERE(YES) indicates that a backup of the entire VSAM sphere is performed. SPHERE(NO) indicates that a backup of the base cluster is performed, and any associated AIXs or PATHs is not backed up. SPHERE(YES) is the default. If SPHERE is specified without NEWNAME, the BACKDS command fails.

GENVSAMCOMPNames**GVCN**

Specifies whether to override the SETSYS DSBACKUP(GENVSAMCOMPNames) value. If GENVSAMCOMPNames(YES) is specified and *dsname* is a VSAM data set and *newdsname* is a migrated or uncataloged data set, the command is processed, and default names are assigned to the data and index components. If GENVSAMCOMPNames(NO) is specified, and *dsname* is a VSAM data set and *newdsname* is a migrated or uncataloged data set, the command fails. If *newdsname* is a cataloged data set and is not migrated, or if *dsname* is a non-VSAM data set, the GENVSAMCOMPNames parameter is ignored.

Defaults: If date and time are not specified, the current date and time are assigned to the backup version.

Note: The following restrictions apply when using the BACKDS NEWNAME command to back up a VSAM data set:

- No more than one PATH can be defined for the data set to be backed up, and no more than one PATH can be defined for the new name data set.
- No more than one AIX can exist for the data set to be backed up, and no more than one AIX can exist for the new name data set.
- The NEWNAME keyword cannot be specified with the UNIT and VOLUME keywords.
- A backup version created with (H)BACKDS NEWNAME DATE always has a VERSION NUMBER = 000.

newpath is a new directory path for the UNIX file backups. When processing a UNIX file, this subparameter requests DFSMSShsm to create a backup of the requested UNIX files as if it had come from the specified directory *newpath*. The specified *newpath* must be a subset of the request directory path, and must start and end with a forward slash. The parameter is the absolute path of the directory name that the files should have after the backup is complete. The backups are made as if they came from root.

The NEWNAME keyword is intended to assist migrating backups from a previous product into DFSMSShsm. Backups would be recovered into a temporary directory, and then those files would be backed up by DFSMSShsm, specifying the new root directory of the files using the NEWNAME keyword. DFSMSShsm matches the last directory subset of the request. The CHANGEDONLY keyword is overridden if specified or defaulted when NEWNAME is specified.

Examples:

```
BACKDS '/tmp/2017-07-08/prod/webapp1/*' RECURSE NEWNAME('/prod/webapp1/') DATE(2017/07/08)
```



```
BACKDS '/tmp/2017-07-08/*' RECURSE NEWNAME('/') DATE(2017/07/08)
```

Incorrect:

```
BACKDS '/tmp/2017-07-08/prod/webapp1/*' RECURSE NEWNAME('/user/myapp/')
```

Examples showing the resulting backup when using different paths specified in the NEWNAME keyword.

- BACKDS '/x/y/z/file.txt' 1.1. NEWNAME('/') is backed up as /file.txt.
- 1.2. NEWNAME('/z/') is backed up as /z/file.txt.
- 1.3. NEWNAME('/y/z/') is backed up as /y/z/file.txt.
- 1.4. NEWNAME('/x/y/z/') is an error.
- 1.5. NEWNAME('/y/') is backed up as /y/z/file.txt.

Note: It is best if the NEWNAME directory exists at the time of the backup. DFSMSHsm stores the name of the containing zFS data set in the backup record for future SMS-management checks. (This attribute is used for the number of versions to keep for roll-off processing, ALTERDS checking, and whether the Recover processing should end due to a No-Crossmounts condition on a recursive file request. If the NEWNAME directory does not exist, DFSMSHsm cannot store the containing zFS name, and some functions, such as ALTERDS might be unable to be performed.

RECURSE: Examine subdirectories of the specified UNIX file path

Explanation: RECURSE is an optional parameter that requests DFSMSHsm go into subdirectories as it looks for files to process. As new directories are examined, the pattern is used to match with file names within that directory.

RECURSE(CROSSMOUNTS|CMT|NOCROSSMOUNTS|NCM)

NOCROSSMOUNTS is the default behavior and means that as DFSMSHsm is traversing the directory sub-tree, it should not cross into a different mounted filesystem while looking for files to process. If a different filesystem is encountered, examination of that sub-tree will stop.

CROSSMOUNTS tells DFSMSHsm that it should cross into different filesystems as it is traversing the file tree while looking for files that match *pattern* to backup.

Defaults: None. Recursion into sub-directories is not done by default. Only the requested directory is examined for files that match *pattern*.

Note: This parameter does not apply to data set backup requests.

RETAIN_DAYS: Specifying the number of days to retain a backup copy

Explanation: RETAIN_DAYS is an optional parameter specifying a number of days to retain a specific backup copy of a data set. If you specify RETAIN_DAYS, number of retain days is a required parameter that specifies a minimum number of days (0–50000) that DFSMSHsm retains the backup copy. If you specify 99999, the data set backup version never expires. Any value greater than 50000 (and other than 99999) causes a failure with an ARC1605I error message. A retain days value of 0 indicates that:

- the backup version might expire within the same day that it was created if EXPIREBV processing takes place or when the next backup version is created,
- the backup version is kept as an active copy before roll-off occurs,
- the backup version is not managed as a retained copy.

Restriction: RETAIN_DAYS applies only to cataloged data sets.

Defaults: None.

Related reading: For more information about the RETAIN_DAYS processing, see [“Specifying the number of days to retain a backup copy of a data set” on page 183](#).

TARGET: Specifying a required target output device

Explanation: TARGET is an optional parameter specifying a required output device type, such as ML1 DASD or backup tape. If you do not specify the TARGET parameter, DFSMSHsm selects the output device.

The parameters of the TARGET parameter are explained:

DASD

specifies that DFSMSHsm backs up the data set to ML1 DASD.

TAPE

specifies that DFSMSHsm backs up the data set to tape.

Defaults: None.

UNIT: Specifying the type of device

Explanation: UNIT is an optional parameter specifying the type of unit where the volume containing the uncataloged data set to be backed up can be allocated.

unittype

is the type of unit.

The following are valid types of DASD units:

- 3380
- 3390
- 9345

Defaults: None.

Note: You must specify the UNIT parameter if you want to back up an uncataloged data set. UNIT, you must also specify the VOLUME parameter. Do not specify this parameter if you want to back up a cataloged data set.

VOLUME: Specifying the volume where the data set resides

Explanation: VOLUME is an optional parameter specifying the volume where the uncataloged data set to be backed up resides.

volser

is the serial number of the volume where the uncataloged data set to be backed up resides.

Defaults: None.

Note: You must specify the VOLUME parameter if you want to back up an uncataloged data set. VOLUME, you must also specify the UNIT parameter. Do not specify this parameter if you want to back up a cataloged data set.

Delete expired backup versions of UNIX files

Retained backup versions that have met their retention period are not automatically deleted. You will have to use the EXPIREBV command to find and delete unneeded backup versions. When the EXPIREBV command is processing a backup of a UNIX file, the management class of the zFS data set containing the UNIX file is used to determine the expiration criteria for the backup.

- If the zFS data set that contained the UNIX file does not exist, the management class that was associated with the file when the last backup version was made is used.
- If the UNIX file is determined to be non-SMS managed, then the DELETEIFBACKEDUP or CATALOGEDDATA subparameters of the NONSMSVERSIONS parameter criteria is used.

Examples of how to code the BACKDS command

The following examples present different ways to code the BACKDS command:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Backing up an SMS-managed data set

In this example, DFSMSHsm backs up an SMS-managed data set to a target device of tape. DFSMSHsm requires a concurrent copy session, and DFSMSHsm returns control after the session is established and before DFSMSHsm makes the copy.

```
BACKDS P234823.PSFB.H39SC230.DS230 TARGET(TAPE) CC(REQUIRED LE)
```

Backing up a cataloged data set

In this example, DFSMSHsm backs up a cataloged data set.

```
BACKDS JBH6798.PSFB.F40RC001.DSET01
```

Backing up an uncataloged data set

In this example, DFSMSHsm backs up an uncataloged data set. You must specify the VOLUME and UNIT parameters to back up an uncataloged data set.

```
BACKDS FMT1315.PROJ53.REV VOLUME(VOL110) UNIT(3390)
```

Chapter 33. BACKFILE command: Backing up a specific UNIX file

The BACKFILE command is an alias to the BACKDS command. All parameters valid on the BACKDS command are acceptable on the BACKFILE command.

Chapter 34. BACKDS command: Backing up a specific data set

The BACKDS command creates a backup version of a specific data set or UNIX file. When you enter the BACKDS command, DFSMSHsm does not check whether the data set has changed or has met the requirement for frequency of backup. When you enter the BACKDS command for a UNIX file, DFSMSHsm will check whether the file has changed since its last backup. When DFSMSHsm processes a BACKDS command, it stores the backup version on either tape or the ML1 volume with the most available space.

Note: If the size of a data set or UNIX file that is targeted to DASD exceeds the free space on the ML1 volume having the most free space, then the data set backup fails. If there is no overflow ML1 volume available or the data set or UNIX file does not qualify for placement on an overflow ML1 volume, then the data set or UNIX file backup fails.

You can issue a BACKDS command for the following:

- An SMS-managed data set (if allowed by its management class).
- A data set that resides on a level 0 volume.
- A migrated data set currently residing on a migration level 1 volume.
- An uncataloged data set (DFSMSHsm is used as the datamover).
- A user data set that resides on a migration level 1 or a DASD migration level 2 volume. We do not recommend that users directly allocate data sets on migration volumes.
- A UNIX file.
- UNIX files within a directory, or directory sub-tree.

Regardless of the status or location of the data set, the volume containing the data set must be mounted and online, or the command fails.

When you want to back up an uncataloged data set, you must specify the volume serial number of the volume that contains the data set. In addition, you must specify the type of unit of the volume specified.

Note: If you specify a data set currently migrated to a migration level 2 volume, DFSMSHsm does not back up the data set.

Note: Any files that reside in an encrypted zFS will be decrypted as they are written to the backup.

Chapter 35. BACKFILE command: Backing up a specific UNIX file

The BACKFILE command is an alias to the BACKDS command. All parameters valid on the BACKDS command are acceptable on the BACKFILE command.

Chapter 36. BACKVOL command: Backing up data sets from volumes or dumping volumes

You can use the BACKVOL command to request that DFSMSHsm perform any of the following tasks:

- Back up eligible data sets on one or a list of volumes, or dump one or a list of DASD volumes
- Back up eligible data sets from all primary volumes
- Back up control data sets and the journal
- Back up eligible data sets from all volumes in one or more storage groups
- Dump all volumes in one or more storage groups

Use the BACKUPDEVICECATEGORY parameter to specify whether to back up the data sets to tape or to DASD daily backup volumes.

Backup messages are always recorded in the backup activity log.

When you issue the BACKVOL command with the DUMP optional parameter, DFSMSHsm requests that DFSMSdss does a full-volume dump of either (1) each DASD volume specified with the VOLUMES parameter or (2) each eligible DASD volume in each storage group specified with the STORAGEGROUP parameter. In the first case, the full-volume dumps are stacked on dump volumes in those dump classes if:

- Dump classes specified on the command call for stacking, or
- The STACK or MINSTACK subparameter is specified

In the second case, the full-volume dumps are stacked on dump volumes if:

- Dump classes specified on the command call for stacking, or
- The STACK or MINSTACK subparameter is specified, or
- The dump classes associated with a storage group or volume call for stacking

For full-volume dumps, you can decide to print dump messages at the system console (TERMINAL). Dump messages are always recorded in the dump activity log.

You can specify the BACKVOL command with the CONTROLDATASETS parameter to back up the DFSMSHsm control data sets and journal or you can specify CONTROLDATASETS(NULLJOURNALONLY) to null the journal without backing up the control data sets. Backup messages for the control data sets are written to the backup activity log.

Backing up data sets from one or more SMS-managed volumes or dumping SMS-managed volumes

The BACKVOL command requests that DFSMSHsm backs up eligible data sets from specific SMS-managed volumes or from the volumes in one or more storage groups. Only those data sets for which the management class allows command backup are processed. The frequency of backup for an SMS-managed data set is obtained from the data set management class attribute.

When you issue the BACKVOL command for an SMS-managed volume, that volume becomes DFSMSHsm-managed. DFSMSHsm maintains control records for the volume until you issue the DELVOL command against it. When processing a BACKVOL command having the STORAGEGROUP parameter, the management class selection criterion that is used is AUTOBACKUP instead of COMMAND. This means that DFSMSHsm treats a BACKVOL STORAGEGROUP command as if it is autobackup running for the referenced volumes.

You can specify, on the command, the type of backup to be done. You can choose to back up the following:

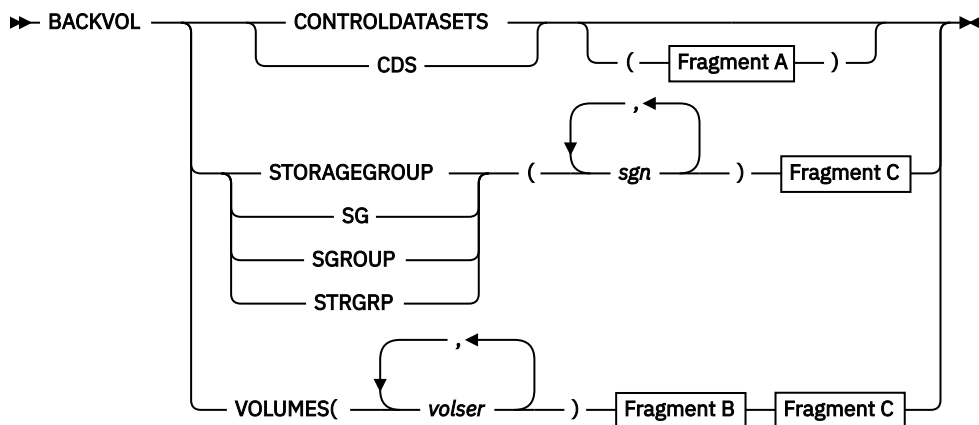
BACKVOL command for DFSMSHsm

- Data sets that have changed since the last backup (INCREMENTAL)
- All data sets on a volume (TOTAL)

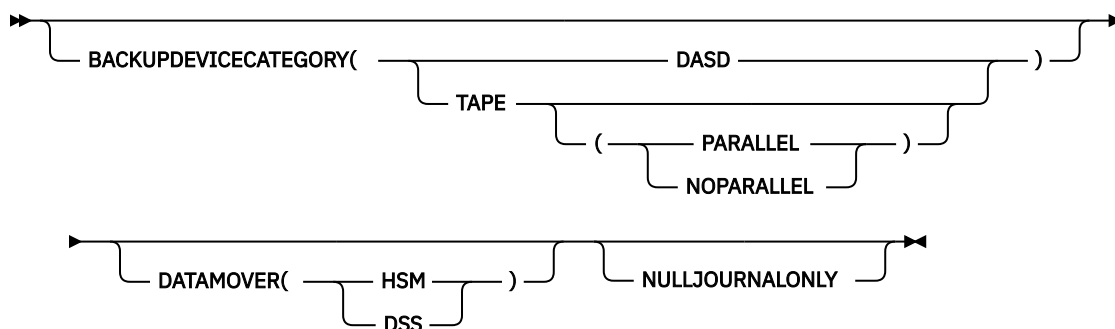
The BACKVOL command requests DFSMSHsm to dump specific SMS-managed volumes or volumes in one or more storage groups. When a BACKVOL command with the DUMP parameter is issued for an SMS-managed volume, the dump functions performed for the volume are the same as those performed for an automatic dump of that volume.

If dump classes are specified on the BACKVOL command, those dump classes are used to perform the dump function, overriding the dump classes defined for the SMS storage group associated with that volume.

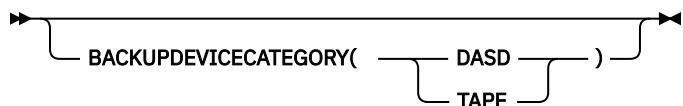
Syntax of the BACKVOL command with SMS-managed volumes



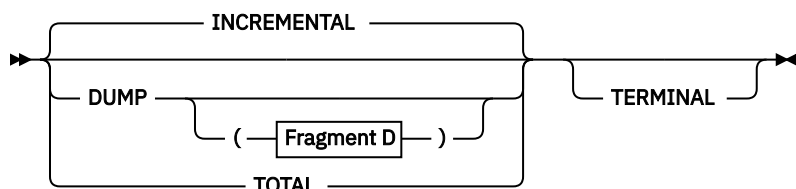
A: CONTROLDATASETS optional parameters



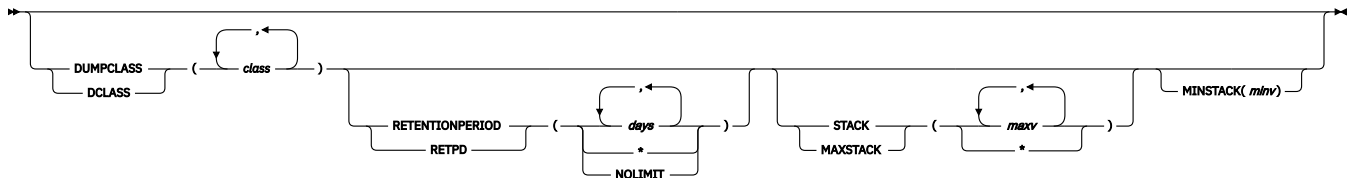
B: VOLUMES optional parameters



C: STORAGEGROUP and VOLUMES optional parameters



D: DUMP optional parameters



Required parameters for SMS-managed volumes

This topic describes the required parameters of the BACKVOL command.

CONTROLDATASETS: Specifying the control data sets to be backed up

Explanation: CONTROLDATASETS is a required parameter specifying that DFSMSHsm back up the control data sets and journal.

Defaults: None.

Note:

1. The BACKVOL CONTROLDATASETS command is not affected by running DFSMSHsm in debug mode (using the SETSYS DEBUG command). Control data sets are backed up in both debug and normal (no debug) mode.
2. Do not issue the CONTROLDATASETS parameter of the BACKVOL command during a period of high DFSMSHsm activity, because the control data sets cannot be changed while this function is running. However, this is *not* true if you are using DFSMSdss with the concurrent copy function as the datamover. Most DFSMSHsm functions change the control data sets. The only way to prevent DFSMSHsm from backing up the control data sets after someone issues the BACKVOL command with the CONTROLDATASETS parameter is to stop DFSMSHsm.
3. DFSMSdss must be specified as the datamover for CDS backup when you are using RLS mode. If BACKVOL CDS is issued, and DFSMSdss was not specified as the datamover, or is not the default, or the backup is directed to tape and PARALLEL was not specified or is not the default, then message ARC0792I will be issued, and the command will fail. If CDS backup is invoked automatically, and either DFSMSdss is not specified as the datamover, or the backup is directed to tape and PARALLEL was not specified, then these values will override the existing values and message ARC0793I will be issued.

STORAGEGROUP: Specifying the storage group to back up or dump

Explanation: STORAGEGROUP(*sgn*,...) is a required parameter specifying that DFSMSHsm backs up the eligible data sets on the volumes in one or more storage groups or it dumps the volumes in one or more storage groups. For *sgn*, substitute the names of the storage groups that DFSMSHsm processes. (A storage group name appearing more than once in the list is backed up or dumped only once.)

If you specify DUMP, DFSMSHsm concurrently targets dump copies to one or more dump classes (specified in the command or associated with the storage groups). If stacking is specified (in the command or in the dump class definitions), DFSMSHsm uses its dump stacking process to stack multiple dump copies on dump volumes. Specifying DUMPCLASS means that DFSMSHsm dumps all storage groups specified to the same set of specified dump classes.

Defaults: None.

Note:

1. A message is issued, and the backup or dump is not processed if DFSMSHsm is running in a non-SMS environment.
2. A message is issued and the backup or dump will not be processed for any volumes which are in initial status or can not be determined to be SMS-managed volumes.
3. If you specify STORAGEGROUP without *sgn*, the command fails. If you specify STORAGEGROUP with more than 30 storage groups, only the first 30 are processed.

4. If you specify DUMP with DUMPCLASS, and at least one of the dump classes specified has a stacking value greater than one (specified with the STACK subparameter or within a dump class definition), DFSMSHsm uses its dump stacking process to stack one or more dump copies on each dump volume selected.
5. If you specify DUMP but not DUMPCLASS, and at least one of the dump classes associated with a specified storage group has a stacking value greater than one, DFSMSHsm uses its dump stacking process to stack one or more dump copies of the volumes in that storage group on dump volumes in the associated dump classes.
6. For any dump class with a stacking value of one (specified on the command or within a dump class definition), DFSMSHsm limits that class's dump volumes to one dump copy per volume.
7. If you specify DUMP but not DUMPCLASS, then depending on how storage groups and their associated dump classes are defined, DFSMSHsm may not dump the eligible volumes in the order in which the storage groups are specified. In particular, if the dump classes for a given storage group do not specify stacking, the volumes of that storage group are dumped (one dump copy per dump volume) after the volumes of those storage groups using stacking.
8. Dump copies of volumes in more than one storage group (specified on the same BACKVOL command) can be stacked on a given dump volume.
9. All stacking to a given dump volume is done while that tape is mounted. DFSMSHsm always selects empty dump volumes to mount as targets for DUMP.
10. When you use BACKVOL to dump one or more storage groups, DFSMSHsm treats all volumes being dumped as having affinity to the host issuing the command which means that they are all processed on the single host where the command is submitted.
11. If a batch job issues the WAIT BACKVOL STORAGEGROUP command, DFSMSHsm processes the command, but does not ensure that the action completes before returning control to the caller. As a result, it is possible for the batch job to complete with return code zero before all of the volumes are processed.
12. If you are using the BACKVOL STORAGEGROUP command, data sets that are assigned to a Management Class with AUTO BACKUP=N will not be considered eligible for backup.

VOLUMES: Specifying the volumes to be backed up or dumped

Explanation: VOLUMES(*volser*,...) is a required parameter specifying that DFSMSHsm is to (1) back up the eligible data sets on, or (2) dump, each of the specified volumes. For *volser*, substitute the serial number of the volumes to be backed up or dumped. (A volume serial number appearing more than once in the list is backed up or dumped only once.) A message is issued and the backup or dump of an SMS-managed volume is not processed if:

- DFSMSHsm is running in a non-SMS environment
- The SMS-managed volume is in *initial* status (contains both SMS-and non-SMS-managed data sets while in the process of being converted to an SMS-managed volume).
- DFSMSHsm cannot determine if the specified volume is SMS-managed.

Defaults: None.

Note:

1. A volume list can include both SMS-managed volumes and non-SMS-managed volumes.
2. If you specify the VOLUMES parameter with more than 80 volumes, DFSMSHsm processes only the first 80 volumes.
3. If DUMP is specified with DUMPCLASS, and at least one of the specified dump classes (or the optional STACK subparameter) specifies a stacking value greater than one, DFSMSHsm attempts to stack the dumps of the volumes in the order specified. A volume in use by another DFSMSHsm volume function will be moved to the end of the stack.

Optional parameters for SMS-managed control data sets

This topic describes the optional parameters of the BACKVOL command.

BACKUPDEVICECATEGORY: Specifying the backup device for the control data sets and journal

Explanation: BACKUPDEVICECATEGORY is an optional parameter of CONTROLDATASETS that specifies the type of device that receives the backup copies of the DFSMSHsm control data sets and the journal when DFSMSHsm backs them up.

Subparameter	Explanation
DASD	BACKUPDEVICECATEGORY(DASD) specifies that DASD is the device on which the backup copies will be recorded.
TAPE	<p>BACKUPDEVICECATEGORY(TAPE) specifies that tape is the device on which the backup copies will be recorded.</p> <p>TAPE(PARALLEL) specifies that all control data sets and the journal should be backed up in parallel. By specifying this subparameter, you can cause one tape drive to be allocated for each control data set and the journal.</p> <p>TAPE(NOPARALLEL) specifies that the control data sets and the journal should be backed up serially. When NOPARALLEL is specified, the control data sets are backed up one at a time followed by the backup of the journal. By specifying this parameter, you can cause only one tape drive to be allocated for all the control data set backups. TAPE(NOPARALLEL) may not be specified when DATAMOVER(DSS) is specified.</p>

SETSYS Defaults: None.

DFSMSHsm Defaults: If you do not specify BACKUPDEVICECATEGORY, the backup device used will be determined by the SETSYS CDSVERSIONBACKUP subparameter BACKUPDEVICECATEGORY. If the SETSYS CDSVERSIONBACKUP subparameter BACKUPDEVICECATEGORY has not been previously specified, the DFSMSHsm default is BACKUPDEVICECATEGORY(TAPE(NOPARALLEL)).

Note:

1. If BACKVOL CDS specifies DATAMOVER(DSS), or if BACKVOL CDS does not specify a datamover and the DFSMSHsm CDS Version Backup environment indicates DATAMOVER(DSS), then TAPE(NOPARALLEL) may not be specified.
2. If you are attempting to backup to tape, ensure that you have previously issued the SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(TAPE (UNITNAME(*unit*))) command to properly configure the CDSVERSIONBACKUP parameter table.

DATAMOVER: Specifying the type of data movement used when backing up the control data sets

Explanation: DATAMOVER is an optional parameter of CONTROLDATASETS specifying the type of data movement to be used when performing a backup of the DFSMSHsm control data sets. DFSMSdss must be specified as the datamover for CDS backup when you are using RLS mode.

Subparameter	Explanation
HSM	DATAMOVER(HSM) specifies that the DFSMSHsm control data sets are backed up using the AMS EXPORT function. The journal is backed up using sequential I/O.

Subparameter	Explanation
DSS	DATAMOVER(DSS) specifies that the DFSMSHsm control data sets are backed up using a DFSMSdss logical dump. Using DFSMSdss as the datamover allows you to exploit the concurrent copy function. The journal is backed up using sequential I/O. Data sets are backed up in parallel.

When you use DFSMSdss rather than DFSMSHsm as the data mover, concurrent copy can be used and the CDSs will be validated as they are backed up. Concurrent copy allows the CDS backup to complete as soon as the journal has been backed up so that DFSMSHsm can begin processing requests before the CDS backups are completed.

BACKVOL Defaults: None.

DFSMSHsm Defaults: If you do not specify DATAMOVER, the type of data movement used is determined by the SETSYS CDSVERSIONBACKUP command subparameter DATAMOVER. If the SETSYS CDSVERSIONBACKUP command subparameter DATAMOVER has not been previously specified, the DFSMSHsm default is DATAMOVER(HSM).

Note:

1. If DATAMOVER is specified, either HSM or DSS must be specified.
2. If BACKVOL CDS specifies DATAMOVER(DSS) without BACKUPDEVICECATEGORY, and the DFSMSHsm CDS version backup environment indicates TAPE(NOPARALLEL), the BACKVOL CDS command will not be successful, and an error message will be generated.
3. Each control data set and the journal will be backed up to separate data sets whose names are generated using the values specified in the SETSYS CDSVERSIONBACKUP command subparameters of MCDSBACKUPDSN, BCDSBACKUPDSN, OCDSBACKUPDSN, and JRNLBACKUPDSN. These names are appended with a DFSMSHsm-defined final qualifier of 'Vnnnnnnn' if the datamover is DFSMSHsm, or 'Dnnnnnnn' if the datamover is DFSMSdss. If the backup fails, independent of the datamover specified, the data set name field of the failed backup will be given a final qualifier of 'Xnnnnnnn'.

NULLJOURNALONLY: Specifying that DFSMSHsm null the journal

Explanation: NULLJOURNALONLY is an optional parameter of CONTROLDATASETS specifying that DFSMSHsm null the journal without making backup copies of the control data sets.

Defaults: None.

Note:

1. If you specify this parameter, no other parameters for the BACKVOL CDS are allowed.
2. The journal must have been inhibited by a journal processing error before this parameter is specified, otherwise the BACKVOL CDS NULLJOURNALONLY command will fail.

Optional parameters for SMS-managed volumes

This topic describes the optional parameters of the BACKVOL command.

BACKUPDEVICECATEGORY: Specifying where the backup versions are to reside

Explanation: BACKUPDEVICECATEGORY is an optional parameter of VOLUME specifying the type of device, tape or DASD, that receives the backup versions of the data sets when they are backed up.

Subparameter	Explanation
TAPE	BACKUPDEVICECATEGORY(TAPE) specifies that the backup versions of the data sets on the volume reside on a tape daily backup volume when the data sets are backed up.

Subparameter	Explanation
DASD	BACKUPDEVICECATEGORY(DASD) specifies that the backup versions of the data sets on the volume reside on a DASD daily backup volume when the data sets are backed up.

Defaults: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is TAPE.

Note:

1. When you specify BACKUPDEVICECATEGORY, you can specify either TAPE or DASD, but not both.
2. The BACKUPDEVICECATEGORY parameter does not apply when you specify the DUMP parameter.

DUMP, INCREMENTAL, and TOTAL: Specifying which data sets are backed up

Explanation: DUMP, INCREMENTAL, and TOTAL are mutually exclusive, optional parameters of VOLUME specifying which data sets are to be backed up. A message is issued and the backup or dump is not processed if:

- DFSMSHsm is running in a non-SMS environment.
- The SMS-managed volume is in *initial* status (contains both SMS-and non-SMS-managed data sets while in the process of being converted to an SMS-managed volume).
- DFSMSHsm cannot determine if the specified volume is SMS-managed.

DUMP

specifies that a DFSMSdss full-volume dump be performed for a single volume. All specified subparameters process the volume and thereby override the dump classes, defined for the storage group which is associated with the volume.

Note: The DUMP subparameters DUMPClass, RETENTIONPERIOD, and STACK, are described separately.

INCREMENTAL

specifies that only those data sets that have never been backed up or have been changed since their last backup version was created are eligible for backup. If a data set is a changed data set, the frequency requirement as defined in the data set's management class must be met before DFSMSHsm backs up the data set.

TOTAL

specifies that all the data sets on the volume are to be backed up unless the data set is excluded by its management class.

Defaults: The default is INCREMENTAL.

DUMPClass: Specifying that each dump copy be targeted to a particular dump class

Explanation: DUMPClass(*class*) is an optional parameter of DUMP requesting that each dump copy be targeted to a particular dump class. At least one *class* and no more than five *classes* must be specified if DUMPClass is specified. If dump classes are not specified, they are taken from the storage group definition to which the volume belongs. If none exist for the storage group and none are specified on the command, the command fails.

Defaults: If the DUMPClass parameter is not specified, the dump classes from the storage group definition to which the volume belongs are used.

Note:

1. DUMPClass is applicable only when the DUMP parameter is used.
2. Do not specify the same dump class more than once.

- Do not specify more than one dump class if it targets cloud object storage. Creating multiple dump copies concurrently for a storage group or volume during one BACKVOL DUMP command or automatic dump session is not supported if one or more of the dump classes (copies) designate cloud object storage.

RETENTIONPERIOD: Requesting a specific retention period for the dump copy

Explanation: RETENTIONPERIOD is an optional parameter of DUMP, requesting a specific retention period for the dump copy or copies.

The list of retention periods corresponds to the list of dump classes. The dump copy targeted to the first dump class in the list is retained for the number of days specified by the first retention period in the list, and so forth.

Subparameter	Explanation
<i>days</i>	RETENTIONPERIOD(days) specifies a number of days to retain the dump copy. For <i>days</i> , substitute a decimal number from 1 to 9999 or NOLIMIT. If <i>days</i> is specified as 5, DFSMSShsm retains the dump copy for five days before expiring the dump copy.
*	RETENTIONPERIOD(*) is specified as a placeholder to denote that the retention period for the corresponding dump class specified with the DUMPCLASS parameter should be used.
NOLIMIT	RETENTIONPERIOD(NOLIMIT) indicates to DFSMSShsm to not expire the dump copy automatically.

Defaults: If fewer retention periods are listed than dump classes, the retention periods are taken from the corresponding dump class definitions.

Note:

- The BACKVOL command fails if the number of retention periods that are specified for RETENTIONPERIOD is greater than the number of classes specified for DUMPCLASS.
- RETENTIONPERIOD is applicable only if DUMPCLASS is specified.
- If DUMPCLASS is specified without RETENTIONPERIOD, the retention periods are taken from the dump classes specified.
- DFSMSShsm will not automatically delete the last and only copy of a source volume, regardless of the RETENTIONPERIOD set. To delete the dump volumes in this copy, issue the DELVOL command with the LASTCOPY option for one of the dump volumes. All of the volumes in the copy will be deleted.

STACK/MAXSTACK and MINSTACK: Specifying the maximum and minimum number of dump copies to be stacked for the specified dump classes

Explanation: STACK/MAXSTACK and MINSTACK are optional parameters of DUMP, specifying the maximum and minimum number of dump copies to stack on a dump volume in a specified dump class.

The list of stack values corresponds to the list of dump classes. A dump volume in the first specified dump class has no more dump copies than specified in the first stack value, and so on.

Dump volumes from multiple copy pools with the same dump class are stacked on the same tape volume as allowed by the values specified for MAXSTACK and MINSTACK.

Subparameter	Explanation
<i>maxv</i>	<p>STACK(<i>maxv</i>) or MAXSTACK(<i>maxv</i>) specifies the preferred maximum number of dump copies that the system should place on a tape volume for the corresponding dump class. This value overrides the value in the dump class definition. In some cases, the available MAXDUMPTASKS tasks might be under utilized if <i>maxv</i> is too large. This might occur when the number of dump copies are not large enough to fill a tape volume to the maximum number of dump copies defined for every available dump task. For <i>maxv</i>, specify a value from 1 to 255.</p> <p>If the STACK/MAXSTACK value is not specified, the value from the corresponding dump class will be used, which then might also be adjusted to be compatible with the MINSTACK value.</p>
<i>minv</i>	<p>MINSTACK(<i>minv</i>) specifies the preferred minimum number of dump copies that the system should place on a tape volume for each corresponding dump class. This value overrides the value in the dump class definitions. The minimum value is used by the system to determine the number of available MAXDUMPTASKS tasks to utilize for the dump copies. The effect is that the system might utilize more available dump tasks for the dump copies as compared to using <i>maxv</i>, but might also have less dump copies on the tape volume. The <i>minv</i> might also be used with the common queue dump group to maximize the use of available dump tasks in all the group members in the HSMplex. The system places dump copies on a tape volume until the <i>maxv</i> is reached. If a value is specified, it must not be greater than any of the <i>maxv</i> values.</p> <p>If the MINSTACK value is not specified, the value from the corresponding dump class will be used, which then might also be adjusted to be compatible with the MAXSTACK value.</p>
*	<p>STACK(*) or MAXSTACK(*) is specified as a placeholder to denote that the stack value for the corresponding dump class should be taken from that dump class definition.</p>

Defaults: If fewer stack values are listed than dump classes, stack values beyond those specified are taken from the corresponding dump class definitions.

Note:

1. The BACKVOL command fails if the number of stack values that are specified for STACK/MAXSTACK is greater than the number of dump classes that are specified for DUMPCLASS.
2. If you use BACKVOL to dump a single volume, the STACK/MAXSTACK parameter is ignored.
3. STACK/MAXSTACK is applicable only if DUMPCLASS is specified and the STACK/MAXSTACK values apply to the specified dump classes.
4. If DUMPCLASS is specified without STACK/MAXSTACK, the stack values are taken from the dump classes that are specified.
5. If DUMPCLASS or STACK/MAXSTACK are not specified, the stack values for each storage group are taken from the dump classes that are associated with that storage group.
6. For environments where the common dump queue is enabled, system affinity of dump copies and the volser order in the BACKVOL VOLUMES parameter is not applicable.
7. STACK/MAXSTACK and MINSTACK are not applicable if the specified dump class designates CLOUD.

TERMINAL: Requesting that backup or dump messages appear at the terminal

Explanation: TERMINAL is an optional parameter of VOLUME specifying that any data set backup or dump messages be sent to the system console and to either the backup activity log or the dump activity log.

Defaults: If you do not specify TERMINAL, the backup messages are sent only to the backup activity log, and the dump messages are sent to the dump activity log.

Examples of how to code the BACKVOL command for SMS-managed volumes

The following examples present different ways to code the BACKVOL command for SMS-managed volumes:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Backing up control data sets

In this example, control data sets are backed up to DASD, using DFSMSdss as the datamover.

```
BACKVOL CONTROLDATASETS(BACKUPDEVICECATEGORY(DASD) DATAMOVER(DSS))
```

Nulling the journal

In this example, the journal is nulled and the control data sets are not backed up.

```
BACKVOL CONTROLDATASETS(NULLJOURNALONLY)
```

Backing up all the data sets on an SMS-managed volume

In this example, all the data sets on an SMS-managed volume are backed up if their management class attributes allow command backup. Because you did not specify TERMINAL, the data set backup messages go only to the backup activity log.

```
BACKVOL VOLUMES(SMS025) TOTAL
```

Backing up eligible data sets on an SMS-managed volume

In this example, the eligible data sets, as defined by their management-class attributes, are backed up. The backup versions of these data sets will reside on tape. The data set backup messages appear at the system console and in the backup activity log.

```
BACKVOL VOLUMES(VOL003) INCREMENTAL TERMINAL +  
BACKUPDEVICECATEGORY(TAPE)
```

Dumping an SMS-managed volume to a specified dump class and retention period

In this example, an SMS-managed volume, SMS13, is dumped to a specific class, EXTRA, and the dump copy is assigned a retention period of 10 days.

```
BACKVOL VOLUMES(SMS13) +  
DUMP(DUMPCCLASS(EXTRA) RETENTIONPERIOD(10))
```


Dumping an SMS-managed volume to a dump class

In this example, the volume SMS13 is dumped to a dump class called OLDCLASS. The retention period is taken from the definition of class OLDCLASS.

```
BACKVOL VOLUMES(SMS13) DUMP(DUMPCCLASS(OLDCLASS))
```

In this example, the volume SMS13 is dumped to the dump classes associated with its storage group definition.

```
BACKVOL VOLUMES(SMS13) DUMP
```

In this example, five dump copies are made of volume SMS13. The retention periods for the first three dump copies are the retention periods from the corresponding dump classes. The retention period for the fourth dump copy is 60 days. The retention period for the fifth dump class is unlimited.

```
BACKVOL VOLUMES(SMS13) +  
        DUMP(DUMPCCLASS(WK,MNTH,DLY,VITREC,SPEC) +  
            RETENTIONPERIOD(*,*,*,60,NOLIMIT))
```

Backing up all defined control data sets

The following example shows how to code the BACKVOL command to back up all defined control data sets.

```
BACKVOL CONTROLDATASETS
```

Backing up data sets in several storage groups

In this example, all the supported data sets on the volumes in the storage groups: HSMONE, HSMTWO, and HSMFOUR are to be backed up by DFSMSHsm.

```
BACKVOL STORAGEGROUP(HSMONE,HSMTWO,HSMFOUR) TOTAL
```

Dumping all volumes of a storage group to a specific dump class

In this example, DFSMSHsm dumps all volumes in storage group, SGSMALL. The dump stacking and retention options are obtained from the definition of dump class, SPECIAL.

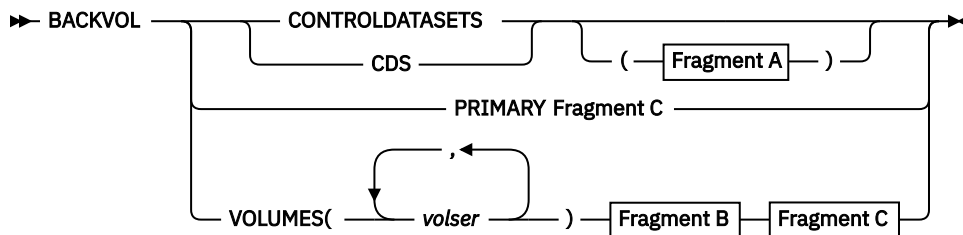
```
BACKVOL STORAGEGROUP(SGSMALL) DUMP(DUMPCCLASS(SPECIAL))
```

Backing up data sets from one or more non-SMS-managed volumes or from all primary volumes or dumping non-SMS-managed volumes

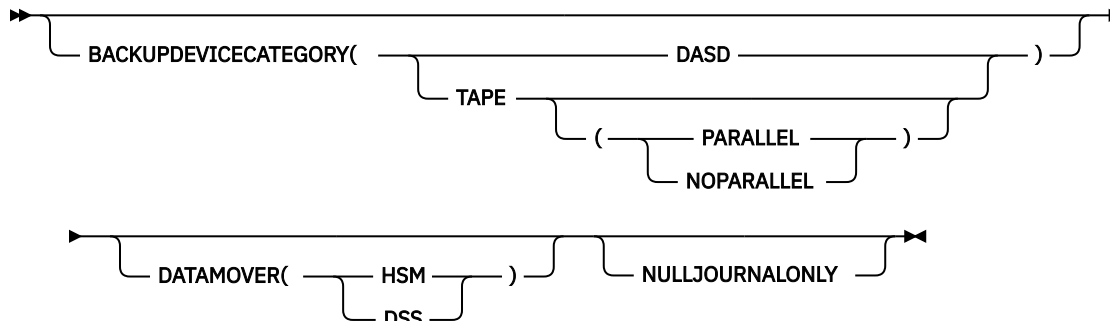
The BACKVOL command can be used to request that DFSMSHsm back up eligible data sets from specific volumes or from all primary volumes, or that it dumps all the data sets from the specified volumes. When backing up non-SMS-managed volumes, you can specify, on the command, the type of backup to be done. You can choose to back up the following:

- Data sets that have not been backed up for a specified number of days (FREQUENCY)
- Data sets that have changed since the last backup (INCREMENTAL)
- All data sets on a volume (TOTAL).

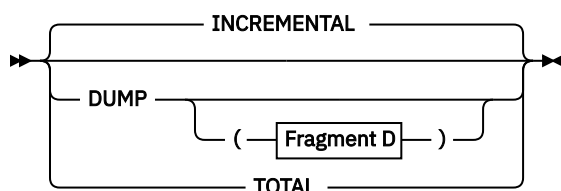
Syntax of the BACKVOL command with non-SMS-managed volumes



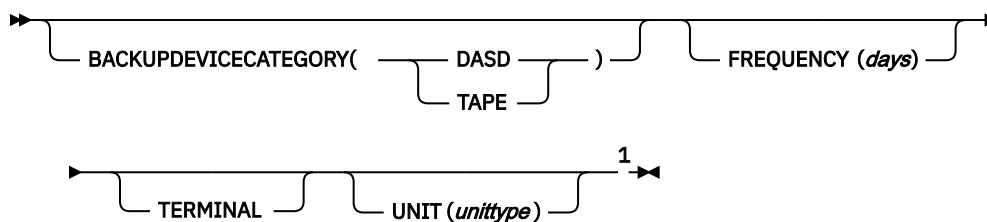
A: CONTROLDATASETS optional parameters



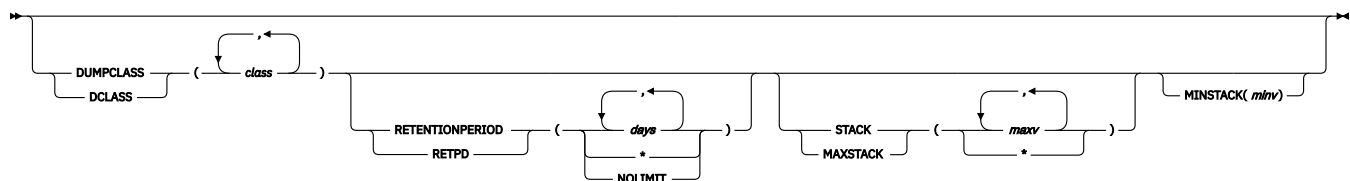
B: VOLUMES optional parameters



C: PRIMARY and VOLUMES optional parameters



D: DUMP optional parameters



Notes:

¹ The UNIT parameter is ignored if you specify the PRIMARY parameter.

Summary of parameters

The following table is a summary of the combination of parameters you can specify to back up eligible data sets, using the BACKVOL command:

Parameter	Related Parameters
BACKUPDEVICECATEGORY	None
CONTROLDATASETS	None
FREQUENCY	INCREMENTAL
INCREMENTAL	None
PRIMARY	None
TOTAL	None
TERMINAL	None
UNIT	VOLUMES(<i>volser</i>)
VOLUMES(<i>volser</i>)	UNIT (volume not managed or owned by DFSMSHsm)

The following table is a summary of the combination of parameters you can specify to dump all data sets from a volume, using the BACKVOL command:

Parameter	Related Parameters
DUMP	VOLUMES(<i>volser</i>)
DUMPClass	DUMP
RETENTIONPERIOD	DUMPClass
STACK	DUMPClass
TERMINAL	None
UNIT	VOLUMES(<i>volser</i>)
VOLUMES(<i>volser</i>)	UNIT (volume not managed or owned by DFSMSHsm)

Required parameters for non-SMS-managed volumes or all primary volumes

This topic describes the required parameters of the BACKVOL command.

CONTROLDATASETS: Specifying the control data sets to be backed up

The description of this parameter is the same in both the SMS-environment and in the non-SMS-environment, refer to [“CONTROLDATASETS: Specifying the control data sets to be backed up”](#) on page 781 for a description of this parameter.

PRIMARY and VOLUMES: Specifying the volumes to be backed up or dumped

Explanation: PRIMARY and VOLUMES(*volser*,...) are mutually exclusive, required parameters specifying whether DFSMSHsm backs up the eligible data sets on the specified volumes or all primary volumes, or whether DFSMSHsm dumps all data sets on the specified volumes.

PRIMARY specifies that DFSMSHsm back up the eligible data sets on all primary volumes, not just those with the primary-volume attribute of automatic backup.

VOLUMES(*volser*,...) specifies that DFSMSHsm is to back up the eligible data sets on the specified volumes or dump all data sets on those volumes. For *volser*, substitute the volume serial number of the volumes to be backed up or dumped. (A volume serial number appearing more than once in the list is backed up or dumped only once.)

Defaults: None.

Note:

1. A volume list can include both SMS-managed volumes and non-SMS-managed volumes.
2. If you specify the VOLUMES parameter with more than 80 volumes, DFSMSHsm processes only the first 80 volumes.
3. If DUMP is specified with DUMPCCLASS, and at least one of the specified dump classes (or the optional STACK subparameter) specifies a stacking value greater than one, DFSMSHsm attempts to stack the dumps of the volumes in the order specified.
4. All stacking to a given dump volume is done while that tape is mounted. DFSMSHsm always selects empty dump volumes as targets for DUMP.
5. When you use BACKVOL to dump one or more specific volumes, DFSMSHsm treats all volumes being dumped as having affinity to the host issuing the command which means that they are all processed on the single host where the command is submitted.

The following notes apply if you have not specified the optional parameter DUMP with the BACKVOL command:

- The FREQUENCY and INCREMENTAL parameters set up the eligibility criteria for data sets.
- You are not required to specify UNIT if you specify a volume managed or owned by DFSMSHsm. If, however, you specify a volume not managed by DFSMSHsm, you must also specify the UNIT parameter. If the list of volumes contains any not managed by DFSMSHsm, each volume must be capable of being allocated on the *unittype* specified with UNIT.
- If you request that DFSMSHsm back up more than one volume (you specified the PRIMARY parameter or the VOLUMES parameter by specifying a list of volume serial numbers), DFSMSHsm starts a separate volume backup task for each primary volume until it reaches the maximum number of backup tasks specified with the MAXBACKUPTASKS parameter of the SETSYS command.

The following notes apply only if you have specified the optional parameter DUMP with the BACKVOL command:

- The DUMP optional parameter is allowed only if you specify VOLUMES(*volser*,...), and *volser* can be for any DASD volume, including DFSMSHsm-owned volumes.
- You are not required to specify UNIT if you specify a volume managed or owned by DFSMSHsm.

Optional parameters for non-SMS-managed control data sets

This topic describes the optional parameters of the BACKVOL command.

BACKUPDEVICECATEGORY: Specifying the backup device for the control data sets and journal

The description of this parameter is the same in both the SMS-environment and in the non-SMS-environment, refer to [“BACKUPDEVICECATEGORY: Specifying the backup device for the control data sets and journal” on page 783](#) for a description of this parameter.

DATAMOVER: Specifying the type of data movement used when backing up the control data sets

The description of this parameter is the same in both the SMS-environment and in the non-SMS-environment, refer to [“DATAMOVER: Specifying the type of data movement used when backing up the control data sets” on page 783](#) for a description of this parameter.

NULLJOURNALONLY: Specifying that DFSMSHsm null the journal

The description of this parameter is the same in both the SMS-environment and in the non-SMS-environment, refer to [“NULLJOURNALONLY: Specifying that DFSMSHsm null the journal” on page 784](#) for a description of this parameter.

Optional parameters for non-SMS-managed volumes or all primary volumes

This topic describes the optional parameters of the BACKVOL command.

BACKUPDEVICECATEGORY: Specifying where the backup versions reside

Explanation: BACKUPDEVICECATEGORY is an optional parameter of PRIMARY and VOLUME specifying the type of device, tape or DASD, that receives the backup versions of the data sets when they are backed up. When using the BACKUPDEVICECATEGORY parameter, one of the following subparameters must be specified:

Subparameter	Explanation
TAPE	BACKUPDEVICECATEGORY(TAPE) specifies that the backup versions of the data sets on the volume reside on a tape daily backup volume when the data sets are backed up.
DASD	BACKUPDEVICECATEGORY(DASD) specifies that the backup versions of the data sets on the volume reside on a DASD daily backup volume when the data sets are backed up.

Defaults: If you do not specify BACKUPDEVICECATEGORY and the volume has been added with the ADDVOL command, the default is the backup device category specified in the ADDVOL command.

Note:

1. When you specify BACKUPDEVICECATEGORY, you can specify either TAPE or DASD, but not both.
2. The BACKUPDEVICECATEGORY parameter does not apply when you specify the DUMP parameter.
3. The BACKUPDEVICECATEGORY parameter will override the UNIT parameter when the specified backup device category does not match the specified unit type. For example, if the specified backup device category is DASD and the specified unit type is 3490 (tape) the volumes will be backed up to DASD.
4. If you are attempting to backup to tape, ensure that you have previously issued the SETSYS CDSVERSIONBACKUP (BACKUPDEVICECATEGORY(TAPE (UNITNAME(*unit*))) command to properly configure the CDSVERSIONBACKUP parameter table.

DUMP, INCREMENTAL, and TOTAL: Specifying which data sets are backed up

Explanation: DUMP, INCREMENTAL, and TOTAL are mutually exclusive optional parameters of the VOLUME parameter specifying which data sets are to be backed up.

DUMP

specifies that a DFSMSdss full-volume dump be performed for a single volume. All subparameters specified are used to process the volume, overriding the dump classes associated with the volume.

Note: The DUMP subparameters DUMPCCLASS, RETENTIONPERIOD, and STACK are described separately.

INCREMENTAL

specifies that only those data sets that have never been backed up or have been changed since their last backup version was created are eligible for backup. If a data set is a changed data set, the frequency requirement as defined in the data set's management class must be met before DFSMSHsm backs up the data set.

TOTAL

specifies that all the data sets on the volume or volumes are backed up by DFSMSHsm if it supports the data set organization. DFSMSHsm backs up all other data sets on the volume, except for data sets that have been excluded by an ALTERDS or HALTERDS VERSIONS(0) command, not just changed data sets that have met the frequency of backup.

Defaults: The default is INCREMENTAL.

DUMPCCLASS: Specifying that each dump copy be targeted to a particular dump class

Explanation: DUMPCCLASS(*class*) is an optional parameter of DUMP requesting that each dump copy be targeted to a particular dump class. Dump classes, *class*, must be predefined with the DUMPCCLASS optional parameter of the DEFINE command.

DUMPCCLASS *is required* unless the specified volume is a primary volume with a dump class specified for it with its ADDVOL command. At least one *class* and no more than five *classes* must be specified if DUMPCCLASS is specified.

Defaults: If the DUMPCCLASS parameter is not specified and the volume is a primary volume with DUMPCCLASS specified with its ADDVOL command, the dump classes specified with the ADDVOL command are used.

Note:

1. DUMPCCLASS is applicable only when the DUMP parameter is used.
2. Do not specify the same dump class more than once.
3. Do not specify more than one dump class if it targets cloud object storage. Creating multiple dump copies concurrently for a storage group or volume during one BACKVOL DUMP command or automatic dump session is not supported if one or more of the dump classes (copies) designate cloud object storage.

FREQUENCY: Specifying the frequency of backup for data sets

Explanation: FREQUENCY(*days*) is an optional parameter of VOLUME specifying the number of days required between backup versions of a data set. DFSMSHsm does not back up the data set unless the specified number of days has elapsed since the last back up of the data set. For *days*, substitute a decimal number from 0 to 999. For example, if *days* is specified as 5, only those data sets that have not been backed up in the last five days are eligible for backup. When you specify 0, DFSMSHsm backs up changed data sets no matter how recently they were backed up.

Defaults: If you do not specify the FREQUENCY parameter with the BACKVOL command or have not specified the FREQUENCY parameter with an ALTERDS command, the value for FREQUENCY previously specified on the SETSYS command or the DFSMSHsm default for FREQUENCY is used.

Note:

1. If you specify FREQUENCY when it does not apply, DFSMSHsm ignores it.
2. The frequency specified with this command does not override the frequency specified with the ALTERDS command; however, it overrides the frequency specified with the SETSYS command and the DFSMSHsm default. DFSMSHsm determines the frequency of backup in the following order:
 - a. ALTERDS
 - b. BACKVOL
 - c. SETSYS
 - d. DFSMSHsm default

RETENTIONPERIOD: Requesting a specific retention period for the dump copy

The description of this parameter is the same in both the SMS-environment and in the non-SMS-environment, refer to [“RETENTIONPERIOD: Requesting a specific retention period for the dump copy”](#) on page 786 for a description of this parameter.

STACK/MAXSTACK and MINSTACK: Specifying the maximum and minimum number of dump copies to be stacked for the specified dump classes

The description of this parameter is the same in both the SMS-environment and in the non-SMS-environment, refer to [“STACK/MAXSTACK and MINSTACK: Specifying the maximum and minimum number of dump copies to be stacked for the specified dump classes” on page 786](#) for a description of this parameter.

TERMINAL: Requesting that backup or dump messages appear at the terminal

The description of this parameter is the same in both the SMS-environment and in the non-SMS-environment, refer to [“TERMINAL: Requesting that backup or dump messages appear at the terminal” on page 788](#) for a description of this parameter.

UNIT: Specifying the type of device

Explanation: UNIT(*unittype*) is an optional parameter of PRIMARY and VOLUME specifying the type of unit where the volume being backed up can be allocated.

For *unittype*, substitute the type of unit where the volume can be allocated.

The following are valid types of DASD units:

- 3380
- 3390
- 9345

Defaults: None.

Note:

1. You need not specify UNIT if you specify a volume that DFSMSHsm manages or owns. If, however, you specify a volume DFSMSHsm does not manage, you must also specify the UNIT parameter.
2. The BACKUPDEVICECATEGORY parameter will override the UNIT parameter when the specified backup device category does not match the specified unit type. For example, if the specified backup device category is DASD and the specified unit type is 3490 (tape) the volumes will be backed up to DASD.
3. If you specify the UNIT parameter for a volume that DFSMSHsm owns or manages, DFSMSHsm uses the specified unit type rather than the one in its volume record.

Examples of how to code the BACKVOL command for non-SMS-managed volumes

The following examples present different ways to code the BACKVOL command for non-SMS-managed volumes:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Backing up control data sets

In this example, control data sets are backed up to DASD, using DFSMSdss as the datamover.

```
BACKVOL CONTROLDATASETS(BACKUPDEVICECATEGORY(DASD) DATAMOVER(DSS))
```

The following example shows how to code the BACKVOL command to back up all defined control data sets according to options specified by the SETSYS CDSVERSIONBACKUP parameter or its defaults.


```
BACKVOL CONTROLDATASETS
```

Nulling the journal

In this example, the journal is nulled and the control data sets are not backed up.

```
BACKVOL CONTROLDATASETS(NULLJOURNALONLY)
```

Backing up all the data sets on a volume

In this example, all the data sets on a volume not managed by DFSMSHsm are backed up. Because you have not specified **TERMINAL**, the data set backup messages go only to the backup activity log.

```
BACKVOL VOLUMES(PUB025) UNIT(3390) TOTAL
```

Backing up eligible data sets on DFSMSHsm-managed volumes

In this example, only two kinds of data sets are backed up:

- Data sets that have never been backed up.
- Data sets that have been changed since the last backup version was created and that meet the specified frequency of two days since the last backup version was created.

The backup versions of these data sets will reside on tape. The data set backup messages appear at the console and in the backup activity log.

```
BACKVOL VOLUMES(VOL003,VOL004,VOL005) INCREMENTAL FREQUENCY(2) +
        BACKUPDEVICECATEGORY(TAPE) TERMINAL
```

Dumping a primary volume to a dump class

In this example, a primary volume, PRIM13, is dumped to a specific class, **EXTRA**, and assigns the dump copy a retention period of 10 days.

```
BACKVOL VOLUMES(PRIM13) +
        DUMP(DUMPCLASS(EXTRA) RETENTIONPERIOD(10))
```

In this example, the volume PRIM13 is dumped to a dump class called **OLDCLASS**. The retention period is the retention period of **OLDCLASS**.

```
BACKVOL VOLUMES(PRIM13) DUMP(DUMPCLASS(OLDCLASS))
```

In this example, the volume PRIM13 is a primary volume dumped to the classes associated with it by the previous **ADDVOL** command with the **AUTODUMP** parameter.

```
ADDVOL PRIM13 PRIMARY(AD(WEEKLY,MONTHLY)) +
        UNIT(3390)
```

```
BACKVOL VOLUMES(PRIM13) DUMP
```

In this example, five dump copies are made of volume PRIM13. The retention periods for the first three dump copies are the retention periods for the corresponding dump classes. The retention period for the fourth dump copy is 60 days. The fifth dump class has an unlimited retention period.

```
BACKVOL VOLUMES(PRIM13) +
        DUMP(DUMPCLASS(WK,MNTH,DLY,VITREC,SPEC) +
        RETENTIONPERIOD(*,*,*,60,NOLIMIT))
```


Chapter 37. BDELETE command: Deleting backup versions of a data set

The BDELETE command deletes all the backup versions of a specific data set or UNIX file or deletes specific backup versions by version number or date. You can specify multiple data set or UNIX file names with the BDELETE command.

If you have cataloged and uncataloged data sets with the same name, DFSMSHsm deletes the backup versions of either the cataloged or uncataloged data sets, not the backup versions of both. See note “2” on page 799.

To delete retired versions, you must use the BDELETE command and must specify the version number or date and time of the retired version.

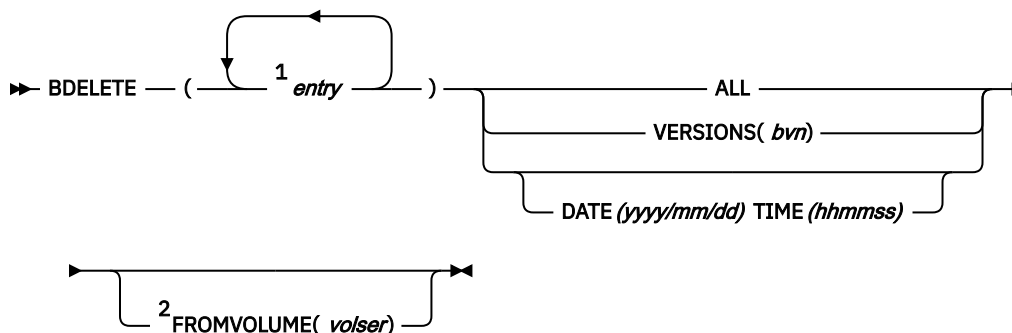
To delete all backup versions of a data set or UNIX file, both active and retained, you must use the ALL parameter on the BDELETE command.

If all backup versions of a data set or UNIX file are deleted, parameters previously specified with the ALTERDS command are no longer in effect.

When the BDELETE command is submitted from a batch environment, no messages indicating the success of this function are issued. You can use the LIST command to obtain information on existing backup versions.

When backup versions of a UNIX file are deleted, the last backup date in the UNIX file metadata is set to the most recent version in the DFSMSHsm inventory. If all backup versions are deleted, the last backup date is reset to January 1, 1970 UTC.

Syntax of the BDELETE command



Notes:

¹ *entry* may be either *dsname* or '*pathname*'.

² The FROMVOLUME parameter does not apply to SMS-managed data sets, cataloged data sets, or UNIX files.

Required parameters of the BDELETE command

This topic describes the required parameters of the BDELETE command.

dsname: Specifying the name of the data set whose backup versions are to be deleted

Explanation: (*dsname*) is a required positional parameter specifying the name of the data set or list of data set names whose backup versions you want to delete. Each data set name you specify must be fully

qualified. For *dsname*, substitute the fully qualified name or list of names of the data sets whose backup versions are to be deleted.

Defaults: None.

Note: Because *dsname* is a required positional parameter, you must specify it immediately after BDELETE.

pathname: Specifying the name of the UNIX file whose backup versions are to be deleted

Explanation: (*'pathname'*) is a required positional parameter specifying the name of the UNIX file or list of UNIX file names whose backup versions that you want to delete. UNIX path names must be an absolute path, starting with the forward slash character, enclosed in single quotation marks. If the UNIX file name has a single quotation mark character as part of the file name, specify two single quotation marks in the *'pathname'* instead of one.

Some special characters that are allowed by the UNIX cannot be processed by the HBDELETE command. Instead, the UNIX shell command, *hbdelete*, can be used to delete backup versions of files with these characters by using the backslash character. For more information, see [unix-hdel](#) in *z/OS DFSMSHsm Managing Your Own Data*. Note the *-o* option for the UNIX *hbdelete* program is equivalent to issuing the BDELETE storage administrator command.

Defaults: None.

Notes:

1. Because *'pathname'* is a required positional parameter, you must specify it immediately after BDELETE.
2. A mixture of UNIX file names and data set names can be specified on one BDELETE request.

ALL, DATE and TIME, and VERSIONS: Specifying which backup versions to delete

Explanation: ALL, DATE and TIME, and VERSIONS are mutually exclusive parameters that specify which backup versions are to be deleted. Specify one (and only one) of the following:

ALL: Deleting all backup versions

Explanation: ALL specifies that all active and retained backup versions of a data set or UNIX file are to be deleted.

Defaults: None.

Note: If you want to delete a retired version, you must specify either VERSIONS or DATE and TIME. A retired version is the backup version DFSMSHsm creates before it scratches a data set during data set retirement. DFSMSHsm does not automatically delete retired versions.

DATE and TIME: Specifying the date and time of the backup version to delete

Explanation: DATE(*yyyy/mm/dd*) is used together to delete any active or retained backup version. They must be used to delete NEWNAME backup copies. For *yyyy/mm/dd*, substitute the date when the backup version to be deleted was created. For *hhmmss*, substitute the time when the backup version to be deleted was created.

DATE and TIME apply to SMS and non-SMS-managed data sets and UNIX files.

Defaults: None.

Restrictions

- DATE and TIME must be specified together.

- FROMVOLUME is mutually exclusive with DATE and TIME.

VERSIONS: Specifying the backup versions to delete

Explanation: VERSIONS(*bvn*) specifies which of the active backup versions of a data set or UNIX file to delete. You can specify multiple version numbers. For *bvn*, substitute the one-to-three-digit decimal number of a particular backup version to be deleted. You can get the backup version number by issuing a LIST command and specifying the data set or UNIX file name and the BACKUPCONTROLDATASET parameter.

A list of backup version numbers applies to the backup versions of any data sets or UNIX file you specify. For example, assume that you specify the following command:

```
BDELETE (VLS6492.PQR.CLIST, VLS6493.STU.CLIST) +
VERSIONS(115,118,120)
```

DFSMSHsm deletes backup version numbers 115, 118, and 120 for data sets VLS6492.PQR.CLIST and VLS6493.STU.CLIST.

This command fails if VERSIONS(0) is specified. Instead, DATE and TIME must be used to uniquely identify a retained backup copy or a backup copy that was created with the NEWNAME keyword.

Note:

1. If you want to delete a retired version, you must specify VERSIONS or DATE and TIME. A retired version is the backup version DFSMSHsm creates before it scratches a data set during data set retirement. DFSMSHsm does not automatically delete retired versions.
2. You cannot use VERSIONS to delete an individual retained backup version.

Defaults: None.

Optional parameters of the BDELETE command

This topic describes the optional parameters of the BDELETE command.

FROMVOLUME: Specifying the volume where the data set resided when the backup versions were created

Explanation: FROMVOLUME(*volser*) is an optional parameter specifying the volume where the uncataloged data set or data sets resided when DFSMSHsm backed up the data sets. You specify this parameter only if the data set was uncataloged. For *volser*, substitute the serial number of the volume where the uncataloged data set or data sets resided when the backup versions were created.

Defaults: None.

Note:

1. The data sets you specified should have been either all cataloged or all uncataloged when DFSMSHsm created the backup versions. If the data sets were all uncataloged, they should have been on the same volume. If the backup versions were not created from the same volume, the backup versions of the data sets are not deleted.
2. If you have cataloged and uncataloged data sets with the same name, DFSMSHsm deletes the backup versions of the cataloged data set unless you specify the FROMVOLUME parameter. If you specify the FROMVOLUME parameter, only the backup version of the uncataloged data set is deleted.
3. FROMVOLUME does not apply to SMS-managed data sets.
4. FROMVOLUME does not apply to cataloged data sets.

Examples of how to code the BDELETE command

The following examples present different ways to code the BDELETE command:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Deleting specific backup versions of two SMS-managed data sets

In this example, the specified backup versions of the two SMS-managed data sets are to be deleted.

```
BDELETE AMS6607.PDS.CLIST VERSIONS(3,8)
BDELETE AMS6609.PDS.CLIST VERSIONS(16,17)
```

Deleting all backup versions of two cataloged data sets

In this example, all the backup versions of the two data sets are deleted if all the data sets except for the retired version have been cataloged prior to backup.

```
BDELETE (AMS6607.A.CLIST, AMS6608.B.CLIST) ALL
```

Deleting particular backup versions of an uncataloged data set

In this example, backup versions numbered 9, 11, 13, and 15 of an uncataloged data set backed up from volume VOL001 are deleted.

```
BDELETE AMS6607.UNC.CLIST FROMVOLUME(VOL001) +
VERSIONS(9,11,13,15)
```


Chapter 38. CANCEL command: Canceling a queued or active DFSMSHsm request

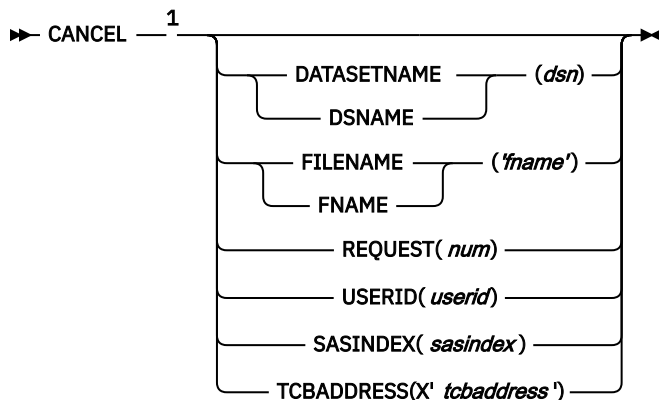
The CANCEL command cancels an active or queued DFSMSHsm data movement request. You can cancel a specific request, all requests for a particular data set or UNIX file, or all requests for a particular user.

Active requests can be canceled by using either the SASINDEX or TCBADDRESS parameters. Queued requests can be canceled by using the DATASETNAME, FILENAME, REQUEST, or USERID parameters.

Note:

1. The CANCEL command can be controlled using a RACF FACILITY class.
2. The CANCEL and QUERY commands have the highest processing priority. Therefore, the CANCEL and QUERY commands cannot be canceled.
3. In a CVQ environment, recall requests can be canceled only from the host that originated the request.
4. Canceling requests running in the CDQ or CVQ will complete for queued requests that originate on the host that issues the CANCEL command for a request number or user ID that matches the parameters on the CANCEL command. Canceling active requests with the TCBADDRESS must be done on the host that is processing the request. A single CANCEL by USERID will affect requests in both common queues. You can use the QUERY COMMONQUEUE(RECOVER) command or QUERY COMMONQUEUE(DUMP) command to determine the requests running in their respective common queues. These commands will identify the host ID of the originating host for the request and determines the host from which a corresponding CANCEL should be issued. For a host that is connected to a CDQ or CVQ, and is *not* the master scheduler, the local queue, and the master scheduler member group queue are searched. For a host that is the master scheduler, only the local queue is searched.

Syntax of the CANCEL command



Notes:

- ¹ The CANCEL command has no required parameters. Issuing the CANCEL command with no parameters results in no action taken and no output.

Related information

[“Syntax of the CANCEL command” on page 801](#)

Optional parameters of the CANCEL command

This topic describes the optional parameters of the CANCEL command.

DATASETNAME: Canceling a queued request by data set name

Explanation: DATASETNAME is a mutually exclusive, optional parameter, specifying which queued DFSMSHsm requests to cancel. All requests for the specified data set name will be canceled.

The following variable is required when using the DATASETNAME parameter:

dsn

Is the fully-qualified data set name for which all requests should be canceled.

Defaults: None.

FILENAME: Canceling a queued request by UNIX file name

Explanation: FILENAME is a mutually exclusive, optional parameter, specifying which queued DFSMSHsm requests to cancel. All requests for the specified UNIX file name will be canceled.

The following variable is required when if you are using the FILENAME parameter. If the CANCEL command is issued from the console, *fname* will need to be enclosed in single quotation marks.

fname

For *fname*, substitute the absolute path and file name for the file for which you want to cancel requests.

If the CANCEL command is issued from the console, *fname* will need to be enclosed in single quotation marks.

Defaults: None.

REQUEST: Canceling a queued request by request number

Explanation: REQUEST is a mutually exclusive, optional parameter, specifying which queued DFSMSHsm request to cancel. The request corresponding to the specified request number will be canceled.

The following variable is required when using the REQUEST parameter:

num

Is the DFSMSHsm request number of the request you want to cancel. Use the QUERY REQUEST command to determine the request number of a queued request.

REQUEST(*num*) can also be used to cancel an entire group of file backups that were requested via a partial path or UNIX file name.

Note:

1. If you issue the CANCEL REQUEST command for requests created by the following commands, all queued requests associated with the specified request number are canceled. This is because these commands spin-off multiple requests with the same request number.
 - A single volume recovery from incrementals
 - BACKVOL volumes list
 - BACKVOL STORAGEGROUP
 - FRBACKUP DUMP
 - FRBACKUP DUMPPONLY
 - FRRECOV COPYPOOL FROMDUMP
 - FRRECOV DSNNAME

Defaults: None.

USERID: Canceling a queued request by user ID

Explanation: USERID is a mutually exclusive, optional parameter, specifying which queued DFSMSHsm requests to cancel. All requests by the specified user ID will be canceled.

The following variable is required when using the USERID parameter:

userid

Is an 8-character alphanumeric user ID for which all requests can be canceled.

Defaults: None.

SASINDEX: Canceling an active request by ABARS secondary address space

Explanation: SASINDEX is a mutually exclusive, optional parameter, specifying which active DFSMSHsm request to cancel. The request corresponding to the specified ABARS secondary address space will be canceled.

The following variable is required when using the SASINDEX parameter:

sasindex

Is the integer value between 1 and 64 returned by the QUERY ACTIVE(TCBADDRESS) command for which the corresponding request should be canceled.

Defaults: None.

TCBADDRESS: Canceling an active request by TCB address

Explanation: TCBADDRESS is a mutually exclusive, optional parameter, specifying which active DFSMSHsm request to cancel. The request corresponding to the specified TCB address will be canceled.

The following variable is required when using the TCBADDRESS parameter:

X'tcbaddress'

Is the 8 digit hexadecimal value returned by the QUERY ACTIVE(TCBADDRESS) command for which the corresponding request should be canceled.

Note: This command should be used with caution.

CANCEL TCBADDR(*nnnnnn*) is not like the CANCEL USERID or CANCEL REQ commands. This command cancels a task that is doing work where a user suspects DFSMSHsm is hung. It should only be used if a problem is suspected and a user does not want to have to cancel DFSMSHsm.

Users canceling a TCBADDR must remember that they are canceling a DFSMSHsm task that is actively doing work. The task is abnormally terminated with an ABEND7D0 which will then cause abnormal termination routines to get control. Cascading errors after the ABEND 7D0 may occur.

Users should issue a HOLD for ALL functions prior to canceling the task. The HOLD will prevent other tasks from being started until the cancel has completed.

Defaults: None.

Examples of how to code the CANCEL command

The following examples present how to code the CANCEL command:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Canceling all requests for a specified user ID

In this example, all requests submitted by the specified *userid* are canceled.

```
CANCEL USERID(L541563)
```

Canceling all requests issued from the system console

In this example, all requests submitted from the system console are canceled. Note that ****OPER*** in the example below is a pseudo *userid* representing the system operator.


```
CANCEL USERID(**0PER*)
```

Canceling all requests for a specified data set name

In this example, all requests for a specified data set name are canceled.

```
CANCEL DATASETNAME(L541563.PSU.N.F230EP03.DSET1)
```

Canceling a particular request

In this example, request number 0015 is canceled.

```
CANCEL REQUEST(15)
```

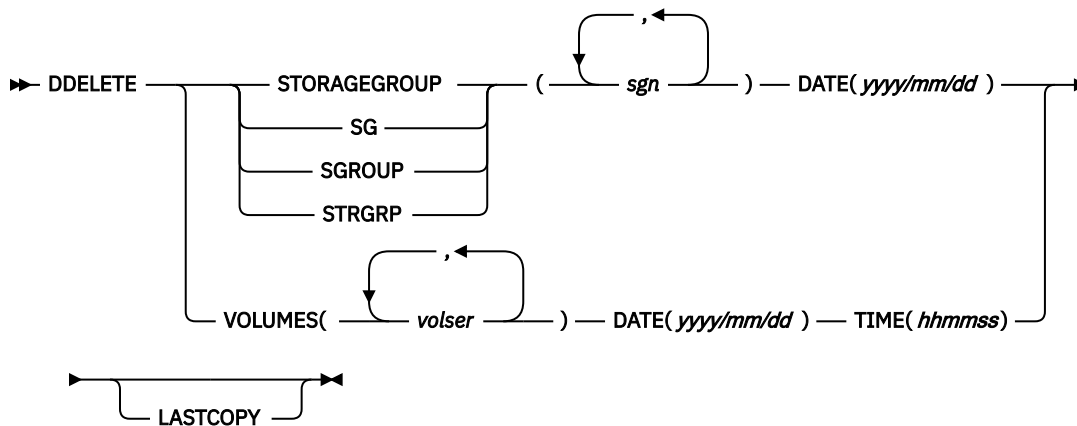

Chapter 39. DDELETE Command – Deleting cloud dump copies

The DDELETE command deletes dump copies residing in the cloud object storage that are associated with the specified storage groups or volumes. The command is intended for occasional cloud dump copy deletion and is not designed for bulk deletion. Use of automatic dump expiration to manage dumps and empty dump container deletion is recommended.

Note:

- DDELETE cannot be used to delete TAPE dump copies or copy pool dump copies. Use the FRDELETE command to delete copy pool dump copies.
- DDELETE does not delete the only valid dump copy of a volume unless the LASTCOPY parameter is specified.
- Do not issue DDELETE command when automatic dump is active.

Syntax of the DDELETE command



Required parameters of the DDELETE command

This topic describes the required parameter of the DDELETE command. The required STORAGEGROUP and VOLUMES parameters are mutually exclusive.

STORAGEGROUP: Specifying the storage group whose dump copies are to be deleted

Explanation: STORAGEGROUP(*sgn*,...) is a required parameter specifying that DFSMSHsm deletes the associated dump copies residing in the cloud object storage. For *sgn*, substitute the names of the storage groups containing volumes from where the dump copies are created. (A storage group name appearing more than one time in the list is a processed once.)

Defaults: None.

Notes:

1. If DFSMSHsm is running in a non-SMS environment, a message is issued, and the dump copies are not processed.
2. If you specify STORAGEGROUP without *sgn*, the command fails. If you specify STORAGEGROUP with more than 30 storage groups, the first 30 are processed.

3. If a batch job issues the WAIT DDELETE STORAGEGROUP command, DFSMSHsm processes the command, but does not ensure that the action completes before returning control to the caller. As a result, it is possible for the batch job to complete with return code zero before all of the dump copies are processed.

VOLUMES: Specifying the volumes whose dump copies are to be deleted

Explanation: VOLUMES(*volser*,...) is a required parameter specifying that DFSMSHsm is to delete the associated dump copies that reside in the cloud object storage. For *volser*, substitute the serial number of the volumes from which the dump copies were created. (A volume serial number appearing more than once in the list is processed only once.)

Defaults: None.

Notes:

1. A volume list can include both SMS-managed volumes and non-SMS-managed volumes.
2. If you specify the VOLUMES parameter with more than 80 volumes, DFSMSHsm processes only the first 80 volumes.

DATE and TIME: Specifying the particular cloud dump copy to delete

Explanation: DATE and TIME(*yyyy/mm/dd*) are used to delete the cloud dump copies of specified STORAGEGROUP volumes. DATE(*yyyy/mm/dd*) and TIME(*hhmmss*) are used together to delete the cloud dump copies of specified VOLUMES.

DATE(*yyyy/mm/dd*)

For *yyyy/mm/dd*, substitute the date when the dump copy to be deleted is created. The oldest generation that is made on the date is deleted.

TIME(*hhmmss*)

For *hhmmss*, substitute the time when the dump copy to be deleted is created. DATE and TIME apply to SMS and non-SMS managed volumes.

Defaults: None.

Restrictions: DATE and TIME are required when VOLUMES parameter is specified. TIME cannot be specified when STORAGEGROUP parameter is specified.

Usage notes: The DATE and TIME parameter formats are different from the formats in the dump copy name. The LIST PRIMARYVOLUME(*volser*) ALLDUMPS BCDS command can be used to display information on dump copies associated with the specified source volume. The DUMPED DATE *yy/mm/dd* and TIME *hh:mm:ss* in the LIST output can be used to determine the date and time values to use in the DDELETE command.

Related reading: For more information about LIST PRIMARYVOLUME ALLDUMPS BCDS command, see Chapter 76 Using LIST command.

Optional parameters of the DDELETE command

This topic describes the optional parameter of the DDELETE command.

LASTCOPY: Invalidating the only valid dump copy of a source volume

Explanation: LASTCOPY is an optional parameter that must be specified to delete a dump copy that is part of the only valid dump copy of a source volume. LASTCOPY is ignored if it is not part of the only copy.

Defaults: None.

Examples of how to code the DDELETE command

The following examples present different ways to code the DDELETE command:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Deleting dump copies of volumes in a storage group

In this example, DFSMSHsm processes dump copies of volumes in the specified storage group. DFSMSHsm attempts to delete the oldest dump copy made on the specified date for the volumes in the storage group.

```
DDELETE STORAGEGROUP(sg1) DATE(2021/10/26)
```

Deleting a dump copy of a specified volume

In this example, DFSMSHsm processes dump copies of the specified volume. DFSMSHsm attempts to delete the dump copy made on the specified date and time. LASTCOPY allows DFSMSHsm to delete the only valid dump copy of a source volume.

```
DDELETE VOLUME(src001) DATE(2021/10/26) TIME(020041) LASTCOPY
```

Chapter 40. DEFINE command: Defining control structures for use by DFSMSHsm

The DEFINE command allows you to specify the following control structures that are used by DFSMSHsm:

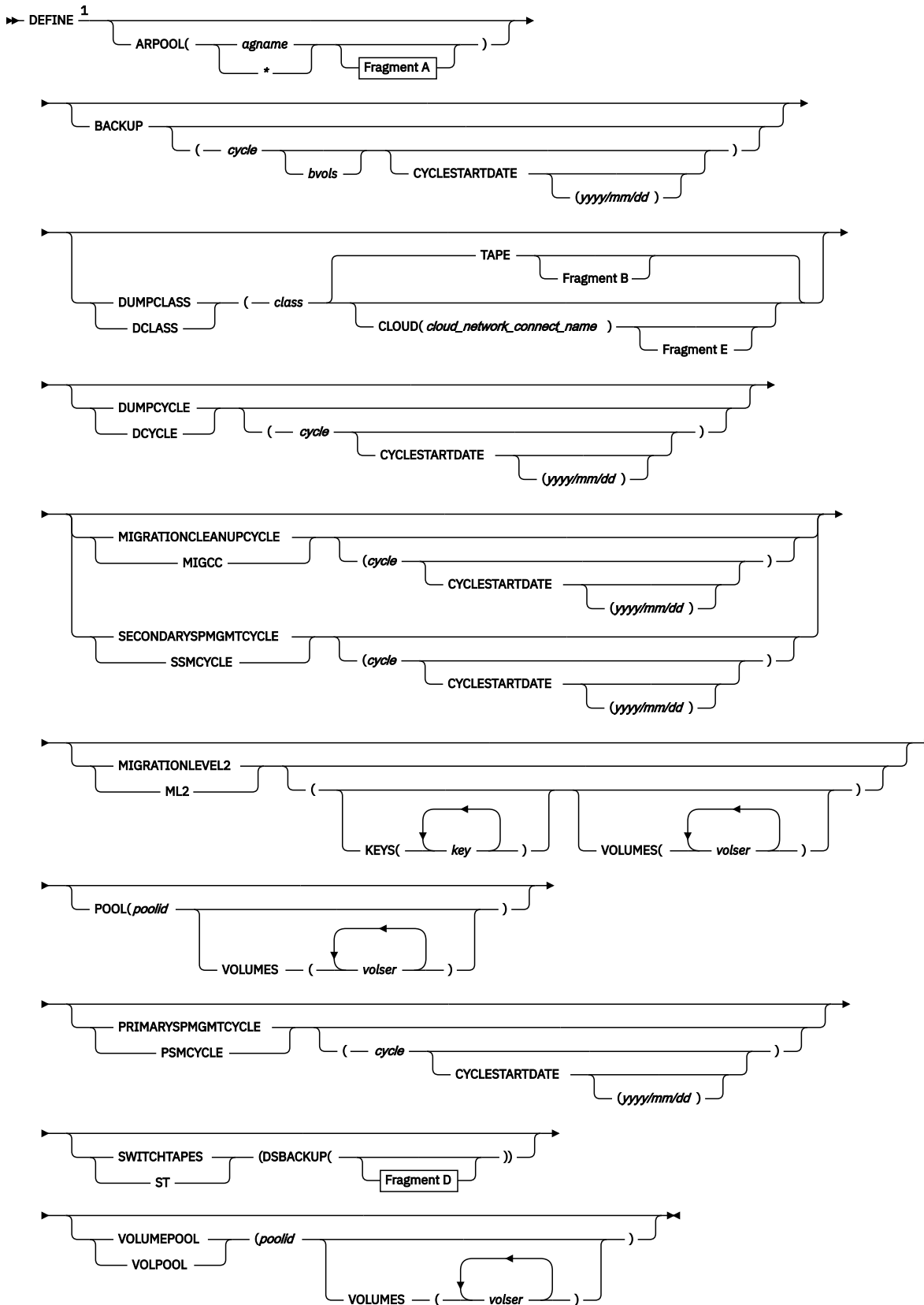
- Backup cycle, start date for the backup cycle, and the number of daily backup volumes to be used for each day in the backup cycle
- Dump class attributes
- Dump cycle and start date for the dump cycle
- Automatic primary space management cycle
- Automatic secondary space management cycle
- Key ranges for DASD migration level 2 volumes
- Pool of volumes to which non-SMS-managed data sets with the same set of initial characters are recalled
- Pool of volumes, that together can be used as a target for recalling a data set migrated from one of the volumes
- Pool of volumes to which backed-up aggregated data sets will be restored during aggregate recovery
- When to release a data set backup tape that is mounted to process data set backup commands and the status of the released tape.

When you use JES3, you must use the ADDVOL command to add primary volumes to DFSMSHsm before you can specify the volumes with the POOL or the VOLUMEPOOL parameter of the DEFINE command.

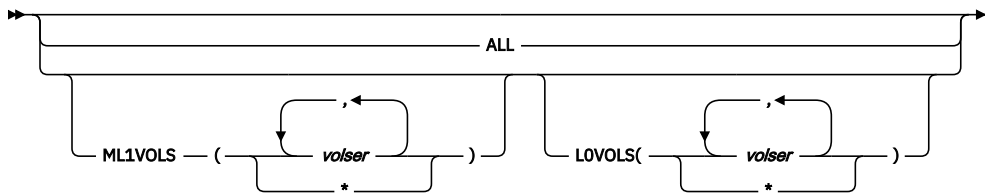
If an SMS-managed volume is specified as part of a data set pool, a message is issued indicating that the volume has not been added to the data set pool. Processing continues for the DEFINE command. If, however, only SMS-managed volumes are specified for the definition of the data set pool, the data set pool definition is rejected, and the DEFINE command fails.

With the exception of the ARPOOL, POOL and VOLUMEPOOL parameters, you do not have to redefine any optional parameters each time you start DFSMSHsm.

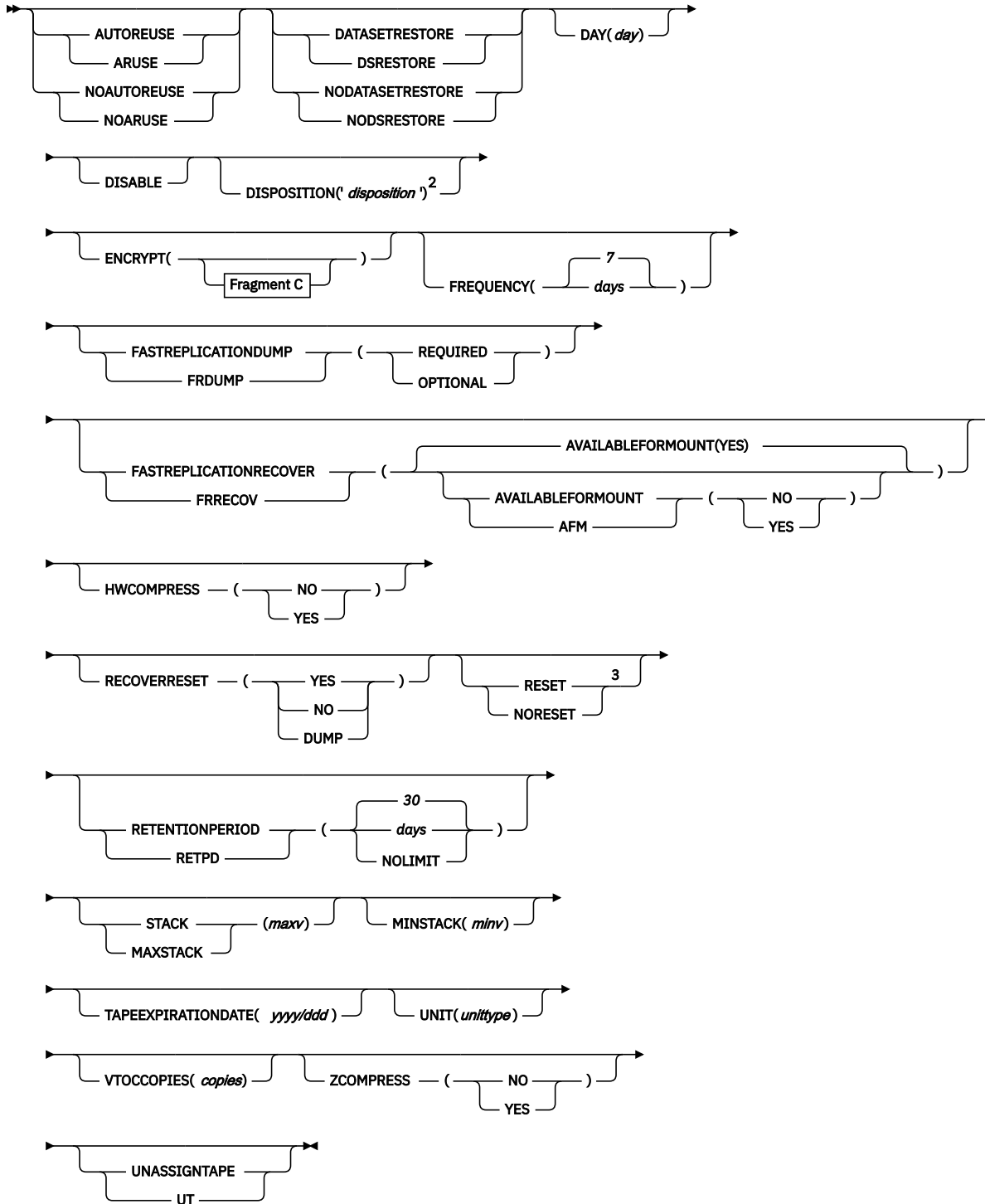
Syntax of the DEFINE command



A: ARPOOL optional parameters

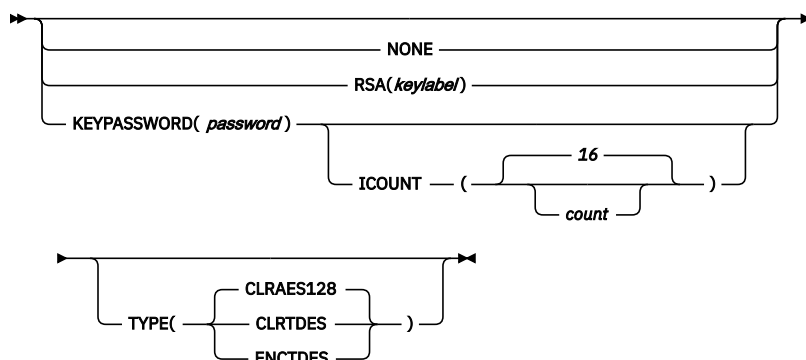


B: DUMPCCLASS optional parameters

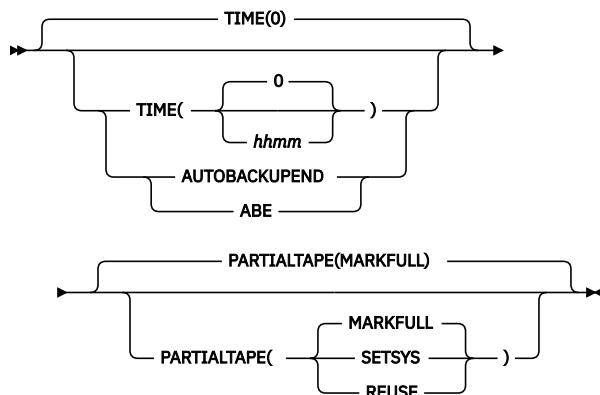


C: ENCRYPT optional parameters

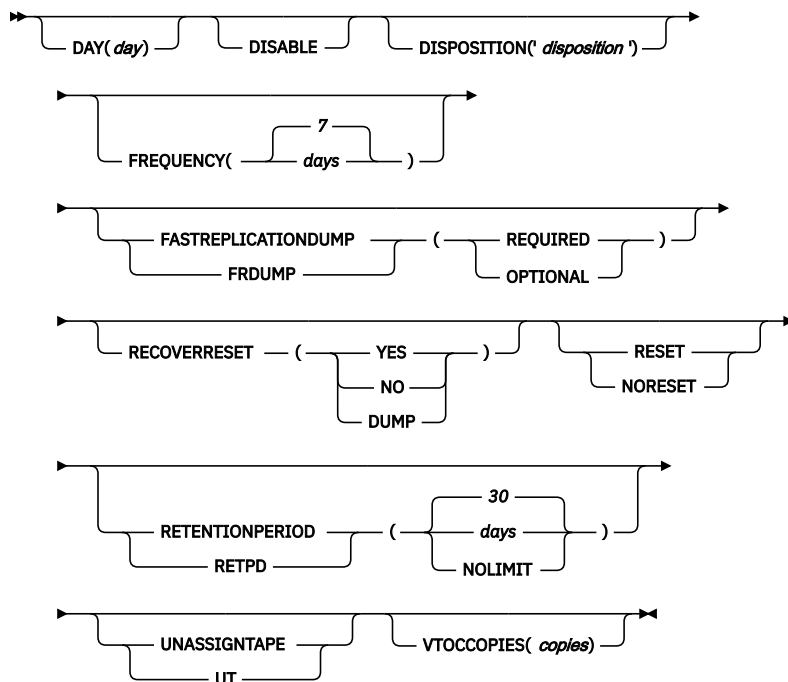
DEFINE command for DFSMSHsm



D: SWITCHTAPES optional parameters



E: DUMPCCLASS optional parameters with CLOUD parameter specified



Notes:

- ¹ You must specify at least one parameter with each DEFINE command.
- ² The single quotation marks (' ') are required if *disposition* is more than one word or if it contains a special character.
- ³ Do not specify the RESET parameter with the FASTREPLICATIONDUMP parameter because the RESET parameter does not apply to copy pool processing.

Optional parameters of the DEFINE command

This topic describes the optional parameters of the DEFINE command.

ARPOOL: Identifying the volumes for aggregate recovery

Explanation: **ARPOOL** is an optional parameter specifying a pool of volumes to be used as target volumes during an aggregate recovery.

Subparameter	Explanation
<i>agname</i>	ARPOOL(<i>agname</i>) specifies the name of the aggregate group. For <i>agname</i> substitute an aggregate group name that has been defined. The defined pool will be used for recovery of that aggregate group's data sets. If no <i>agname</i> is specified, the command fails.
*	ARPOOL(*) specifies that the volumes are assigned to a general pool of aggregate recovery volumes. The general pool of volumes is used when recovering an aggregate group that does not have a specific ARPOOL defined.

You can use the following optional subparameters with the ARPOOL parameter.

Subparameter	Explanation
ALL	ALL specifies that all currently mounted ADDVOL ML1 and primary volumes be defined as the respective volume pools for aggregate recovery. Specifying the ALL parameter is equivalent to specifying the LOVOLS(*) and ML1VOLS(*) parameters. ALL is mutually exclusive with the LOVOLS and ML1VOLS parameters.
LOVOLS	LOVOLS(<i>volser</i> ...) specifies the list of volumes used as target volumes during recovery of nonmigrated non-SMS-managed data sets. If you specify a parameter of LOVOLS(*), all currently mounted ADDVOL primary volumes (up to 59) will be used as the primary volume pool for aggregate recovery. LOVOLS is mutually exclusive with the ALL subparameter. You can use LOVOLS with ML1VOLS.
ML1VOLS	ML1VOLS(<i>volser</i> ...) specifies the list of volumes as target volumes during recovery of migrated data sets to ML1 volumes. If you specify a parameter of ML1VOLS(*), all currently mounted ADDVOL ML1 volumes will be used as the ML1 volume pool for aggregate recovery. ML1VOLS is mutually exclusive with the ALL subparameter. You can use ML1VOLS with LOVOLS.

Defaults: None.

Note:

1. If the DEFINE ARPOOL is issued without the ALL, ML1VOLS, and LOVOLS parameters, the aggregate recovery pool specified by *agname* is deleted. If an aggregate group name of * is issued without the ALL, M1VOLS, and LOVOLS parameters, the general pool of aggregate recovery volumes is deleted.
2. If a DEFINE ARPOOL command has not been issued when an ARECOVER command is issued, DFSMSHsm uses all currently mounted ADDVOL primary and ML1 volumes as a temporary ARPOOL.
3. The LOVOLS in an ARPOOL are used to recover non-SMS-managed data sets. In an SMS environment, allocation is directed by the ACS routines. LOVOLS are used only in an SMS environment if the ACS routines do not direct allocation to any other volume.
4. You cannot define the same volume as both LOVOLS and ML1VOLS within a pool or between pools.

5. You cannot define SMS-managed volumes to an ARPOOL.
6. A volume previously added to DFSMSHsm with the PRIMARY parameter of the ADDVOL command can subsequently be defined to the ARPOOL only by using the LOVOLS optional parameter.
7. A volume previously added to DFSMSHsm with the ML1 parameter of the ADDVOL command can subsequently be defined to the ARPOOL only by using the ML1VOLS optional parameter.
8. Since the DEFINE ARPOOL command is only in effect for the duration of DFSMSHsm startup, it is recommended that you add this command to the DFSMSHsm parmlib member (ARCCMDxx) on the system that is the target of the aggregate recovery.
9. Do not allow the DEFINE ARPOOL to default to the current ADDVOL volumes for all aggregates. If you allow this, all aggregate recoveries will choose the same volume as target volumes, causing contention for those volumes. To reduce volume contention and to improve performance, it is more beneficial to define a unique set of volumes to each aggregate being recovered, especially if more than 1 ABARS address space is active simultaneously.

BACKUP: Specifying the backup cycle

Explanation: **BACKUP** is an optional parameter specifying the days in a backup cycle when automatic backup is done and the number of backup volumes DFSMSHsm uses each day that volume backup runs.

BACKUP(*cycle*): For *cycle*, substitute a string of alphabetic Ys and Ns. Each Y represents a day in the cycle when you want automatic backup done and each N represents a day in the cycle when you do not want automatic backup done. You can represent up to 31 days with the Ys and Ns. When you specify BACKUP, the current day is the first day of a cycle unless you specify the cycle start date.

BACKUP(*cycle bvol*s): For *bvol*s, substitute a decimal number from 1 to 9999 to represent the number of DASD backup volumes DFSMSHsm assigns to a given day in the backup cycle before initiating spill or cleanup processing on full DASD daily backup volumes. This value is called the *daily limit*.

CYCLESTARTDATE can be used with this parameter and is listed alphabetically after this discussion.

Defaults: If you specify BACKUP without *cycle*, the backup cycle defaults to a seven-day cycle with automatic backup every day. If you do not specify a minimum number of backup volumes with *bvol*, the number defaults to one.

Note:

1. You must specify BACKUP if you want to use DFSMSHsm backup.
2. For DASD backup, ensure that the number of backup cycle days is greater than the number of SETSYS VERSIONS. (For example, if you specify SETSYS VERSIONS(3), then BACKUP cycle days must be 4 or greater.)

CYCLESTARTDATE: Specifying the start date of a cycle

Explanation: **CYCLESTARTDATE** is an optional subparameter specifying the start date or reset date of a cycle. It can be used with the following parameters:

- DEFINE BACKUP
- DEFINE DUMPCYCLE
- DEFINE MIGRATIONCLEANUPCYCLE
- DEFINE PRIMARYSPMGMTCYCLE
- DEFINE SECONDARYSPMGMTCYCLE

A leading zero is not required for a one-digit month or day. For a start date, specify *yyyy/mm/dd*.

Defaults: If CYCLESTARTDATE is specified without a start date or if CYCLESTARTDATE is not specified at all, then:

- If a date is not already stored, DFSMSHsm sets the start date to the current date.

- Or, if a date is already stored and the cycle length is changed, DFSMSHsm sets the start date to the current date.
- Or, if a date is already stored and the cycle length is unchanged, DFSMSHsm does not change the cycle start date.

Note:

1. The cycle start date is saved between DFSMSHsm startups.
2. The start date specified with the CYCLESTARTDATE parameter cannot be a date in the future. The date must be a date prior to or equal to the date when the DEFINE command is issued.
3. Understanding the difference between cycle days and calendar days is important. The first cycle day, DAY 1, begins at the start time of the function. The second cycle day, DAY 2, begins when this start time has again been reached, regardless of when the new calendar day begins. Third and subsequent days follow the same pattern. For instance, if the cycle start time for a function is 1100 (11 a.m.) on Monday, DAY 1 is from 11 a.m. on Monday to 10:59 a.m. on Tuesday. DAY 2 begins after 11 a.m. on Tuesday and continues until 10:59 a.m. on Wednesday.
4. If you specify CYCLESTARTDATE, you must also specify *cycle*; otherwise, the command will fail.

DUMPCLASS: Adding or changing a volume dump class

Explanation: DUMPCLASS is an optional parameter specifying a volume dump class to be added or changed.

Subparameter	Explanation
class	DUMPCLASS(class): For <i>class</i> , substitute from one to eight characters for the class name to be defined. The first character must be an alphabetic or national (# @ \$) character. The remaining characters can be alphabetic, national, or numeric characters. Only one class can be specified on a single DEFINE command.

Tape Library Relationship: A unit name can be used for filtering in the ACS routines. If you specify the unittype for this command, it will be sent to the ACS routines in SMS for the DEFINE function.

If the class has not been defined previously, a new backup control data set (BCDS) record is created describing the dump class. Unspecified parameters will have their DFSMSHsm defaults supplied.

A previously defined class will have its existing BCDS record. Unspecified parameters remain unchanged from the previous definition, unless either the DATASETRESTORE or AUTOREUSE subparameters, or both, have been specified and the RETENTIONPERIOD is specified as NOLIMIT. In this case, the DATASETRESTORE and AUTOREUSE subparameters are set to NODATASETRESTORE and NOAUTOREUSE.

For SMS-managed volumes, you must use the DUMPCLASS parameter to define dump classes referred to by storage groups.

The subparameters of the DUMPCLASS parameter are listed in alphabetical order and can be found at the following locations:

Subparameter	For more information, see
AUTOREUSE NOAUTOREUSE	“DUMPCLASS(AUTOREUSE and NOAUTOREUSE): Specifies whether DUMP volumes are automatically available for reuse when invalidated” on page 816
CLOUD TAPE	“DUMPCLASS(CLOUD TAPE): Specifying where the dump copies to reside” on page 817
DATASETRESTORE NODATASETRESTORE	“DUMPCLASS(DATASETRESTORE and NODATASETRESTORE): Data set restore is allowed from a full-volume dump copy in this dump class” on page 818

Subparameter	For more information, see
DAY	“DUMPCLASS(DAY): Specifying the target day of the dump cycle for this class” on page 819
DISABLE	“DUMPCLASS(DISABLE): Specifying the volume dump class and making it unavailable as a target of any full-volume dump” on page 820
DISPOSITION	“DUMPCLASS(DISPOSITION): Specifying the disposition of dump volumes” on page 820
ENCRYPT	“DUMPCLASS(ENCRYPT): Specifying whether to encrypt the dump data with host based encryption” on page 820
FASTREPLICATIONDUMP	“DUMPCLASS(FASTREPLICATIONDUMP): Specifying whether a dump class is required or optional” on page 821
FASTREPLICATIONRECOVER	“DUMPCLASS(FASTREPLICATIONRECOVER): Specifying whether a dump class copy is immediately available to be mounted for recovery” on page 822
FREQUENCY	“DUMPCLASS(FREQUENCY): Specifying the minimum number of days that must elapse between volume dumps of a volume to this class” on page 822
HWCOMPRESS	“DUMPCLASS(HWCOMPRESS): Specifying whether to compress the data to be encrypted” on page 823
RECOVERRESET	“DUMPCLASS(RECOVERRESET): Specifying whether to reset the data-set-changed indicator in the VTOC entry” on page 823
RESET NORESET	“DUMPCLASS(RESET and NORESET): Specifying whether DFSMSdss resets the change bit for each data set following a full-volume dump” on page 824
RETENTIONPERIOD	“DUMPCLASS(RETENTIONPERIOD): Specifying the enforcement of retention periods for the dump copies in this class” on page 824
STACK	“DUMPCLASS(STACK/MAXSTACK) and DUMPCLASS(MINSTACK): Specifying the maximum and minimum number of dump copies that are stacked for this dump class” on page 825
TAPEEXPIRATIONDATE	“DUMPCLASS(TAPEEXPIRATIONDATE): Specifying a unique expiration date for all dump copies created in a dump” on page 827
UNIT	“DUMPCLASS(UNIT): Specifying a default tape unit name” on page 827
VTOCCOPIES	“DUMPCLASS(VTOCCOPIES): Specifying the number of dump VTOC copy data sets to keep for dump copies created in this class” on page 828
ZCOMPRESS	“DUMPCLASS(ZCOMPRESS): Specifying whether to use zEDC Services on the dump data” on page 829

DUMPCLASS(AUTOREUSE and NOAUTOREUSE): Specifies whether DUMP volumes are automatically available for reuse when invalidated

Explanation: **AUTOREUSE** and **NOAUTOREUSE** are mutually exclusive, optional subparameters of **DUMPCLASS**, specifying whether dump volumes with retention periods in the defined class are automatically available for reuse when automatically invalidated.

Subparameter	Explanation
AUTOREUSE	DUMPClass(AUTOREUSE) specifies that when DFSMSHsm initiates the invalidation of a dump copy, the dump volumes are automatically made available for reuse.
NOAUTOREUSE	DUMPClass(NOAUTOREUSE) specifies that when DFSMSHsm initiates the invalidation of a dump copy, the dump volumes are marked unavailable for selection by DFSMSHsm. The ADDVOL command can be used to identify to DFSMSHsm that the volume is available for reuse. The use of NOAUTOREUSE is dependent upon using the SETSYS TAPEDELETION option of HSM TAPE for the dump category.

Defaults: If AUTOREUSE or NOAUTOREUSE are specified, the default is NOAUTOREUSE when a dump class is being defined for the first time.

If a dump class is already defined and certain characteristics are being changed, and if AUTOREUSE or NOAUTOREUSE are specified, then the previous specification remains unchanged unless the retention period is NOLIMIT. In this case, the dump class is set for NOAUTOREUSE.

Note:

1. You can use the DELVOL command with the PURGE parameter to invalidate the only valid dump copy.
2. NOAUTOREUSE must be specified if volumes for this dump class are to leave the confines of the data processing center and the SETSYS TAPEDELETION option of HSM TAPE has been specified for this dump class. The NOAUTOREUSE parameter prevents mount requests for volumes that are not immediately available. These volumes must be identified to DFSMSHsm by issuing the ADDVOL command, but not until they are returned from their off-site storage location. If AUTOREUSE is used in this environment, then DFSMSHsm might call for a specific mount for output of a recently released tape before the tape is again physically present in the data processing center.
3. The disposition of tapes automatically made available for reuse depends on the TAPEDELETION parameter of the SETSYS command. If you are using the TAPEDELETION option of SCRATCH, then the tape is released from DFSMSHsm when invalidated and DFSMSHsm does not know of it to consider it for subsequent use.
4. If you use NOLIMIT, you can use only the NOAUTOREUSE subparameter.
5. AUTOREUSE and NOAUTOREUSE are not applicable when CLOUD subparameter is also designated.

DUMPClass(CLOUD | TAPE): Specifying where the dump copies to reside

Explanation: **CLOUD(*cloud_network_connection_name*) | TAPE** is an optional parameter of DUMPClass specifying the device category where the dump copies are to be stored.

Subparameter	Explanation
CLOUD(<i>cloud_network_connection_name</i>)	DUMPClass(CLOUD(<i>cloud_network_connection_name</i>)) specifies the associated volumes are to be dumped directly to the specified cloud object storage. The <i>cloud_network_connection_name</i> is the name of the SMS network connection construct or the CDA provider file. If the specified name does not match what is in the SMS construct or the CDA provider file, the DEFINE DUMPClass fails. To remove DUMPClass(CLOUD) designation for the dump class, ensure all associated dump copies have been deleted, then specify DUMPClass(TAPE) .
TAPE	DUMPClass(TAPE) specifies that the associated primary volumes are to be dumped to tape volumes.

Defaults: If you do not specify this parameter, the default is TAPE for a new dump class, or the previous dump device category for an existing dump class.

The following DEFINE DUMPCLASS parameters are not applicable when the dump copy is to reside in the cloud object storage.

AUTOREUSE | NOAUTOREUSE

These settings apply to dump tape volumes.

DATASETRESTORE | NODATASETRESTORE

Neither parameter settings are applicable to dump copies residing in cloud storage. Data set recovery from full volume dump copy residing in cloud object storage is not supported at this time.

ENCRYPTION

These settings are not applicable in transparent cloud tiering.

FRRECOV(AFM(NO | YES))

These settings apply to dump tape volumes.

HWCOMPRESS(NO | YES)

These settings are not applicable in transparent cloud tiering.

STACK | MAXSTACK, MINSTACK

These settings apply to dump tape volumes.

TAPEEXPIRATIONDATE

These settings apply to dump tape volumes.

UNIT(*unitname*)

These settings apply to dump tape volumes.

When the CLOUD(*cloud_network_connection_name*) parameter is specified to an existing dump class, DFSMSHsm will do the following:

- Fail the DEFINE DUMPCLASS command if one or more dump tape volumes assigned to the class contained part of a valid dump copy. Any dump volumes containing valid data in the class remain unchanged. Any dump volumes that did not contain part of a valid dump copy may become unassigned.
- Unspecified *applicable* parameters remain unchanged from the previous definition.
- Unspecified *inapplicable* parameters are ignored or reset to a default value compatible to the CLOUD(*cloud_network_connection_name*) option.

DUMPCLASS(DATASETRESTORE and NODATASETRESTORE): Data set restore is allowed from a full-volume dump copy in this dump class

Explanation: DATASETRESTORE and NODATASETRESTORE are mutually exclusive, optional subparameters of DUMPCLASS, specifying whether to allow a physical data set restore from a full-volume dump copy in this dump class. This support applies to the RECOVER or HRECOVER commands that are issued for a single data set.

Subparameter	Explanation
DATASETRESTORE	DUMPCLASS(DATASETRESTORE) specifies that physical restores of individual data sets from dump volumes in this class are allowed. The dump volumes must remain available to satisfy mount requests. A physical data set restore requires the mounting of one or more tapes to complete the request. DFSMSHsm or DFSMSdss do not keep track of exactly where the data set is on the dump tapes. A dump VTOC copy data set is needed for data set restores.
NODATASETRESTORE	DUMPCLASS(NODATASETRESTORE) specifies that dump volumes in this class are not available to do individual physical data set restores. All data set recoveries are from incremental backup versions unless a dump volume is explicitly requested on the RECOVER command.

Defaults: If neither DATASETRESTORE nor NODATASETRESTORE is specified, the default is NODATASETRESTORE when a dump class is being defined for the first time.

If a dump class is already defined and certain characteristics are changed, and if neither DATASETRESTORE nor NODATASETRESTORE is specified, the previous specification remains unchanged unless the retention period is NOLIMIT. In this case, the dump class is set for NODATASETRESTORE.

Note:

1. If DATASETRESTORE is selected and the data set resides or has resided on a volume that was dumped to this class, DFSMSHsm determines whether the most current backup has been done by the incremental backup process or by the full-volume process. When the most current version is on a dump volume, DFSMSDss performs a physical data set restore from the copy. When the most current version is an incremental backup version or if the backup version cannot be determined, DFSMSHsm performs a data set recovery of an incremental version. If there are no records in BCDS and there is no catalog entry for the data set on the volume from which the dump copy has been made, the recovery fails unless you specify the RECOVER command with the FROMDUMP parameter.
2. If NODATASETRESTORE is selected and the request was explicitly from a dump copy (FROMDUMP keyword), DFSMSHsm fails the recovery request unless the dump volume was specified (DUMPVOLUME(valid)).
3. If NODATASETRESTORE is selected and the request was not explicitly from a dump copy, DFSMSHsm always performs a data set recovery of an incremental backup version.
4. A RETENTIONPERIOD(NOLIMIT) for the class overrides the DATASETRESTORE option and prevents physical data set restores from a dump volume in this class.
5. If you use NOLIMIT, you can use only the NODATASETRESTORE subparameter.
6. Neither DATASETRESTORE nor NODATASETRESTORE are applicable to data set recovery from a copy pool.
7. AUTOREUSE or NOAUTOREUSE is not applicable to data set recovery from a CLOUD dump class.

DUMPCCLASS(DAY): Specifying the target day of the dump cycle for this class

Explanation: **DAY** is an optional subparameter of DUMPCCLASS specifying the target day of the dump cycle for this class. DAY restricts that this class be targeted during automatic dump processing only on the specified day of the cycle.

DUMPCCLASS(DAY(*day*)): For *day*, substitute a decimal number to specify the target day of the dump cycle. A value of zero indicates that a previous association with a day in the dump cycle should be removed. Any other numeric value must be from one to a value less than or equal to the number of days specified in the dump cycle. *day* must be a 'Y' day specified in the DUMPCYCLE parameter. If a frequency is also specified, the frequency must be satisfied before the class is chosen. Full-volume dumps requested by command, however, are processed independently of the DAY specified.

DAY can be used instead of the FREQUENCY parameter, if you desire to dump to this class every cycle repeat. DAY allows a subset of the primary volumes to be dumped on one day and another subset to be dumped on another day. For example, if the primary volume subsets are defined by two dump classes and the dump classes are identical except for the DAY parameter, one subset of primary volumes can be dumped on one day and the other subset of primary volumes can be dumped on another day. The primary volumes can retain the same frequency and retention characteristics.

DUMPCCLASS Defaults: If the DAY parameter is not specified for the DUMPCCLASS, determination of whether to dump a volume is based on other parameters. If the FREQUENCY parameter criterion is met, a volume will be dumped on each 'Y' day specified in the DEFINE DUMPCYCLE.

Note: The FREQUENCY parameter defaults to seven days. However, in some cases, you may want to specify FREQUENCY(1). If for some reason autodump does not run during its normally scheduled time and manual dumps are requested later, and if the DAY specified in this parameter occurs sooner in the cycle than the FREQUENCY days specified, autodump does not run on the day specified. This problem can be avoided by specifying FREQUENCY(1).

DUMPCLASS(DISABLE): Specifying the volume dump class and making it unavailable as a target of any full-volume dump

Explanation: **DISABLE** is an optional subparameter of DUMPCLASS specifying that a dump class be unavailable as the target of any full-volume dumps. When you specify **DISABLE**, DFSMSHsm automatically unassigns any dump volumes that are empty and currently added to this class. The unassigned dump volumes are available for selection to any other class that uses the same unit type. Information about valid dumps remains in the BCDS, but no further dumps to this class are allowed, and no more dump volumes can be added to this class. The definition of the dump class also remains in the BCDS and the dump class definition will be shown in the LIST DUMPCLASS output.

Defaults: None.

Note:

1. You can enable a disabled dump class by issuing a DEFINE command without specifying the **DISABLE** subparameter.
2. Any other parameters specified on the same command are ignored.
3. Use the LIST command to find any volumes that are currently assigned to a particular class.

DUMPCLASS(DISPOSITION): Specifying the disposition of dump volumes

Explanation: **DISPOSITION** is an optional subparameter of DUMPCLASS specifying the disposition of the dump volumes after they have received part of a dump copy.

DUMPCLASS(DISPOSITION('disposition')): For *disposition*, substitute a character string from 1 to 20 characters describing the intended disposition of the dump volumes. The *disposition* is inserted into message ARC0637I, which is issued for each dump copy successfully created for a source volume. If *disposition* is not specified, message ARC0637I is sent to the dump activity log, but not to the system console.

Defaults: None.

Restriction: The single quotation marks (' ') are required if *disposition* is more than one word or if it contains a special character.

DUMPCLASS(ENCRYPT): Specifying whether to encrypt the dump data with host based encryption

Explanation: **ENCRYPT** is an optional parameter that allows you to specify whether the data to be dumped will be encrypted with host based encryption. Indicate the method of host based encryption through the **NONE**, **RSA**, or **KEYPASSWORD** parameters. To have data encrypted, you must specify either **RSA** or **PASSWORD** (but not both) in the dump class. Do not use the **ENCRYPT** parameter to specify tape hardware encryption. You can use the SMS data class to specify tape hardware encryption.

The following are parameters of the **ENCRYPT** parameter:

NONE

specifies that data encryption is not performed for this dump class. This option overrides the encryption settings, if any, in the existing dump class.

RSA

specifies the RSA key label to be used when secure cryptographic hardware is used to encrypt the data to be dumped. *keylabel* identifies the key label that is to be used to encrypt the data. The key label is 1-64 characters long, and the first character must be an alphabetic or special character.

KEYPASSWORD

specifies that the data is to be encrypted without secure cryptographic hardware. The password, which is used to generate a key for the data encryption, is 8-32 characters long.

ICOUNT

specifies the number of hash iterations to be performed on the password when the PKCS #12 algorithm is used to generate the key. For the iteration count (*count*), specify an integer between 1 and 10000.

TYPE

specifies the type of encryption to be performed on the data.

CLRAES128

encrypts the dumped data with a clear 128-bit AES key.

CLRTDES

encrypts the dumped data with a clear, triple-length DES key

ENCTDES

encrypts the dumped data with a secure triple-length DES key.

Defaults

- If you specify the KEYPASSWORD keyword without specifying ICOUNT (and ICOUNT was not previously set in the dump class definition), the default value for ICOUNT is 16.
- If you do not specify the TYPE keyword (and TYPE was not previously set in the dump class definition), CLRAES128 encryption is used by default.

Note:

1. Before using the DUMPCLASS(ENCRYPT) parameter, review the topic about considerations for host-based encryption in [z/OS DFSMSdss Storage Administration](#).
2. The ICSF address space must be started successfully, regardless of the processor you are running, to use the DUMPCLASS(ENCRYPT) parameter.
3. The set of dump classes that constitute a dump generation (that is, the set of dump classes specified on the BACKVOL command or in the storage group definition) must have the same encryption and HWCOMPRESS settings, or the dump operation will fail.
4. If you specify the KEYPASSWORD keyword, you cannot specify the ENCTDES parameter with the TYPE keyword.

DUMPCLASS(FASTREPLICATIONDUMP): Specifying whether a dump class is required or optional

Explanation: FASTREPLICATIONDUMP is an optional keyword and is applicable only to dump classes associated with copy pools.

Subparameter	Explanation
REQUIRED	DUMPCLASS(FASTREPLICATIONDUMP(REQUIRED)) indicates that every volume in a copy pool must have a successful dump copy to represent this dump class before the DASD backup version that is being dumped is replaced with a more recent DASD backup copy. After a DASD backup copy is replaced, you can no longer attempt to create dump copies from it.
OPTIONAL	DUMPCLASS(FASTREPLICATIONDUMP(OPTIONAL)) indicates that even though every volume in the copy pool might not have a successful dump copy to represent this dump class, if an FRBACKUP request is submitted to replace the DASD backup copy, the command is allowed to proceed.

Example: In a particular environment, assume that two dump classes are defined to create two dump copies. One dump class is named ONSITE and the other OFFSITE. The dump copies created for ONSITE are the primary dump copies used for recovery and have the specification of FASTREPLICATIONDUMP(REQUIRED). The dump copies for dump class OFFSITE are secondary dump copies and have the specification of FASTREPLICATIONDUMP(OPTIONAL).

The FASTREPLICATIONDUMP keyword is mutually exclusive with the RESET keyword. Any attempts to define a dump class with both RESET and FASTREPLICATIONDUMP will fail.

DUMPCCLASS(FASTREPLICATIONRECOVER): Specifying whether a dump class copy is immediately available to be mounted for recovery

Explanation: **DUMPCCLASS(FASTREPLICATIONRECOVER)** is an optional keyword. When specified, the AVAILABLEFORMOUNT parameter must also be specified as YES or NO, as follows:

AVAILABLEFORMOUNT(YES)

Specifies that a dump class copy is available for recovery and does not need to be specified on the FRRECOV command to be used.

AVAILABLEFORMOUNT(NO)

Specifies that a dump class copy is not available for immediate recovery and must be specifically specified on the FRRECOV command to be used.

If multiple dump copies are created for a backup version and AVAILABLEFORMOUNT(YES) is specified for more than one of the dump classes, the first dump copy found with YES specified will be used.

If AVAILABLEFORMOUNT(NO) is specified for all dump classes, you must specify DUMPCCLASS or DUMPVOLUME on the FRRECOV command, otherwise the command will fail. The DUMPCCLASS and DUMPVOLUME keywords of the FRRECOV command override the DUMPCCLASS FASTREPLICATIONRECOVER setting.

The FASTREPLICATIONRECOVER dump class setting applies to tape processing at the data set and volume level.

Defaults: If FASTREPLICATIONRECOVER is not specified for a dump class, the default value is AVAILABLEFORMOUNT(YES).

DUMPCCLASS(FREQUENCY): Specifying the minimum number of days that must elapse between volume dumps of a volume to this class

Explanation: **FREQUENCY** is an optional subparameter of DUMPCCLASS specifying the minimum number of days that must elapse from the last volume dump of the volume to this class before the volume can be automatically dumped again.

DUMPCCLASS(FREQUENCY(days)): For *days*, substitute a decimal number from 0 to 999.

Automatic full-volume dumps are included in the test for satisfying the requirement defined by the FREQUENCY parameter. Full-volume dumps requested by command, however, are processed independently of the FREQUENCY specified with the DEFINE command. The frequency for the class is considered met if DFSMSHsm has no record of dumping the volume in question to this class. Therefore, the frequency is met for each volume dumped to the class for the first time. DFSMSHsm keeps track of only the last five distinct dump classes to which a volume has been dumped.

Defaults: If you do not specify FREQUENCY, the default for *days* is 7.

Note:

1. Coordinate the FREQUENCY parameter with DUMPCYCLE parameter days to ensure that schedules for automatic dumps do not conflict. **Example:** The following example shows an apparent conflict:

```
DEFINE DUMPCYCLE(NNNNNNY CYCLESTARTDATE(84/01/01)) +
      DUMPCCLASS(A FREQ(8))
```

In this example, DUMPCYCLE specifies a 7-day cycle, with the dumps taken on a Saturday. The dump frequency, FREQ, specifies an 8-day elapsed time. The dumps to class A would be performed every other Saturday, beginning with the second Saturday, because the dump frequency would not have elapsed by the time the dump cycle had completed. If, however, the volume had never been dumped by DFSMSHsm, the initial dump would be on the first Saturday, and then every other Saturday.

To be more consistent with the dump cycle, choose the following value: `FREQ(7)`. This causes the required number of days to elapse before the next 'Y' day in the processing of automatic dump.

2. The `DUMPCLASS` parameter `FREQUENCY` is calculated from the last date that the level 0 volume was dumped while not in debug mode. The date that the level 0 volume was last dumped is not updated when DFSMSHsm is running in debug mode because the dump prevents volume dumps from processing on schedule when no debug is specified. In debug mode, if the date last dumped is zeros, and if the frequency value is not calculated, then the level 0 volume is eligible to be dumped, and the date last dumped is not updated.

DUMPCLASS(HWCOMPRESS): Specifying whether to compress the data to be encrypted

Explanation: `HWCOMPRESS` is an optional parameter specifying whether DFSMSdss is to perform hardware-assisted compression on the dump data. Typically you can use it in conjunction with host based encryption specified in the `DUMPCLASS`, but not in conjunction with tape hardware encryption.

Subparameter	Explanation
HWCOMPRESS(YES)	DUMPCLASS(HWCOMPRESS(YES)) When YES is specified, hardware-assisted compression will be performed. Specify this option only when encryption is being used; it specifies that the data should be compressed before it is encrypted.
HWCOMPRESS(NO)	DUMPCLASS(HWCOMPRESS(NO)) When NO is specified, hardware-assisted compression will not be performed. <code>HWCOMPRESS(NO)</code> can be specified to override an existing setting of <code>HWCOMPRESS(YES)</code> . If the data is not being encrypted, specify <code>HWCOMPRESS(NO)</code> so that tape hardware compression is used.

Defaults: When `HWCOMPRESS` is not specified and was not previously set in the existing dump class definition, NO is used by default.

DUMPCLASS(RECOVERRESET): Specifying whether to reset the data-set-changed indicator in the VTOC entry

Explanation: `RECOVERRESET` is an optional parameter specifying whether or not the data-set-changed indicator in the VTOC entry should be reset for all data sets that are restored during full volume recover processing.

Subparameter	Explanation
RECOVERRESET(YES)	DUMPCLASS(RECOVERRESET(YES)) When YES is specified, the data-set-changed indicator in the VTOC (DS1DSCHA) is reset (set to OFF) for all data sets on the volume being recovered.
RECOVERRESET(NO)	DUMPCLASS(RECOVERRESET(NO)) When NO is specified, the data-set-changed indicator in the VTOC is not reset for any of the data sets on the volume being recovered. This action indicates that the data-set-changed setting retains the value the data sets had the dump was created.

Subparameter	Explanation
RECOVERRESET(DUMP)	DUMPCLASS(RECOVERRESET(DUMP)) When DUMP is specified, and the RESET keyword was specified in the dump class at the time the dump was created, the data-set-changed indicator in the VTOC is reset for all data on the volume being recovered. If RESET was not specified, or NORESET was specified, or the dump copy was created prior to z/OS V2R1 DFSMSHsm, then the data-set-changed indicator is not reset and retains the value it had when the dump copy was created. Note that this parameter is not applicable for FRRECOV (fast replication recovery) because RESET cannot be specified in dump classes associated with the FRBACKUP function. For dump classes used during FRRECOV processing, specify either YES or NO.

Defaults: When RECOVERRESET is not specified and was not previously set in the existing dump class definition, YES is used by default.

DUMPCLASS(RESET and NORESET): Specifying whether DFSMSdss resets the change bit for each data set following a full-volume dump

Explanation: **RESET** and **NORESET** are mutually exclusive, optional subparameters of **DUMPCLASS** specifying whether DFSMSdss resets the change bit for each data set after a full-volume dump is successfully performed.

Subparameter	Explanation
RESET	DUMPCLASS(RESET) specifies that the change bit for each data set should be reset.
NORESET	DUMPCLASS(NORESET) specifies that the change bit for each data set should not be reset.

The specified option affects whether DFSMSHsm makes an incremental backup version of the data sets the next time the volume is processed by incremental backup with the BACKVOL command and whether the data sets have not been changed since the full-volume dump.

If multiple dump copies are made concurrently, each copy is associated with a different dump class. If RESET is specified for any of these dump classes, then the RESET option is specified on the command to DFSMSdss.

Defaults: If neither RESET nor NORESET is specified, the default is NORESET when a dump class is defined for the first time.

Note:

1. If a dump class is already defined (certain characteristics are changed) and neither RESET nor NORESET is specified, the previous specification remains unchanged.
2. If you use the APPLYINCREMENTAL subparameter with the FROMDUMP parameter of the RECOVER command, specify the NORESET subparameter.
3. Do not specify RESET for a dump class that is used for creating fast replication dump copies.

DUMPCLASS(RETENTIONPERIOD): Specifying the enforcement of retention periods for the dump copies in this class

Explanation: **RETENTIONPERIOD** is an optional subparameter of **DUMPCLASS** specifying whether you want DFSMSHsm to enforce retention periods for the dump copies in this class.

Subparameter	Explanation
<i>days</i>	DUMPClass(RETENTIONPERIOD(<i>days</i>)) specifies that DFSMSHsm enforces retention periods for the dump copies in this dump class and then automatically invalidates them when the retention period has been met. After the number of <i>days</i> has elapsed, the dump volumes are automatically reused if the AUTOREUSE subparameter is specified for the class. If AUTOREUSE is not specified, <i>days</i> may be used to specify an expected date to delete the volumes. If the maximum number of dump generations is reached, another dump is performed. If the oldest generation contains a dump copy with an explicit retention period that has not been reached, DFSMSHsm invalidates the oldest dump generation and issues a message reporting this action. The maximum number of dump generations is 100 generations. For <i>days</i> , enter a decimal number between 1 and 9999.
NOLIMIT	DUMPClass(RETENTIONPERIOD(NOLIMIT)) specifies that there is no predetermined expiration date for the dump copies in this DUMPClass. DFSMSHsm does not automatically enforce an expiration date. The dump copy can be invalidated only by an explicit request using the DELVOL command. If the maximum number of dump generations is reached, another dump will be performed. The oldest generation contains a dump copy with RETENTIONPERIOD(NOLIMIT). The maximum number of dump generations is 100 generations. If DFSMSHsm must invalidate a dump copy without a retention period, a message is issued reporting this action.

Defaults: If you do not specify RETENTIONPERIOD, the default is 30 days.

Note:

1. If you use NOLIMIT, also use the NOAUTOREUSE and the NODATASETRESTORE subparameters. If you use AUTOREUSE or DATASETRESTORE, the NOLIMIT parameter overrides the values. The values become NOAUTOREUSE and NODATASETRESTORE respectively.
2. DFSMSHsm does not automatically delete the last and only copy of a source volume, regardless of the RETENTIONPERIOD set. To delete the dump volumes in this copy, issue the DELVOL command with the LASTCOPY option for one of the dump volumes. All of the volumes in the copy are deleted.

DUMPClass(STACK/MAXSTACK) and DUMPClass(MINSTACK): Specifying the maximum and minimum number of dump copies that are stacked for this dump class

Explanation

STACK/MAXSTACK and MINSTACK are optional parameters of the DEFINE DUMPClass command, specifying how many dump copies DFSMSHsm should place on a dump volume (assigned to the specified dump class) during one invocation of automatic dump or BACKVOL dump.

Dump volumes from multiple copy pools with the same dump class will be stacked on the same tape volume as allowed by the values specified for MAXSTACK and MINSTACK.

Subparameter	Explanation
<i>maxv</i>	<p data-bbox="513 191 1476 405">DUMPClass(STACK(<i>maxv</i>)) or DUMPClass(MAXSTACK(<i>maxv</i>)) specifies the preferred maximum number of dump copies that the system should place on a tape volume. In some cases, the available MAXDUMPTASKS tasks may be under utilized if <i>maxv</i> is too large. This could occur when the number of dump copies are not large enough to fill a tape volume to the maximum number of dump copies defined for every available dump task. For <i>maxv</i>, specify a value from 1 to 255.</p> <p data-bbox="513 428 1476 516">When defining a new DUMPClass, if the STACK or MAXSTACK parameter is not specified and the MINSTACK parameter is specified, the MINSTACK value will be used for MAXSTACK. If neither is specified, the default value of 1 will be used.</p> <p data-bbox="513 539 1476 659">When modifying an existing DUMPClass and the MINSTACK parameter is not specified and the STACK or MAXSTACK parameter is specified, then the specified STACK or MAXSTACK parameter must be greater than or equal to the previously defined MINSTACK value for the existing DUMPClass.</p>
<i>minv</i>	<p data-bbox="513 695 1476 1035">DUMPClass(MINSTACK(<i>minv</i>)) specifies the preferred minimum number of dump copies that the system should place on a tape volume. If not specified, the value used by the system will be <i>maxv</i>. The minimum value will be used by the system to determine the number of available MAXDUMPTASKS tasks to use for the dump copies. The effect is that the system could utilize more available dump tasks for the dump copies as compared to using <i>maxv</i>, but could also have less dump copies on the tape volume. <i>minv</i> could be used in conjunction with the common queue dump group to maximize the use of available dump tasks in all the group members in the HSMplex. The system will place dump copies on a tape volume until the <i>maxv</i> is reached. If a value is specified, it must not be greater than <i>maxv</i>.</p> <p data-bbox="513 1058 1476 1146">When defining a new DUMPClass, if the MINSTACK parameter is not specified and the STACK or MAXSTACK parameter is specified, the MAXSTACK value will be used for MINSTACK. If neither is specified, the default value of 1 will be used.</p> <p data-bbox="513 1169 1476 1289">When modifying an existing DUMPClass and the STACK or MAXSTACK parameter is not specified and the MINSTACK parameter is specified, then the specified MINSTACK parameter must be less than or equal to the previously defined STACK or MAXSTACK value for the existing DUMPClass.</p>

All stacking to a given tape is done while that tape is mounted. DFSMSHsm selects empty dump volumes as targets for auto or command dump. Each dump copy is written in its own file, with a unique data set name. As dump generations are rolled off when new dumps are created, a dump volume may contain a combination of valid and not valid dump copies.

A given dump volume:

- Regardless of the STACK/MAXSTACK and MINSTACK value that is used to write to it, is associated with only one dump class.
- Never contains more than one dump copy of the same source volume.
- Can contain dump copies of both SMS-managed volumes and non-SMS-managed volumes.
- Contains dump copies of only source volumes with affinity to the autodump host, or source volumes without such affinity, not a combination of both. System affinity can only be specified for SMS-managed volumes.

Defaults

If you do not specify STACK/MAXSTACK and MINSTACK for a new dump class, DFSMSHsm uses the value of 1. If you do not specify STACK/MAXSTACK and MINSTACK for a dump class that is already defined, the previous value remains unchanged.

Usage notes:

1. If you specify a stack value greater than 99, verify the BCDS maximum record size is 6544 bytes. If it is not, redefine the BCDS with a maximum record size of 6544 bytes. For more information about redefining the BCDS and maximum record size requirements, see [z/OS DFSMSHsm Implementation and Customization Guide](#).
2. DFSMSHsm ignores any STACK/MAXSTACK and MINSTACK value from DEFINE during a BACKVOL DUMP command for a list of volumes, unless you specify the dump class with the DUMPCLASS parameter on the BACKVOL command.
3. VTOCCOPIES are created or not created in the same manner for all volumes dumped to the same dump class.
4. Any RETENTIONPERIOD or EXPIRATIONDATE for the dump class applies to all the dump copies that are written to its dump volumes (all dump copies stacked on a dump volume expire on the same day).

DUMPCLASS(TAPEEXPIRATIONDATE): Specifying a unique expiration date for all dump copies created in a dump

Explanation: **TAPEEXPIRATIONDATE** is an optional subparameter of DUMPCLASS specifying a unique expiration date for the dump tape header labels of all dump copies created in this dump class.

DUMPCLASS(TAPEEXPIRATIONDATE(yyyyddd)): For *yyyyddd*, specify a 7-digit number in the format of *yyyyddd*. The *yyyy* value and the *ddd* value are the year and day of the year for the expiration date of the dump copies. For *yyyy*, specify a year between the current year and 2155. For *ddd*, specify a day between 001 and 366.

Defaults: None.

Note:

1. For TAPEEXPIRATIONDATE to be effective, the security options in effect, as established with a SETSYS command, must include EXPIRATION or EXPIRATIONINCLUDE.
2. Values for *yyyyddd* of 1999365 and 1999366 are recognized as “never expire” dates. Values of 1998000 and 1999000 are recognized as meaningful to certain tape management systems.
3. If you do not specify TAPEEXPIRATIONDATE for a dump class and a tape security option of EXPIRATION or EXPIRATIONINCLUDE is in effect, DFSMSHsm provides an expiration date of 1999 for 365 days for dump volumes written for this class.
4. Past dates are not recognized as expiration dates.

DUMPCLASS(UNIT): Specifying a default tape unit name

Explanation: **UNIT** is an optional subparameter of DUMPCLASS specifying the default unit name for allocation when no volumes are added to the class and when the class has been selected as the target of a dump copy.

DUMPCLASS(UNIT(*unittype*)): For *unittype*, substitute a valid esoteric name or a valid unit type.

The following are valid types of tape units:

- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. If you specify:

- 3480 or 3480X, you use all the functions of the 3480 Magnetic Tape Subsystem.
- 3490, you use all the functions of the 3490 Magnetic Tape Subsystem.
- 3590-1, you use all the functions of the 3590 Magnetic Tape Subsystem.

If you define both 3592-E05 and 3592-J1A to the same esoteric group, the E05 must use the EFMT1 recording format.

You can define a 3592-E05 or 3592-J1A to the 3590-1 generic group. If so, you must ensure drive compatibility or segmentation when using the 3590-1 in a non-SMS environment.

Defaults: If you do not specify UNIT when you define a dump class for the first time, the unit type specified with the SETSYS UNITNAME parameter is the default.

If a dump class is already defined and certain characteristics are being changed, and if UNIT is not specified, the previous specification remains unchanged.



Attention: If UNIT is **not** specified on either the DEFINE or the SETSYS command, the default unit name that is used for allocation is a 3590-1 unit type.

DUMPCCLASS(UNASSIGNTAPEUNIT): Specifying unassigning empty dump volumes

Explanation: **UNASSIGNTAPE** is an optional subparameter of DUMPCCLASS specifying that DFSMSHsm unassigns any dump volumes that are empty and currently added to this class. The unassigned dump volumes are available for selection to any other class that uses the unit type.

Defaults: None.

Note:

- UNASSIGNTAPE can be specified with TAPE or CLOUD dump classes.
- UNASSIGNTAPE is ignored if DISABLE is also specified.
- Use the LIST command to find any dump volumes that are currently assigned to a particular class.

DUMPCCLASS(VTOCCOPIES): Specifying the number of dump VTOC copy data sets to keep for dump copies created in this class

Explanation: **VTOCCOPIES** is an optional subparameter of DUMPCCLASS specifying the number of dump VTOC data sets to keep for dump copies created in this class.

DUMPCCLASS(VTOCCOPIES(*copies*)): For *copies*, substitute a number from 0 to 100. The value indicates how many of the existing dump copies of a given volume dumped to this class should have dump VTOC copy data sets kept associated with them. For example, use 3 as a value for *copies*. If there are six generations of dumps for a given volume and generations 0, 2, and 4 have copies in a class with VTOCCOPIES(3), then all of these generations have VTOC copy data sets kept for them. This happens regardless of what other dump copies might exist in the same dump generations, or whether generations 1, 3, and 5 have dump VTOC copies. If a new dump copy is created in the same class for the same volume, the oldest dump copy in this dump class no longer requires the dump VTOC copy data set. The dump VTOC copy data set is deleted if no other dump copy in the same generation requires it to be kept.

VTOCCOPIES is also used to determine if a dump VTOC copy data set should be created when a volume is dumped. If *copies* is greater than 0, a dump VTOC copy data set is created. If *copies* is 0 and the volume is not dumped to multiple dump classes concurrently, a dump VTOC copy data set is not created. If *copies* is 0 and the volume is dumped to multiple dump classes concurrently, a dump VTOC copy data set is created if any of the dump classes being dumped to has VTOCCOPIES(*copies*) greater than 0.

The dump VTOC copy data sets are written to migration level 1 volumes. Space required for these data sets varies with the size of the original VTOC. Only part of the data set VTOC entry for each data set from

the original VTOC is contained in the VTOC copy data set. When a dump generation no longer has a dump VTOC copy data set associated with it, or when the number of dump VTOC copies kept is zero, DFSMSHsm cannot provide the following:

- Physical data set restore support from the associated dump copies
- Listing capabilities for the dump copy using the DUMPCONTENTS parameter with the LIST DUMPVOLUME command
- Physical data set restore from dump copies in cloud object storage is not supported, therefore, specify VTOCCOPIES(0) for CLOUD dump classes is recommended.
- VTOCCOPIES are not needed for data set recovery from copy pools, so specifying VTOCCOPIES(0) for dump classes used for copy pool dump is recommended.

Dump VTOC copy data sets are not created for or maintained for dumps of DFSMSHsm-owned volumes. When the number of dump VTOC copies kept is zero, DFSMSHsm cannot provide listing capability for the dump copy using the DUMPCONTENTS parameter with the LIST DUMPVOLUME command.

Defaults: If you do not specify VTOCCOPIES when a dump class is defined for the first time, a default of 2 is used for non-copy pool volumes, and a default of 0 is used for copy pool volumes and volumes to be dumped to cloud object storage.

If a dump class is already defined and certain characteristics are changed, and if VTOCCOPIES is not specified, the previous specification remains unchanged.

DUMPCCLASS(ZCOMPRESS): Specifying whether to use zEDC Services on the dump data

Explanation: **ZCOMPRESS** is an optional subparameter specifying whether DFSMSdss is to use zEDC Services on the dump data.

Subparameter	Explanation
ZCOMPRESS(YES)	<p>DUMPCCLASS(ZCOMPRESS(YES)) When YES is specified, compression with zEDC will be performed, if this function is supported by the system. If zEDC Services are <i>not</i> available, the other compression type will be used.</p> <p>For a CLOUD dump class, CDA compression with zEDC Services will be used when a DIRECT cloud provider is specified in the CLOUD parameter.</p>
ZCOMPRESS(NO)	<p>DUMPCCLASS(ZCOMPRESS(NO)) When NO is specified, compression with zEDC will <i>not</i> be performed. ZCOMPRESS(NO) can be specified to override an existing setting of ZCOMPRESS(YES).</p> <p>For a CLOUD dump class, CDA compression will not be used when a DIRECT cloud provider is specified in the CLOUD parameter.</p>

Defaults: When ZCOMPRESS is not specified and was not previously set in the existing dump class definition, NO is used by default.

DUMPCYCLE: Specifying the automatic dump cycle

Explanation: **DUMPCYCLE** is an optional parameter specifying the days in a cycle when the automatic dump process should run.

DUMPCYCLE(*cycle*): For *cycle*, substitute a string of alphabetic Ys and Ns. Each Y represents a day in the cycle when automatic dump is to run, and each N represents a day in the cycle when automatic dump is not to run. You can represent up to 31 days with the Ys and Ns. When you specify *cycle*, the current day is the first day of the cycle unless you specify the cycle start date.

CYCLESTARTDATE can be used with this parameter and is covered separately under [“CYCLESTARTDATE: Specifying the start date of a cycle” on page 814.](#)

Defaults: None.

Note:

1. *Cycle* must be defined sometime before the automatic dump process is permitted to run.
2. *Cycle* is ignored if a volume dump is requested with the BACKVOL command.
3. *Cycle* has no meaning unless the AUTODUMPSTART and BACKUP parameters of the SETSYS command are specified for the DFSMSHsm startup.
4. Dump cleanup also runs on an 'N' day in the dump cycle. Cleanup functions include expiration of expired dump copies and deletion of excess dump VTOC copy data sets and deletion of empty dump containers.

Note: The deletion of excess dump VTOC copy data sets occur on both 'Y' and 'N' days of the dump cycle. Empty dump container deletion only occurs on an 'N' day in the dump cycle.

MIGRATIONCLEANUPCYCLE and SECONDARYSPMGMTCYCLE: Specifying the cycle for automatic secondary space management functions

Explanation: **MIGRATIONCLEANUPCYCLE** and **SECONDARYSPMGMTCYCLE** are mutually exclusive, optional parameters that apply to all secondary space management functions, including migration cleanup and ML1 to ML2 migration.

Both of these parameters allow you to specify the days in a cycle when migration cleanup and ML1 to ML2 migration run.

These two parameters have identical function. The parameter name **MIGRATIONCLEANUPCYCLE** has been retained in order to maintain compatibility, but the parameter name **SECONDARYSPMGMTCYCLE** has been selected as the preferred name, because it more accurately describes the function.

MIGRATIONCLEANUPCYCLE(*cycle*) or **SECONDARYSPMGMTCYCLE(*cycle*)**: For *cycle*, substitute a string of alphabetic Ys and Ns. Each Y represents a day in the cycle when migration cleanup and level 1 to level 2 migration run. Each N represents a day in the cycle when migration cleanup and level 1 to level 2 migration do not run. You can represent up to 31 days with the Ys and Ns. When you specify **MIGRATIONCLEANUPCYCLE** or **SECONDARYSPMGMTCYCLE**, the current day is the first day of the cycle, unless you specify a cycle start date.

Note: In either of the above environments, if **MAXSSMTASKS(EMPTYCONTAINERDELETION(Y))** is specified or defaulted to, the empty container deletion will run every day as every day is considered as the first day of the secondary space management cycle.

CYCLESTARTDATE can be used with either parameter and is covered separately under [“CYCLESTARTDATE: Specifying the start date of a cycle” on page 814.](#)

Defaults: If you do not specify **MIGRATIONCLEANUPCYCLE** or **SECONDARYSPMGMTCYCLE**, *cycle* defaults to a one-day cycle. Therefore, if a start time has been specified by a SETSYS **SECONDARYSPMGMTSTART** command, automatic secondary space management functions will run every day.

If you do not specify both *cycle* and **CYCLESTARTDATE**, DFSMSHsm sets *cycle* to a one-day cycle and **CYCLESTARTDATE** to the current date. Therefore, if a start time has been specified by a SETSYS **SECONDARYSPMGMTSTART** command, automatic secondary space management functions run every day.

Note: It is important to understand the difference between cycle days and calendar days. The first cycle day, DAY 1, begins at the start time of the function. The second cycle day, DAY 2, begins when this start time has again been reached, regardless of when the new calendar day begins. Third and subsequent days follow the same pattern. For instance, if the cycle start time for a function is 1100 (11 a.m.) on Monday, DAY 1 runs from 11 a.m. on Monday to 10:59 a.m. on Tuesday. DAY 2 begins after 11 a.m. on Tuesday and continues until 10:59 a.m. on Wednesday.

MIGRATIONLEVEL2: Specifying DASD level 2 volumes and their associated key range structure

Explanation: **MIGRATIONLEVEL2** is an optional parameter specifying the division of key ranges for DASD migration level 2 volumes. When you specify **MIGRATIONLEVEL2**, you can also specify **KEYS** to indicate how data sets are assigned to the available DASD migration level 2 volumes.

Subparameter	Explanation
KEYS	<p>MIGRATIONLEVEL2(KEYS(<i>key</i> ...)) specifies an ascending alphanumeric sequence of lower key-range boundaries, where the set of initial characters of the data set name is compared against the key. Any data set having the set of initial characters within the specified alphanumeric range migrates to the volume associated with that range. You specify only the lower boundary of the key range for a volume. The maximum number of key ranges you can define is 61.</p> <p>You specify one key less than the number of key ranges being defined because the lower boundary for the first key range defaults to the lowest key possible. For example, KEYS(L T) identifies three key ranges: the first key range begins with A and ends with K9999999, the second key range begins with L and ends with S9999999, and the third key range begins with T and ends with Z9999999.</p> <p>For <i>key</i> ..., substitute one or more strings of 1 to 8 characters, starting with an alphabetic or \$, #, or @. The remaining characters can be alphanumeric and \$, #, or @.</p>
VOLUMES	<p>MIGRATIONLEVEL2(VOLUMES(<i>volser</i> ...)) VOLUMES is an optional subparameter specifying a list of DASD migration level 2 volumes, assigned to the specified key ranges. If you specify fewer volumes than key ranges, DFSMSHsm assigns an available DASD migration level 2 volume to the key range when it needs another volume. An available DASD migration level 2 volume is one that is not associated with a key range and has not been associated with a key range since the last ADDVOL or DELVOL command was issued. If you specify more volumes than key ranges, the extra volumes are not associated with any key range. Before you can specify any DASD migration level 2 volumes with the DEFINE command, you must add the volumes to DFSMSHsm control by using the ADDVOL command.</p> <p>For <i>volser</i> ..., substitute the serial numbers of the DASD migration level 2 volumes, assigned to the corresponding specified key ranges. Only one volume is associated with each key range at a time.</p> <p>When the specified volume for a key range is full, DFSMSHsm assigns an available DASD migration level 2 volume to that key range. To cover all specified key ranges, you must specify one more volume than you specify keys.</p>

Defaults: If you do not specify **KEYS**, DFSMSHsm assumes that you are defining one key range for all possible keys.

Note:

1. You must issue a **DEFINE** command with the **MIGRATIONLEVEL2** parameter before DFSMSHsm can process level 1 to level 2 migration.
2. If you want to reuse the DASD migration level 2 volume after it has been removed from a key range, you must use the **DELVOL** command with the **UNASSIGN** parameter.
3. DFSMSHsm does not process a change to the key range definition while migration to DASD migration level 2 volumes is occurring.

POOL: Identifying the non-SMS-managed volumes where the data sets with the same first qualifier are recalled

Explanation: **POOL** is an optional parameter specifying the set of initial characters of the data set name for a group of data sets and the set of non-SMS-managed volumes where the group of data sets is to be recalled. Each *poolid* and associated volume list is a user-defined pool. When recall occurs, the set of initial characters of the data set name forces the recall of those data sets to one of the specific volumes identified with the **VOLUMES** subparameter. To remove the association between a pool ID and its associated volumes from pool control, you omit the **VOLUMES** subparameter when you specify **POOL**.

POOL(poolid): For *poolid*, substitute the set of initial characters of the data set name of those data sets you want recalled to the specified volume or volumes. The *poolid* can end with a period if **POOL** is the first keyword on the command.

Subparameter	Explanation
VOLUMES	<p>POOL(poolid VOLUMES(volser ...)) specifies the volumes where the data sets with the set of initial characters specified with the POOL parameter are recalled. When you specify multiple volumes for a pool ID, DFSMSHsm tries to recall data sets to the volume with the most space. If there is not enough space on the volume, DFSMSHsm can try recalling the data set to up to four different volumes from the same pool.</p> <p>For <i>volser ...</i>, substitute the serial numbers of the volumes where the group of data sets is to be recalled.</p>

Defaults: None.

Note:

1. When DFSMSHsm checks whether a data set belongs to a pool, it checks the pools in the order you defined them and chooses a pool whose ID matches the initial characters of the data set name. If you define pools, so one is a subset of the other such as MYDSN and MYDSNAME, define the more restrictive pool first (in this case, MYDSNAME) as shown in the following example:

```
DEFINE POOL(MYDSNAME VOLUMES(VOL001))
DEFINE POOL(MYDSN VOLUMES(VOL002 VOL004))
```

- If you do not define the pools in the order shown in the example, as soon as DFSMSHsm sees the MYDSN part of MYDSNAME, it finds a match, and chooses VOL002 even if you wanted to use VOL001.
2. The “JES3 considerations” on page 102 contains detailed information about defining pools in a JES3 environment.
3. You can specify up to 140 volumes.
4. If an SMS-managed volume is specified as part of a data set pool, a message is issued indicating that the volume was not added to the data set pool. Processing continues for the **DEFINE** command. If, however, only SMS-managed volumes are specified for the definition of the data set pool, the data set pool definition is rejected and the **DEFINE** command fails.
5. Volumes specified in the **DEFINE POOL** command must also be added as primary volumes. To do this, use of the **ADDVOL** command.

PRIMARYSPMGMTCYCLE: Specifying the cycle for automatic primary space management

Explanation: **PRIMARYSPMGMTCYCLE** is an optional parameter specifying the days in a cycle when automatic primary space management runs.

PRIMARYSPMGMTCYCLE(cycle): For *cycle*, substitute a string of alphabetic Ys and Ns. Each Y represents a day in the cycle when automatic primary space management is to run. Each N represents a day in the

cycle when automatic primary space management is not to run. You can represent up to 31 days with the Ys and Ns. The current day is the first day of the cycle, unless you specify the cycle start date.

CYCLESTARTDATE can be used with this parameter and is covered separately under [“CYCLESTARTDATE: Specifying the start date of a cycle”](#) on page 814.

Defaults: If you do not specify PRIMARYSPMGMTCYCLE or if you specify PRIMARYSPMGMTCYCLE without specifying *cycle*, automatic primary space management will default to a one-day cycle. Therefore, if a start time has been specified by a SETSYS PRIMARYSPMGMTSTART or AUTOMIGRATIONSTART command, automatic primary space management functions run every day.

If you do not specify both *cycle* and CYCLESTARTDATE, DFSMSHsm will set the cycle to a one-day cycle. Therefore, if a start time has been specified by a SETSYS PRIMARYSPMGMTSTART or AUTOMIGRATIONSTART command, automatic primary space management functions will run every day.

Note:

1. If you specify CYCLESTARTDATE, you must also specify *cycle*; otherwise, the DEFINE PRIMARYSPMGMTCYCLE command will fail.
2. The first day of the cycle is DAY 1. The second day in the cycle is DAY 1 until the start time for the function is reached; then it becomes DAY 2. Third and subsequent days follow the same pattern.

SWITCHTAPES: Releasing mounted tapes for command data set backup

Explanation: SWITCHTAPES is an optional parameter that defines when to release those tapes mounted to process data set backup commands.

Subparameter	Explanation
DSBACKUP	SWITCHTAPES(DSBACKUP) is a required subparameter that specifies that the data set backup function performs a demount of all mounted data set backup tapes.

You can use the following optional parameters with the SWITCHTAPES parameter.

Subparameter	Explanation
TIME	<p>SWITCHTAPES(DSBACKUP(TIME(<i>hhmm</i>))) specifies the time of day to demount mounted data set backup tapes. For <i>hhmm</i>, specify the planned demount time. If you specify TIME with <i>hhmm</i> as zero, DFSMSHsm does not demount any mounted tapes. If the tape is mounted but not actively processing a data set, the demount occurs at the time indicated by TIME. If a tape is actively processing a data set, DFSMSHsm demounts the tape at the end of the current data set.</p> <p>Note: The <i>hhmm</i> format expresses time as a four-digit decimal number that is based on the 24-hour clock. The <i>hh</i> specifies the hour and <i>mm</i> specifies the minutes. For example, you can specify 1:15 p.m. as 1315, and you can specify midnight as 2400. The maximum value you can specify for <i>hh</i> is 24, the maximum value you can specify for <i>mm</i> is 59, and the maximum value you can specify for <i>hhmm</i> is 2400.</p>
AUTOBACKUPEND	SWITCHTAPES(DSBACKUP(AUTOBACKUPEND)) specifies that DFSMSHsm demounts all tapes that were mounted for data set backup at the end of the autobackup cycle. Before the autobackup cycle ends, all retries (data sets that were in use when the initial attempt to backup occurred) complete.
PARTIALTAPE	SWITCHTAPES(DSBACKUP(PARTIALTAPE)) specifies the method that DFSMSHsm uses to mark a single-file format output tape as full or available for reuse when a switch tape event occurs.

Subparameter	Explanation
SETSYS	SWICHTAPES(DSBACKUP(PARTIALTAPE(SETSYS))) specifies that the SETSYS PARTIALTAPE value is used.
REUSE	SWICHTAPES(DSBACKUP(PARTIALTAPE(REUSE))) specifies that all partial tapes demounted with the SWICHTAPES option remain selectable in the DFSMSHsm inventory as partial tapes.
MARKFULL	SWICHTAPES(DSBACKUP(PARTIALTAPE(MARKFULL))) specifies that DFSMSHsm marks full all partial tapes demounted with the SWICHTAPES option. If backups are going to the IBM Virtual Tape Server (VTS) subsystem, refer to the SETSYS PARTIALTAPE parameter description for performance considerations.

Defaults: TIME(0) is the default if you specify the SWICHTAPES(DSBACKUP) command without specifying the TIME or AUTOBACKUPEND parameters. MARKFULL is the default if you specify the SWICHTAPES(DSBACKUP) command without specifying the PARTIALTAPE parameter.

Note: If you specify the MARKFULL parameter of the SWICHTAPES(DSBACKUP) command, DFSMSHsm marks full the tapes that are released from command data set backup. Any other partial tapes in the DFSMSHsm backup inventory are unaffected, and those partials remain as selection candidates and can be immediately selected and mounted for new output processing.

VOLUMEPOOL: Identifying groups of volumes where data sets are recalled

Explanation: **VOLUMEPOOL** is an optional parameter specifying a set of volumes where a data set is recalled based on the volume from which the data set migrated. A data set that migrates from a volume in the volume pool is recalled to a volume in the volume pool.

VOLUMEPOOL(poolid): For *poolid*, substitute from 1 to 8 alphanumeric characters for the name of the volume pool.

Subparameter	Explanation
VOLUMES	<p>VOLUMEPOOL(poolid VOLUMES(volser)) specifies the volumes belonging to the volume pool where the data sets are recalled. When you specify multiple volumes for a pool ID, DFSMSHsm tries to recall data sets to the volume with the most space. If there is not enough space on the volume, DFSMSHsm tries recalling the data set to up to four different volumes in the volume pool.</p> <p>For <i>volser</i>, substitute the serial number of the primary volume or volumes making up the volume pool where data sets are recalled.</p>

Defaults: None.

Note:

1. In a JES3 environment, at least one of the volumes in the volume pool must be added by the ADDVOL command. This JES3 requirement prevents the user from defining a volume pool that cannot be changed and fails the recall every time it is selected. It fails the recall because a volume cannot be selected for recall unless it has been added by the ADDVOL command. It cannot be changed, except at DFSMSHsm startup, because the DEFINE and ADDVOL commands can be used only at DFSMSHsm startup.
2. In a JES2 environment, no volume in the volume pool needs to be added with the ADDVOL command or mounted for the DEFINE command. The RECALL command, however, still requires a volume added by the ADDVOL command before the volume can be selected for the recall. After DFSMSHsm startup, the ADDVOL command can be issued to add a volume, or the DEFINE command can be reissued with a new combination of volumes to change the association between volumes within a volume pool.

3. No checking is done to see if a volume is SMS-managed. A data set migrated from a volume that has been converted to SMS is recalled (if it is to be recalled as non-SMS) to a non-SMS-managed volume in the volume pool. The pool is not used for data sets recalled as SMS.
4. If *poolid* is specified without VOLUMES being specified, the volume pool becomes empty.
5. The hierarchy of selecting a candidate volume for recall is as follows:
 - A limit of 140 volumes can be defined to a volume pool.
 - If a data set being recalled is associated with a data set pool, the candidate volumes are selected from the data set pool.
 - If the volume from which the data set migrated is part of a volume pool, the candidate volume is selected from the volume pool.
 - If a volume is defined in multiple volume pools, the first definition is used. For example, if a data set was migrated from volume VP0001 and is recalled, it is recalled to volume pool SAMPLE1:

```
DEFINE VOLUMEPOOL(SAMPLE1 VOLUMES(VP0001 VP0003))
DEFINE VOLUMEPOOL(SAMPLE2 VOLUMES(VP0001 VP0002 VP0003))
```

- If the volume from which the data set migrated is not part of a volume pool, the candidate volumes are selected based upon the SETSYS RECALL values of ANYSTORAGEVOLUME or PRIVATEVOLUME.

DUMPCCLASS(TAPE | CLOUD): Specifying where the dump copies to reside

Explanation: **TAPE | CLOUD(*cloud_network_connection_name*)** is an optional parameter of DUMPCCLASS that specifies the device category where the dump copies are stored.

Subparameter	Explanation
TAPE	DUMPCCLASS(TAPE) specifies that the associated primary volumes are dumped to tape volumes.
CLOUD(<i>cloud_network_connection_name</i>)	<p>DUMPCCLASS(CLOUD(<i>cloud_network_connection_name</i>)) specifies that the associated volumes are dumped directly to the specified cloud object storage. The <i>cloud_network_connection_name</i> is the name of the SMS network connection construct or the CDA provider file (when OA66197 enabled). If the specified name does not match what is in the SMS construct or the CDA provider file (when OA66197 enabled), the DEFINE DUMPCCLASS fails.</p> <p>To remove the DUMPCCLASS(CLOUD) designation for the dump class, ensure that all associated dump copies are deleted, then specify DUMPCCLASS(TAPE).</p>

Defaults: If you do not specify this parameter, the default is TAPE for a new dump class, or the previous dump device category for an existing dump class.

Note: The following DEFINE DUMPCCLASS parameters are not applicable when the dump copy is resides in the cloud object storage:

- AUTOREUSE | NOAUTOREUSE - These settings apply to dump tape volumes.
- DATASETRESTORE | NODATASETRESTORE - Neither parameter settings are applicable to dump copies residing in cloud storage. Data set recovery from full volume dump copy residing in cloud object storage is not supported as of z/OS V2R5.
- ENCRYPTION - These settings are not applicable in transparent cloud tiering.
- FRRECOV(AFM(NO | YES)) - These settings apply to dump tape volumes.
- HWCOMPRESS(NO | YES) - These settings are not applicable in transparent cloud tiering.
- STACK | MAXSTACK, MINSTACK - These settings apply to dump tape volumes.
- TAPEEXPIRATIONDATE - These settings apply to dump tape volumes.

UNIT(*unitname*) - These settings apply to dump tape volumes.

Examples of how to code the DEFINE command

The following examples present different ways to code the DEFINE command:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Specifying key ranges for DASD migration level 2 volumes

In this example, key ranges are assigned to the specified migration level 2 volumes. Data sets with the set of initial characters of the data set name within key range A through F9999999 migrate to the level 2 volume with volume serial number VOL002. Data sets with the set of initial characters of the data set name within key range G through L9999999 migrate to the level 2 volume with volume serial number VOL005. Data sets with the set of initial characters of the data set name within key range M through R9999999 migrate to the level 2 volume with volume serial number VOL007. Data sets with the set of initial characters of the data set name within key range S through Z9999999 migrate to the level 2 volume with volume serial number VOL009.

```
DEFINE MIGRATIONLEVEL2(KEYS(G M S) VOLUMES(VOL002 VOL005 +
VOL007 VOL009))
```

Specifying a pool of volumes

In this example, two volumes are specified as the pool used when data sets with the pool ID FE.T3322 are recalled.

```
DEFINE POOL(FE.T3322 VOLUMES(FET001 FET002))
```

Specifying the automatic primary space management cycle

In this example, an automatic primary space management cycle of five days is defined. The automatic primary space management functions are to run on the fifth day of the cycle. CYCLESTARTDATE without a date specified is redefined to today's date.

```
DEFINE PRIMARYSPMGMTCYCLE(NNNNY CYCLESTARTDATE)
```

Specifying the automatic secondary space management cycle

In this example, an automatic secondary space management cycle of seven days is defined. The automatic secondary space management functions are to run on the seventh day of the cycle. CYCLESTARTDATE without a date specified is redefined to today's date.

```
DEFINE SECONDARYSPMGMTCYCLE(NNNNNY CYCLESTARTDATE)
```

Specifying a backup cycle

In this example, a backup cycle of seven days is defined, with two volumes used for each day in the cycle when volume backup is processed. The example specifies that the automatic backup function is processed on the second and fifth days of the cycle. The starting date for the backup cycle is October 1, 1997.

```
DEFINE BACKUP(NYNNYNN 2 CYCLESTARTDATE(1997/10/01))
```


Specifying a dump cycle

In this example, a dump cycle of seven days is defined, with a cycle start date of 96/12/29. If the cycle start date is a Sunday, the specified dump cycle results in the automatic dump function running Monday through Friday, but not running on Saturday or Sunday.

```
DEFINE DUMPCYCLE(NYYYYYN +
  CYCLESTARTDATE(1996/12/29))
```

Specifying the number of VTOC copy data sets to keep for each volume dumped

In this example, 12 copies of the VTOC copy data sets are to be kept for a dumped volume.

```
DEFINE DUMPCCLASS(class VTOCCOPIES(12))
```

Specifying a dump class for a volume

In this example, a dump class for a 3590-1 type volume with a class name of *weekly* is defined. The other parameters used in our example are:

- DFSMSdss resets the change bit for each data set.
- A physical data set restore is allowed.
- The volume can be automatically reused when invalidated by the retention period of 15 days.
- The target day of the dump for this dump class is the sixth day of the dump cycle.
- The dump volume is to be dumped every Friday night.
- A maximum of ten dump copies can be placed on this volume.

```
DEFINE DUMPCCLASS(WEEKLY UNIT(3590-1) +
  RESET DATASETRESTORE AUTOREUSE +
  RETENTIONPERIOD(15) DAY(6) +
  DISPOSITION('DUMP EVERY FRI NITE.') +
  (STACK(10)))
```

Specifying a pool of volumes to be used for recovering an aggregate group

In this example, four volumes are specified to be used for recovery of the aggregate group *PAY1*. Two volumes are for recovery of level 0 non-SMS-managed data sets, and two volumes are for recovery of migrated data sets.

```
DEFINE ARPOOL(PAY1 ML1VOLS(MPAY1 MPAY2) +
  L0VOLS(PAY001 PAY002))
```

Specifying when to demount and deallocate output tapes that are used by command data set backup

In this example, output tapes that are used by command data set backup are demounted at the end of the autobackup cycle, and the devices are deallocated. Single-file format output tapes are not available for reuse when a switch tape event occurs.

```
DEFINE SWICTAPES(DSBACKUP(AUTOBACKUPEND PARTIALTAPE(MARKFULL)))
```

Specifying encryption and HWCOMPRESS settings

In this example, the data will be compressed with hardware-assisted compression and then encrypted prior to being dumped.

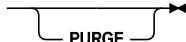
DEFINE command for DFSMSHsm

```
DEFINE DUMPCCLASS(ENCRYPT(RSA(CCA.CRT08.INT.ENC.1024S0F)  
TYPE(CLRTHDES)) HWCMPRESS(YES))
```


Chapter 41. DELETE Command: Deleting a migrated data set from a migration volume

The DELETE command deletes a migrated data set without recalling the data set. When you specify the DELETE command, DFSMSHsm deletes the MCDS data set record and the migrated data set. DFSMSHsm does not delete backup versions of the data set. If you want to delete the backup versions of a deleted data set, you must use the BDELETE command. See [Chapter 37, “BDELETE command: Deleting backup versions of a data set,” on page 797](#) for information about the BDELETE command.

Syntax of the DELETE command

► DELETE — *dsname* — 

Required parameters of the DELETE command

This topic describes the required parameter of the DELETE command.

dsname: Specifying the data set to be deleted

Explanation

dsname is a required parameter specifying the fully qualified name of the migrated data set that you want to delete. DFSMSHsm deletes the data set without recalling it.

Defaults

None.

Usages notes:

1. You cannot specify an alias for *dsname*. If you do, DFSMSHsm does not delete the data set.
2. DFSMSHsm fails any DFSMSHsm delete command that specifies a member name of a partitioned data set.
3. DFSMSHsm will not uncatalog an original data set unless the cataloged volume serial is MIGRAT. This prevents DFSMSHsm from inadvertently scratching a newly cataloged data set of the same name.
4. The command fails for both SMS-managed and non-SMS-managed data sets if the *dsname* specified on the command is determined by DFSMSHsm to be one of the following:
 - An associated part of a VSAM data set (for example, a data component name, index component name, path name) rather than the base cluster name.
 - The name of a migration copy data set.

Optional parameters of the DELETE command

This topic describes the optional parameter of the DELETE command.

PURGE: Specifying deletion of a data set

Explanation

PURGE is an optional parameter specifying deletion of the data set *dsname* even if the data set is still within its retention period.

Defaults

If PURGE is not specified, DFSMSHsm deletes data set *dsname* only if it has exceeded its retention period.

Examples of how to code the DELETE command

The following examples present different ways to code the DELETE command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Deleting a migrated data set

In this example, a migrated data set (SMS or non-SMS) is deleted from a migration volume without recalling the data set.

```
DELETE WIN3357.PLAN.PLI
```

Deleting a migrated data set within its retention period

In this example, a migrated data set (SMS or non-SMS) is deleted from a migration volume even if it is within its retention period.

```
DELETE L541563.PSF.N.F230EP03.DSET1 PURGE
```

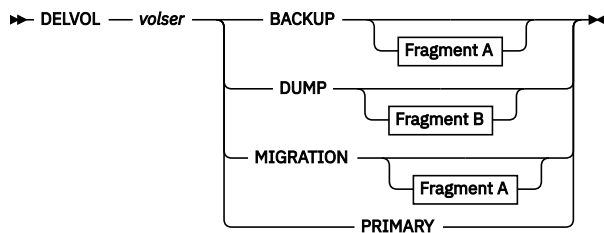

Chapter 42. DELVOL command: Removing a volume from DFSMSHsm management

The DELVOL command reassigns the volume as not managed or owned by DFSMSHsm. You can remove a primary volume (in this instance, PRIMARY includes both volumes added using the ADDVOL command and SMS-managed volumes), migration volume, dump volume, dump version of a copy pool, or backup volume from the list by specifying the serial number and the category of the volume you want to remove. If a DELVOL command is entered for an SMS-managed volume, the MCV record for the volume, if it exists, is deleted. You can remove only one volume each time you issue the DELVOL command unless the volume specified is part of a valid dump copy, then all volumes containing part of the same dump copy are removed from DFSMSHsm.

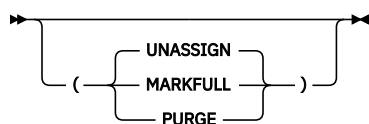
Although you can respecify a deleted non-SMS-managed volume with the ADDVOL command, historical information about the deleted volume is lost. This information is included in the MCDS and BCDS volume records. If you use ADDVOL to respecify the volume, DFSMSHsm recreates the volume record without the historical information.

Note: The DELVOL command must be issued on each system in the DFSMSHsm plex where this volume is to be used.

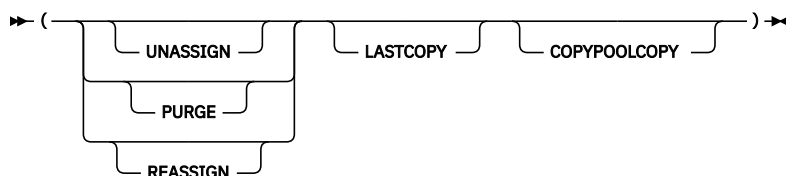
Syntax of the DELVOL command



A: BACKUP and MIGRATION optional parameters



B: DUMP optional parameters



Note:

1. If you specify the COPYPOOLCOPY, LASTCOPY, MARKFULL, PURGE, REASSIGN, or UNASSIGN subparameter of the BACKUP, DUMP, or MIGRATION parameter, you must put parentheses around the subparameter.
2. If the volume serial number is for an SMS-managed volume, only the PRIMARY parameter applies.
3. If a DELVOL MIGRATION command is used to delete a migration volume while the automatic secondary space management function is running, the DELVOL MIGRATION command fails.
4. If SMS storage groups change, use the DELVOL command to remove from DFSMSHsm control those volumes that were removed from the SMS storage groups before daily automatic functions started.

Remove those volumes on all DFSMSHsm systems that know about them so that the MVT entries are no longer present.

Required parameters of the DELVOL command

This topic describes the required parameters of the DELVOL command.

volser: Specifying the volume you are deleting

Explanation: *volser* is a required positional parameter for which you substitute the serial number of the volume you want to delete.

Defaults: None.

Note:

1. Because *volser* is a required positional parameter, you must specify it immediately after the command name.
2. Intermittent DELVOLs of active SMS-managed volumes have little effect, because the list of volumes is obtained prior to each automatic function, and MVT and MCV records for the volumes are built at that time.

BACKUP, DUMP, MIGRATION, and PRIMARY: Specifying the type of volume

Explanation: **BACKUP, DUMP, MIGRATION, and PRIMARY** are mutually exclusive, required parameters specifying what type of volume to delete.

BACKUP specifies that a backup volume is reassigned as an unassigned volume or, if the PURGE subparameter is specified, deleted from DFSMSHsm control.

DUMP specifies that a dump volume is disassociated from the dump class to which it is currently assigned and all record of its contents is discarded. In addition, the PURGE or REASSIGN subparameters can be specified to remove the volume from the control of DFSMSHsm or to retain the associated dump class respectively. If the volume specified is part of a valid dump copy, all volumes containing part of the same dump copy are treated similarly.

MIGRATION specifies that a migration volume is changed as follows:

- In the case of a migration level 1 volume, the volume is deleted from DFSMSHsm control.
- In the case of a DASD migration level 2 volume, the volume is no longer associated with its key range and is available for selection. If the optional PURGE subparameter is specified, the volume is removed from the control of DFSMSHsm.
- In the case of a tape migration level 2 volume, the volume is marked as not selected and is therefore available unless full. If the optional PURGE subparameter is specified, the volume is removed from the control of DFSMSHsm if no valid data exists on the volume.

Note: In the preceding cases, DFSMSHsm determines from the serial number whether the volume you have identified with the *volser* positional parameter is a migration level 1 volume or a migration level 2 volume.

PRIMARY specifies that a primary volume (a volume added using the ADDVOL command or an SMS-managed volume) is deleted from DFSMSHsm control. DFSMSHsm removes the record of the deleted volume from the MCDS.

Defaults: None.

Note:

1. If you issue the BACKUP parameter of the DELVOL command while the backup volume is in use, the command fails.
2. If you issue the MIGRATION parameter of the DELVOL command while automatic volume space management or automatic secondary space management is running, the command fails.

3. If you issue the PRIMARY parameter of the DELVOL command while that volume is being processed by automatic space management, automatic backup, or automatic dump, the command fails.
4. If you issue the DUMP parameter of the DELVOL command while automatic dump is running, the command fails.
5. If you use the MIGRATION parameter of the DELVOL command to delete a migration level 2 tape volume that has incomplete data set information saved in the OCDS TTOC record, the command fails.

Optional parameters of the DELVOL command

This topic describes the optional parameters of the DELVOL command.

COPYPOOLCOPY: Deleting a copy pool dump volume

Explanation: **COPYPOOLCOPY** is an optional subparameter that must be specified to delete a dump volume that contains one or more copy pool dump copies. During DELVOL processing, a REQUIRED COMPLETE or ALL COMPLETE dump version of a copy pool is marked as a partial so that dump processing for that version can be resumed at a later time if the DASD copy is still valid.

Defaults: None.

LASTCOPY: Invalidating the only valid dump copy of a source volume

Explanation: **LASTCOPY** is an optional subparameter that must be specified to delete a dump volume that is part of the only valid dump copy of a source volume. LASTCOPY is ignored if it is not part of the only copy.

Defaults: None.

MARKFULL, PURGE, REASSIGN, and UNASSIGN: Specifying the options for deleting a backup, dump, or ML2 volume

Explanation: **MARKFULL**, **PURGE**, **REASSIGN**, and **UNASSIGN** are mutually exclusive, optional subparameters of the BACKUP, DUMP, and MIGRATION parameters specifying whether the volume is deleted, reassigned, or marked full.

Note:

1. MARKFULL applies only to tape backup and migration volumes.
2. These subparameters do not apply to primary or migration level 1 volumes.

MARKFULL with BACKUP or MIGRATION

When you specify MARKFULL with the BACKUP or MIGRATION parameters, MARKFULL specifies that the requested backup or migration volumes be marked full. No other processing will be performed. This subparameter allows you to make a tape volume available for RECYCLE or TAPECOPY processing by marking the volume full. It can also be used to prevent a volume from being selected for further output processing.

PURGE with BACKUP

When you specify PURGE with the BACKUP parameter, PURGE specifies that DFSMSHsm delete all volume information, including the tape table of contents (TTOC) for a tape backup volume. DFSMSHsm removes the record of the deleted volume from the BCDS. In addition, DFSMSHsm deletes any association with a day in the backup cycle or with a set of spill backup volumes.

If the volume is a tape volume, all valid backup versions that exist on the tape volume are deleted, and the data on the tape is no longer usable. If an error occurs in deleting a valid backup version, DFSMSHsm continues deleting all other backup versions but does not delete the backup volume in which the error has occurred.

PURGE with DUMP

When you specify PURGE with the DUMP parameter, PURGE specifies that DFSMSHsm delete all volume information for a dump volume. DFSMSHsm removes the record of the deleted volume from the BCDS. In addition, DFSMSHsm deletes any association with the day in the dump cycle or with the dump class. DFSMSHsm also deletes any valid dump copies. The dump volume then becomes a scratch tape.

Depending upon the tape security option used for protection by DFSMSHsm, the volume may have to be reinitialized before being reused by another user or job.

PURGE with MIGRATION

When you specify PURGE with the MIGRATION parameter, PURGE specifies that DFSMSHsm delete all volume information for a non-SMS migration level 2 volume, including the TTOC for a tape non-SMS migration level 2 volume. Before DFSMSHsm removes a tape migration level 2 volume from its control, DFSMSHsm checks whether the volume still contains valid data. If it does, DFSMSHsm fails the DELVOL command and issues message ARC0260I.

If the PURGE parameter has been used because the ML2 tape was lost or destroyed and if the command fails and message ARC0260I is displayed, perform the following steps:

1. Issue the MIGRATE VOLUME (*volser* DBA(0)) command to remove all record of valid data. Only non-SMS data sets whose expiration dates have been reached are scratched. If the ML2 tape volume contains SMS data, the volume is not eligible for DELVOL processing.
2. Reissue the DELVOL PURGE command.

Note: Do this only if the tape is totally unusable because the DELVOL PURGE command removes what may be the only copy of the data. If backup copies are available, recover the data sets that the MIGRATE command deleted. If the volume is DASD or if the volume is a tape that does not contain valid data, DFSMSHsm removes the record of the deleted volume from the MCDS.

REASSIGN with DUMP

When you specify the REASSIGN subparameter with the dump parameters, REASSIGN specifies that the contents of that volume are to be discarded. The volume is also made available and is associated with its original dump class. The original dump class is the class specified with the ADDVOL command. If no class is specified with ADDVOL, then there is no associated dump class after the DELVOL REASSIGN. DFSMSHsm can reuse the reassigned volume without reinitialization. REASSIGN is applicable only for dump volumes.

UNASSIGN with BACKUP

When you specify the UNASSIGN subparameter with the BACKUP parameter, UNASSIGN specifies that the volume is no longer associated with the day in the backup cycle or with the set of spill volumes. The volume becomes an unassigned backup volume. All volume and data set information is kept in the BCDS and, for tape backup volumes, in the OCDS. The unassigned volume can be selected as a daily or spill backup volume.

UNASSIGN with DUMP

When you specify UNASSIGN with the DUMP parameter, UNASSIGN specifies that the volume is to remain known to DFSMSHsm but must be removed from its current use. The volume is also disassociated from the dump class to which it is currently assigned. All records of the volume's contents must be discarded. DFSMSHsm can reuse the volume without reinitialization. If the volume is available for reuse, it is available to any dump class requiring a volume of the same unit type.

UNASSIGN with MIGRATION

When you specify UNASSIGN with the MIGRATION parameter, UNASSIGN specifies that the DASD migration level 2 volume is no longer associated with its key range and is available for selection. If the

volume is a tape migration level 2 volume, DFSMSHsm marks the volume as not selected. Therefore, the tape migration level 2 volume is available unless it is full.

Defaults: If you do not specify PURGE, REASSIGN, or UNASSIGN, the default is UNASSIGN.

Note:

1. You can delete valid data from a tape backup volume or dump volume if you specify the PURGE parameter. DFSMSHsm deletes all control data set information associated with each valid backup version dump copy on the purged tape backup volume or dump volume. As a result, DFSMSHsm can no longer recover backup versions from the tape volume.
2. You specify only a migration level 2 volume when you use the PURGE or UNASSIGN subparameter with the MIGRATION parameter. If you specify a migration level 1 volume, DFSMSHsm ignores the parameter and a purge is performed.
3. If you use the PURGE parameter to remove a tape volume from control of DFSMSHsm and the tape volume is protected by a password or expiration date, reinitialize the tape volume. You reinitialize the tape volume with the IEHINITT utility. If you do not reinitialize the purged tape volume and another program tries to write on the tape volume, message IEC512I is issued and the tape volume is unloaded.
4. If you do not reinitialize the purged tape volume, and if DFSMSHsm protected the tape volume with an expiration date, and another program tried to write on the tape volume, the following occurs:
 - Message IEC507D is issued.
 - The operator is prompted to approve or disapprove the attempt to write over an expiration-date-protected data set whose expiration date has not been reached. If the operator disapproves the use of the tape volume, the tape volume is unloaded and removed from the inventory of backup or migration volumes.
5. Messages IEC512I and IEC507D occur only if the program trying to write on the tape volume is not using the volume verification exit to override the password or expiration date protection on the first file on the tape volume.
6. If you use the ADDVOL command to add a purged tape volume to the list of tape volumes owned by DFSMSHsm and the purged tape volume was previously protected by a password or expiration date, reinitialize the tape volume. If you do not reinitialize the purged tape volume and another program tries to write on the tape volume, message IEC512I is issued, the tape volume is unloaded, and the tape volume is removed from the inventory of backup, dump, and migration volumes.
7. If a dump volume has more than one dump copy stacked on it, you must use the PURGE parameter to delete the volume.

Examples of how to code the DELVOL command

The following examples present different ways to code the DELVOL command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Deleting a primary volume

In this example, a primary volume (added using the ADDVOL command or SMS-managed) is deleted from DFSMSHsm control.

```
DELVOL VOL005 PRIMARY
```

Deleting a migration level 1 volume

In this example, a migration level 1 volume is deleted from DFSMSHsm control.

```
DELVOL MIG003 MIGRATION
```


Marking a migration level 2 tape volume full

In this example, a migration level 2 tape volume is marked full. The MARKFULL parameter does not delete the volume.

```
DELVOL MIG003 MIGRATION(MARKFULL)
```

Unassigning a backup volume

In this example, a backup volume becomes an unassigned backup volume of DFSMSHsm.

```
DELVOL BAC001 BACKUP(UNASSIGN)
```

Deleting a tape migration level 2 volume

In this example, a tape migration level 2 volume is purged from DFSMSHsm if it does not contain any valid data.

```
DELVOL TML203 MIGRATION(PURGE)
```

Deleting a DASD migration level 2 volume

In this example, a DASD migration level 2 volume is purged from DFSMSHsm.

```
DELVOL DML201 MIGRATION(PURGE)
```

Deleting a tape backup volume

In this example, a tape backup volume is deleted from DFSMSHsm's control. All record of any valid backup versions on the tape is also deleted.

```
DELVOL TAPE01 BACKUP(PURGE)
```

Deleting a dump volume

In this example, a dump volume is deleted and its contents are discarded. The volume is made available and is associated with its original dump class.

```
DELVOL DUPT01 DUMP(REASSIGN)
```

Deleting a dump volume that is part of the only valid dump copy of a source volume

In this example, a dump volume is part of the only valid dump copy of a source volume. The volume is made available but is not associated with any dump class.

```
DELVOL DUPT02 DUMP(UNASSIGN LASTCOPY)
```

Purging the records for a dump volume that is part of the only valid dump copy of a source volume

In this example, a dump volume is part of the only valid dump copy of a source volume. The volume is purged from DFSMSHsm.

```
DELVOL DUPT02 DUMP(PURGE LASTCOPY)
```


Deleting a copy pool dump volume

In this example, an unneeded dump volume that contains one or more copy pool dump copies is deleted from DFSMSHsm control.

```
DELVOL DUPT03 DUMP(COPYPOOLCOPY)
```

Chapter 43. DISPLAY command: Displaying DFSMSHsm storage locations

The DISPLAY command is used for maintenance purposes only; for information, see [*z/OS DFSMSHsm Diagnosis*](#).

Chapter 44. EXPIREBV command: Deleting unwanted backup versions of data sets and UNIX files

The EXPIREBV command is used to delete unwanted backup and expired ABARS versions of SMS-managed and non-SMS-managed data sets and unwanted backup versions of UNIX files from DFSMShsm-owned storage. The optional parameters of the EXPIREBV command determine the deletion of the backup versions of non-SMS-managed data sets and UNIX files. The management class attributes determine the deletion of backup versions of SMS-managed data sets. The management class fields **Retain Extra Versions** and **Retain Only Version** determine which ABARS versions or incremental backup versions are deleted. The RETAIN DAYS value that is provided during the data set or UNIX file backup request determines how long a data set backup copy is kept for both SMS-managed and non-SMS-managed data sets and UNIX files.

A detailed description of eligibility considerations for both SMS-managed and non-SMS-managed data sets is provided in the [Chapter 8, “Aggregate backup and recovery support \(ABARS\),” on page 319](#).

EXPIREBV determines if a backed-up data set should be considered SMS-managed (and thus whether to apply management-class attributes or EXPIREBV command parameters in expiring cataloged backup versions) by the most recent status of the data set:

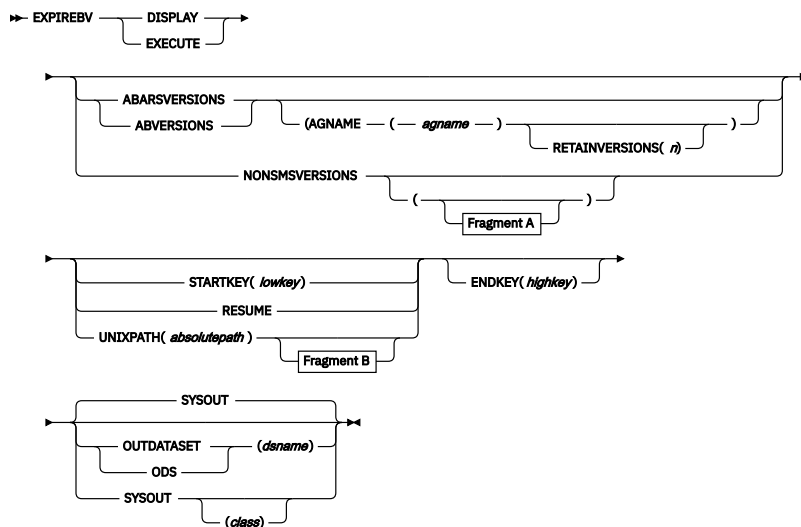
- If the data set is still cataloged, SMS-managed status is taken from the data set’s current catalog entry.
- If the data set has been deleted (that is, if it is no longer cataloged), SMS-managed status is the same as it was the last time the data set was backed up.

The SMS Management class of the zFS data set that contains the UNIX file system will apply to the UNIX file. Specifically, the number of backup versions attribute in the management class will apply to backup versions of the UNIX file. If the zFS that contains the UNIX file is non-SMS managed, then the value of SETSYS VERSIONS will apply.

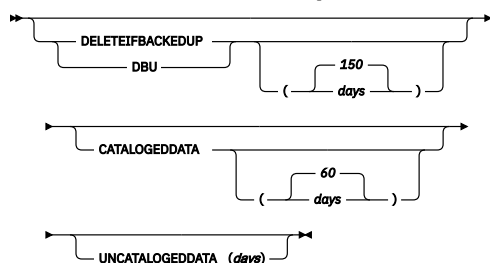
When the EXPIREBV command is processing a backup of a UNIX file, the management class of the zFS data set containing the UNIX file is used to determine the expiration criteria for the backup. If the zFS data set that contained the UNIX file does not exist, the management class that was associated with the file when the last backup version was made is used. If the UNIX file is determined to be non-SMS-managed, then the DELETEIFBACKEDUP or CATALOGEDDATA sub-parameters of the NONSMSVERSIONS parameter criteria is used.

Note: The EXPIREBV command processes a backup of a UNIX file only if the containing file system is mounted. If the file system is unmounted, the backup version processing is skipped.

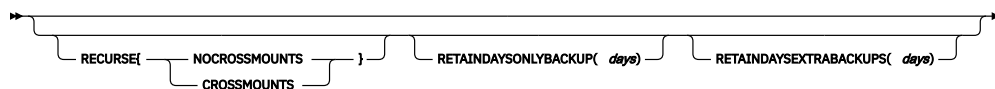
Syntax of the EXPIREBV command



A: NONSMSVERSIONS Optional Parameters



B: UNIXPATH Optional Parameters



Note: The NONSMSVERSIONS parameter does not apply to SMS-managed data sets.

Required parameters of the EXPIREBV command

This topic describes the required parameters of the EXPIREBV command.

DISPLAY and EXECUTE: Specifying whether to delete the backup versions

Explanation

DISPLAY and EXECUTE are mutually exclusive, required parameters specifying whether to delete the ABARS or backup versions or just to produce a listing of the ABARS or backup versions eligible for deletion.

DISPLAY

produces a line in the listing for every ABARS or backup version eligible for deletion based on the appropriate input parameters or management class attributes. The listing is written to a particular output class or DASD data set, depending on the value of the OUTDATASET | SYSOUT parameter.

EXECUTE

specifies that the ABARS or backup versions should be deleted based on the input parameters and management class attributes. The output consists of an ARC0734I message for each backup version eligible for deletion. The messages are written to the backup activity log, depending on the values specified by SETSYS ACTLOGMSGLVL:

- FULL specifies that all ARC0734I messages are produced.
- EXCEPTIONONLY specifies that an ARC0734I message is produced only when there is a failure.
- REDUCED has the same effect as EXCEPTIONONLY.

Defaults

None.

Usage notes:

1. If the DISPLAY parameter is specified, no backup versions are deleted. If, however, any backed up, cataloged data sets have been scratched and do not already have a scratch date recorded, the current date is recorded as the scratch date for those data sets. This scratch date is used by subsequent EXPIREBV commands to determine whether active backup versions of these cataloged data sets are eligible for deletion.
2. When EXPIREBV is invoked and it detects a scratched data set, it creates a scratch date for that data set. The date is stored in the BCDS MCB record in a 4-byte field MCBSCDR (offset X'48', packed decimal). The date can be displayed using the FIXCDS command. When EXPIREBV is again invoked and it encounters the same data set—which now has an established scratch date—it deletes the data set (when the data set meets the criteria for deletion). Thus, if you are running EXPIREBV for the first time, you do not get a list of deleted data sets. On subsequent occurrences, you will get a list, as long as the data sets meet the criteria for deletion.

Optional parameters of the EXPIREBV command

This topic describes the optional parameters of the EXPIREBV command.

ABARSVERSIONS: Specifying that only aggregate group versions are to be processed

Explanation

ABARSVERSIONS is an optional parameter specifying that only ABARS activity records (ABR) in the BCDS are processed by the EXPIREBV command. This parameter is mutually exclusive with the NONSMSVERSIONS parameter and is functionally exclusive to processing backup versions of SMS data sets. This parameter can be used to delete old ABR records.

The following are optional parameters of the ABARSVERSIONS parameter:

AGNAME(*agname*)

Process only the activity records associated with the specified aggregate group (*agname*).

RETAINVERSIONS(*n*)

Retain a specific number (*n*) of ABR versions. This parameter is used with the AGNAME parameter when EXECUTE is specified. If you do not want to retain any versions of a particular aggregate group's ABR record, specify zero. Specifying zero allows you to clean up the ABR records for aggregate groups no longer being processed.

Defaults

None.

Usage notes:

1. Running EXPIREBV for ABARS should only be needed if you have an aggregate that you no longer use and you need to clean up old records. For more information, see [“Expiring ABR records in the BCDS” on page 354](#).

NONSMSVERSIONS: Specifying expiration criteria for backup versions of data sets not managed by SMS

Explanation

NONSMSVERSIONS is an optional parameter specifying that you want backup versions of non-SMS-managed data sets processed during processing of the EXPIREBV command.

The following are optional parameters of the NONSMSVERSIONS parameter:

DELETEIFBACKEDUP

Controls the deletion of retired versions. If the age of a retired version exceeds *days*, the retired version is deleted along with all other backup versions that were made for the same cataloged non-SMS-managed data set.

CATALOGEDDATA

Controls the deletion of backup versions (other than the retired version) of non-SMS-managed data sets that were cataloged when they were backed up. If the difference between the scratch date recorded by a prior EXPIREBV command and the current date exceeds *days*, all backup versions of the cataloged data set, except the retired version, are deleted. If the cataloged data set has not been deleted, the *days* parameter is ignored and all excess backup versions (as a result of an (H)ALTERDS command) are deleted.

UNCATALOGEDDATA

Controls the deletion of backup versions made of non-SMS-managed data sets that were uncataloged when they were backed up. If the age of a backup version exceeds *days*, the backup version is deleted.

days

Number of days (0–9999) to keep a retired version (DELETEIFBACKUP), a backup version of a cataloged data set after the EXPIREBV command determines the data set has been deleted (CATALOGDDATA), or a backup version of an uncataloged data set (UNCATALOGEDDATA).

Defaults

If RETAINDDAYS is specified at the time of data set backup, the RETAINDDAYS value takes precedence over NONSMSVERSION. If RETAINDDAYS is not specified and you do not specify NONSMSVERSIONS on the command, backup versions of non-SMS-managed data sets are not processed.

If you specify DELETEIFBACKEDUP with no *days* value, the DFSMSHsm default is 150 days. If you do not specify DELETEIFBACKEDUP on the command, retired versions are not processed.

If you specify CATALOGEDDATA with no *days* value, the default for *days* is 60. If you do not specify CATALOGEDDATA on the command, backup versions of cataloged, non-SMS-managed data sets are not processed.

If you specify UNCATALOGEDDATA, you must specify *days*. If you do not specify UNCATALOGEDDATA on the command, backup versions of uncataloged, non-SMS-managed data sets are not processed.

Usage notes:

1. The result of the CATALOGEDDATA and UNCATALOGEDDATA subparameters is based on whether a data set was cataloged or uncataloged when a given backup version was created, not its current status.
2. Eligibility for deletion of a backup version under the CATALOGEDDATA criterion is based on the number of days that have passed since a prior EXPIREBV command determined that a cataloged data set has been scratched. This means that you must issue the EXPIREBV command a minimum of two times: once to record that the cataloged data set has been scratched and once to calculate whether the difference between the current date and the recorded scratch date exceeds the specified number of days.
3. At least one of the optional subparameters of NONSMSVERSIONS must be specified.

4. The EXPIREBV command examines only the latest backup version for a data set name to determine the existence of a retired version, and it treats all backup versions older than the latest version (and all versions of data sets uncataloged at the time of backup) as nonretired versions. (A retired version is the backup version DFSMSHsm creates before it scratches a data set during data set retirement. DFSMSHsm does not automatically delete retired versions.)
5. If the SETSYS versions parameter is decreased and the EXPIREBV command is executed prior to the time the next backup version is created, all excess versions—whether cataloged or uncataloged—are deleted.
6. Eligibility for deletion of a backup version under the CATALOGEDDATA criterion for existing cataloged data sets is based on the current version limit.

STARTKEY and RESUME: Determining the starting point for BCDS processing during command execution

Explanation

STARTKEY and RESUME are mutually exclusive, optional parameters that you specify to control from which BCDS record the processing is started during command execution.

STARTKEY

Specifies a start key.

lowkey

Provides a starting point for processing in the BCDS. The value of *lowkey* can be up to 44 alphanumeric characters that identify the data set for which you want the deletion of backup versions to begin. Numbers are higher in the range (they follow Z) than letters. The value specified cannot end in a period. If fewer than 44 characters are entered, the value specified is appended with X'00' to make 44 characters.

RESUME

Starts processing at the point the EXPIREBV command left off the last time an EXPIREBV command was processed on this host.

Defaults

If you do not specify STARTKEY or RESUME, processing begins at the beginning of the BCDS. If you specify RESUME and the most recent EXPIREBV command processed on this DFSMSHsm host did not end early, processing begins at the beginning of the BCDS.

Usage notes:

1. During processing of the EXPIREBV command, the current key (data set name) is saved periodically.

UNIXPATH: Specifying the UNIX file, empty directory, or nonempty directory to analyze for backup

Explanation

UNIXPATH is used to specify an absolute path (that is, starting from the '/' [root] directory) to a UNIX file, empty directory, or nonempty directory to analyze for backup expiration. If a UNIX file or empty directory is specified, then only that is analyzed. If a nonempty UNIX directory is specified, then the directory node itself is checked (in the case it was backed up when it was an empty directory) and then all direct children.

The following are optional parameters of the UNIXPATH parameter:

absolute path

Absolute UNIX path starting from the root (that is, '/') directory. The absolute path might contain one of the following:

1. Absolute UNIX path to a file (for example, '/absolute/path/myfile.json')

2. Absolute UNIX path to an empty directory (for example, '/absolute/path/emptydir/').
3. Absolute UNIX path to a nonempty directory (for example, '/absolute/path/nonemptydir/')

Note: When a directory (whether empty or nonempty) is specified a trailing '/' must be included. For example:

Incorrect '/absolute/path/mydir'

Correct '/absolute/path/mydir/'

Defaults

None.

Usage notes:

1. Specifying UNIXPATH alone does not cause DFSMSHsm to recurse into subdirectories unless the RECURSE option is explicitly specified also.
2. The RESUME option, as it relates to the UNIXPATH('absolute/path') being the most recent EXPIREBV command run but was CANCELED for whatever reason, is targeted as follows:
 - If UNIXPATH only was specified (for example, without RECURSE) and when processing it is CANCELED and later RESUME was specified (caveat is no other EXPIREBV commands are run in between that CANCELED instance and trying to RESUME), DFSMSHsm continues as follows:
 - When a UNIX file is specified in this scenario, RESUME causes DFSMSHsm to continue processing siblings within the same containing directory as the original UNIX file (DFSMSHsm does not recurse into subdirectories).
 - When an empty UNIX directory is specified in this scenario, RESUME causes DFSMSHsm to continue processing siblings within the same containing directory as the original UNIX file (DFSMSHsm does not recurse into subdirectories).
 - When a nonempty UNIX directory is specified in this scenario, RESUME causes DFSMSHsm to continue traversing the sibling linked list of the given directory from that last saved key in the BCR (Recall that EXPIREBV processing state is periodically saved in the BCR record). Note that this continuation in processing looks at siblings that are UNIX files or empty UNIX directories, it does not recurse into subdirectories when they are found.
3. Issuing the RESUME option after a successful completion of an EXPIREBV UNIXPATH (+- any other optional parameters) causes EXPIREBV to begin from the first record in the BCDS, as is the case with other combinations that finish successfully and then are RESUMEd (for example, STARTKEY). It is important to note is that DFSMSHsm does not continue processing from within the UNIX file tree if RESUME is issued after a successful EXPIREBV UNIXPATH run.

RECURSE : Used with UNIXPATH to recurse into subdirectories of the directory specified with UNIXPATH

Explanation

RECURSE is an optional keyword exclusive to UNIXPATH, that when used with UNIXPATH causes DFSMSHsm to recurse into subdirectories of the specified directory with UNIXPATH. This allows logically related subtrees within the file system to be checked together.

The following are optional parameters of the UNIXPATH parameter:

NOCROSSMOUNTS

Indicates that, during recursion into subdirectories of the containing parent directory, DFSMSHsm should not recurse into subdirectories belonging to a different mount than the containing parent.

CROSSMOUNTS

Indicates that, during recursion into subdirectories of the containing parent directory, DFSMSHsm should recurse into subdirectories belonging to a different mount than the containing parent.

Defaults

If nothing is specified with the RECURSE option, then NOCROSSMOUNTS is assumed by default.

Usage notes:

1. If the absolute path specified with UNIXPATH is to a file or empty directory, RECURSE is ignored.
2. The RESUME option, as it relates to the UNIXPATH('absolutepath') + RECURSE(CROSSMOUNTS | NOCROSSMOUNTS) being the most recent EXPIREBV command run but was CANCELED for whatever reason, is targeted as follows:
 - If UNIXPATH('absolutepath') + RECURSE(CROSSMOUNTS | NOCROSSMOUNTS) is specified, and at some point during processing it is CANCELED and later RESUME is specified, it is considered an error and processing stops with message ARC0685I issued.

RETAINDAYONLYBACKUP: Optional keyword used in conjunction with UNIXPATH to override the RETAINDAYONLYBACKUP value assigned from the Management Class

Explanation

User may need the ability to override the RETAINDAYONLYBACKUP value assigned from the Management Class on an individual Unix file(s) or [sub]directory basis.

Defaults

None.

Usage notes: If not specified, the RETAINDAYONLYBACKUP value from the Management Class is used.

RETAINDAYSEXTRABACKUPS: Optional keyword used in conjunction with UNIXPATH to override the RETAINDAYSEXTRABACKUPS value assigned from the Management Class

Explanation

User may need the ability to override the RETAINDAYSEXTRABACKUPS value assigned from the Management Class on an individual Unix file(s) or [sub]directory basis.

Defaults

None.

Usage notes: If not specified, the RETAINDAYSEXTRABACKUPS value from the Management Class is used.

ENDKEY: Determining the ending point for BCDS processing during command execution

Explanation

ENDKEY is an optional parameter that you specify to control on which BCDS record the processing is completed during command execution.

The following is a required parameter of the ENDKEY parameter:

highkey

Provides an ending point for processing in the BCDS. The value of *highkey* can be up to 44 alphanumeric characters that identify the data set for which you want the deletion of backup versions

to end. Numbers are higher in the range (they follow Z) than letters. The value specified cannot end in a period. If fewer than 44 characters are entered, the value specified is appended with X'FF' to make 44 characters.

Defaults

If the ENDKEY subparameter is not specified, the planned end key is different, depending upon whether you specify RESUME. If the RESUME parameter is specified without ENDKEY, processing continues until the last specified planned end key, saved from the last EXPIREBV command issued for this host. If RESUME is specified without ENDKEY and the most recent EXPIREBV command processed in this host did not end early, processing continues to the end of the BCDS. If neither RESUME nor ENDKEY is specified, processing continues to the end of the BCDS.

Usage notes:

1. When processing is started for the EXPIREBV command, the planned ending key is saved.
2. If the value of *highkey* is equal to the value of *lowkey*, DFSMSHsm attempts to process only the data set (MCB) records with those initial characters. If the value of *highkey* is less than the value of *lowkey*, DFSMSHsm fails the command with an error message.

OUTDATASET and SYSOUT: Specifying the output location for EXPIREBV DISPLAY listing

Explanation

OUTDATASET and SYSOUT are mutually exclusive, optional parameters specifying the output location for the listing produced while running the EXPIREBV command with the DISPLAY option.

OUTDATASET

Directs output to a data set.

If the data set does not exist, DFSMSHsm dynamically allocates and catalogs an output data set with the following characteristics:

- Data set name specified (*dsname*)
- Record format of fixed-blocked with ANSI control characters (FBA)
- Logical record length of 121
- Data set is system reblockable
- Primary allocation of 20 tracks (see note)
- Secondary allocation of 50 tracks (see note)
- Unit of SYSALLDA (see note)

If the data set already exists, DFSMSHsm will use the data set. The data set must have the following characteristics:

- The data set must be cataloged and on DASD.
- The data set record format must be FBA and the logical record length must be 121.
- The data set is system reblockable.
- The user can choose the primary space allocation.
- If DFSMSHsm needs additional extents after the primary space allocation, DFSMSHsm uses a secondary space allocation of 50 tracks. (see note)
- If the data set does not contain data, DFSMSHsm starts writing output data at the beginning of the data set.
- If the data set contains data, DFSMSHsm writes the output data after the existing data.

dsname

Name of the data set to which DFSMSHsm writes the output data. For *dsname*, substitute the fully qualified name of the data set that receives the EXPIREBV output.

SYSOUT

Directs output to a specified output class. This is the default.

class

Alphanumeric character representing the output class. The default class is A, unless otherwise specified by the SETSYS command.

Defaults

The default is SYSOUT, and the default for *class* is the value specified with the SETSYS command. If you do not use the SYSOUT parameter of the SETSYS command to specify the output location, the default is class A.

Usage notes:

1. These parameters do not apply when the EXECUTE option is specified. If either of these parameters is entered when it does not apply, it is ignored.
2. If you select the OUTDATASET option, you can use the PATCH command to change the unit name, primary allocation, and secondary allocation. If you select the SYSOUT option, you can use the PATCH command to change whether, and how, DFSMSHsm limits the lines of SYSOUT output. Refer to the *z/OS DFSMSHsm Implementation and Customization Guide* for a list of the available PATCH commands.

Examples of how to code the EXPIREBV command

The following examples present different ways to code the EXPIREBV command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Displaying backup versions within a key range that qualify for deletion

In this example, a starting message is issued to the user. An output line is directed to SYSOUT class H for every SMS backup version that qualifies for deletion. All SMS backup versions with key ranges from A000 to L888 are processed. Deletion qualification is determined by the SMS management class attributes for each data set as they are defined for that data set. No deletions are performed and no non-SMS processing is done. An ending message is issued to the user.

```
EXPIREBV DISPLAY STARTKEY(A000) ENDKEY(L888) SYSOUT(H)
```

Deleting SMS backup versions and specific non-SMS versions that qualify for deletion

In this example, a starting message is directed to the backup activity log and to the user. DFSMSHsm attempts to delete all SMS backup versions that qualify for deletion. An ARC0734I message is directed to the backup activity log under the control of the SETSYS ACTLOGMSGLVL. Deletion is determined by the SMS management class attributes as they are defined for each data set.

DFSMSHsm processes all retired versions of non-SMS data sets. DFSMSHsm attempts to delete all retired versions and associated cataloged backup versions which are over 100 days old. An ARC0734I message is written to the backup activity log under control of the SETSYS ACTLOGMSGLVL command. An ending message is directed to the backup activity log and to the user.

```
EXPIREBV EXECUTE NONSMSVERSIONS(DBU(100))
```


Displaying SMS backup versions and non-SMS cataloged versions that qualify for deletion

In this example, a starting message is directed to the user. DFSMSHsm processes all SMS backup versions. An output line is directed to SYSOUT class A for each backup version that qualifies for deletion. Deletion is determined by the SMS management class attributes as they are defined for each data set.

An output line is directed to SYSOUT class A for all non-SMS backup versions of deleted data sets that have been cataloged and more than five days have elapsed since EXPIREBV determined that the data set was deleted. DFSMSHsm does not perform any deletions. An ending message is directed to the user.

```
EXPIREBV DISPLAY NONSMSVERSIONS(CATALOGEDDATA(5))
```

Displaying, in an output dataset, the SMS backup versions and non-SMS uncataloged versions that qualify for deletion

In this example, a starting message is directed to the user. DFSMSHsm processes all SMS backup versions. An output line is directed to the output dataset for each backup version that qualifies for deletion. Deletion is determined by the SMS management class attributes as they are defined for each data set.

DFSMSHsm processes all uncataloged backup versions, and an output line is directed to the output dataset for each uncataloged backup version that is over 10 days old, thereby qualifying for deletion. DFSMSHsm does not perform any deletions. An ending message is directed to the user.

```
EXPIREBV DISPLAY NONSMSVERSIONS(UNCATALOGEDDATA(10)) ODS(DSNAME)
```

Resuming deletion of SMS backup versions that qualify following release of a previously held EXPIREBV command

In this example, EXPIREBV processing had been held, with the HOLD EXPIREBV command, before a prior EXPIREBV command had finished. Later, EXPIREBV processing was released by issuing the RELEASE EXPIREBV command and then resumed by issuing the EXPIREBV RESUME EXECUTE command.

```
EXPIREBV RESUME EXECUTE
```

A starting message is directed to the backup activity log and the user. Processing begins where the prior EXPIREBV command left off and continues to the ENDKEY specified in the prior command. If no ENDKEY is specified, processing continues to the end of the BCDS.

DFSMSHsm attempts to delete all SMS backup versions that qualify for deletion. An ARC0734I message is directed to the backup activity log under the control of the SETSYS ACTLOGMSGVLVL. Deletion is determined by the SMS management class attributes as they are defined for each data set.

No non-SMS backup versions are processed. An ending message is directed to the backup activity log and the user.

Sample of a printer listing when you specify EXPIREBV ABARSVERSIONS with the DISPLAY parameter

Figure 94 on page 861 is a sample of a printer listing when you specify EXPIREBV ABARSVERSION with the DISPLAY parameter.


```

DISPLAY OF ABARS VERSIONS ELIGIBLE FOR EXPIRATION AT 18:21:50 ON 1992/08/25 FOR 381A
COMMAND INPUT: ABARSVERSIONS AGNAME(PAY1) RETAINVERSIONS(0000)
SUBMITTER'S USERID: **OPER-
AGNAME = PAY1                      VERSION = 0001
AGGREGATE VERSION KEY = PAY1.1992238000101          DELETED*
(* DETERMINED ON 1992/08/25)
AGGREGATE GROUP FILES                      FILE STATUS
PAY1.C.C01V0001                          NOT IN CATALOG
PAY1.D.C01V0001                          NOT IN CATALOG
PAY1.O.C01V0001                          NOT IN CATALOG
AGNAME = PAY1                      VERSION = 0002
AGGREGATE VERSION KEY = PAY1.1992238000201          DELETED*
(* DETERMINED ON 1992/08/25)
AGGREGATE GROUP FILES                      FILE STATUS
PAY1.C.C01V0002                          UNCATALOGED
PAY1.D.C01V0002                          UNCATALOGED
PAY1.O.C01V0002                          UNCATALOGED
END OF DISPLAY - 00000002 ABARS  VERSIONS ELIGIBLE FOR EXPIRATION

```

Figure 94. Sample Printer List of Data Sets When You Specify EXPIREBV ABARSVERSIONS and DISPLAY

Sample of a printer listing when you specify EXPIREBV with the DISPLAY parameter

Table 57 on page 861 presents the header information when you specify EXPIREBV and DISPLAY.

Table 57. Headings of Output When You Specify EXPIREBV and DISPLAY	
Printer Output Heading	Description
SYS CAT	YES indicates that the backup version was made from a cataloged data set.
GEN NMBR	This field contains the generation number of the backup version. The most recent backup version is 0, the next most recent is 1, and so forth.
AGE	This is the number of days between the creation of the backup version and the execution of the EXPIREBV command.
RET VERS	YES indicates that the version being listed is a retired version. NO indicates that the version is not a retired version.
BACK PROF	YES indicates that a RACF discrete backup profile exists. NO indicates that a RACF discrete backup profile does not exist.

Figure 95 on page 861 is a sample of a printer listing when you specify EXPIREBV with the DISPLAY parameter. Each data set listed is deleted if the same EXPIREBV command is issued with the EXECUTE parameter.

```

DISPLAY OF BACKUP VERSIONS ELIGIBLE FOR EXPIRATION AT 05:44:14 on 1991/01/07 for SYSTEM=3081
COMMAND INPUT: STARTKEY(M734413.F240BV09.N02.NAME44.XXX.NAMEIS44.BYT) ENDKEY(M734413.F240BV09.N04.NAME44.XXX.NAMEIS44.BYT)
DSNAME = M734413.F240BV09.N02.NAME44.XXX.NAMEIS44.BYT NOW CATALOGED, AS SMS
MANAGEMENT CLASS USED = DEFAULT1
BACKUP VERSION DSNAME                      SYS  GEN  RET  BACK
CAT  NMBR AGE  VERS PROF
DFHSM.BACK.T074205.M734413.F240BV09.H7001 YES  003  006  NO  NO
DFHSM.BACK.T214205.M734413.F240BV09.H7001 YES  002  006  NO  NO
DFHSM.BACK.T534105.M734413.F240BV09.H7001 YES  004  006  NO  NO
DFHSM.BACK.T464205.M734413.F240BV09.H7003 YES  001  004  NO  NO
DSNAME = M734413.F240BV09.N03.NAME44.XXX.NAMEIS44.BYT NOW CATALOGED, AS SMS
MANAGEMENT CLASS USED = DEFAULT1
BACKUP VERSION DSNAME                      SYS  GEN  RET  BACK
CAT  NMBR AGE  VERS PROF
DFHSM.BACK.T094205.M734413.F240BV09.H7001 YES  003  006  NO  NO
DFHSM.BACK.T234205.M734413.F240BV09.H7001 YES  002  006  NO  NO
DFHSM.BACK.T554105.M734413.F240BV09.H7001 YES  004  006  NO  NO
DFHSM.BACK.T474205.M734413.F240BV09.H7003 YES  001  004  NO  NO
DSNAME = M734413.F240BV09.N04.NAME44.XXX.NAMEIS44.BYT NOW CATALOGED, AS SMS
MANAGEMENT CLASS USED = DEFAULT1
BACKUP VERSION DSNAME                      SYS  GEN  RET  BACK
CAT  NMBR AGE  VERS PROF
DFHSM.BACK.T104205.M734413.F240BV09.H7001 YES  003  006  NO  NO
DFHSM.BACK.T244205.M734413.F240BV09.H7001 YES  002  006  NO  NO
DFHSM.BACK.T564105.M734413.F240BV09.H7001 YES  004  006  NO  NO
DFHSM.BACK.T494205.M734413.F240BV09.H7003 YES  001  004  NO  NO
END OF DISPLAY - 00000012 BACKUP VERSIONS ELIGIBLE FOR EXPIRATION

```

Figure 95. Sample Printer List of Data Sets When You Specify EXPIREBV and DISPLAY

Chapter 45. FIXCDS command: Displaying or modifying MCDS, BCDS, and OCDS records

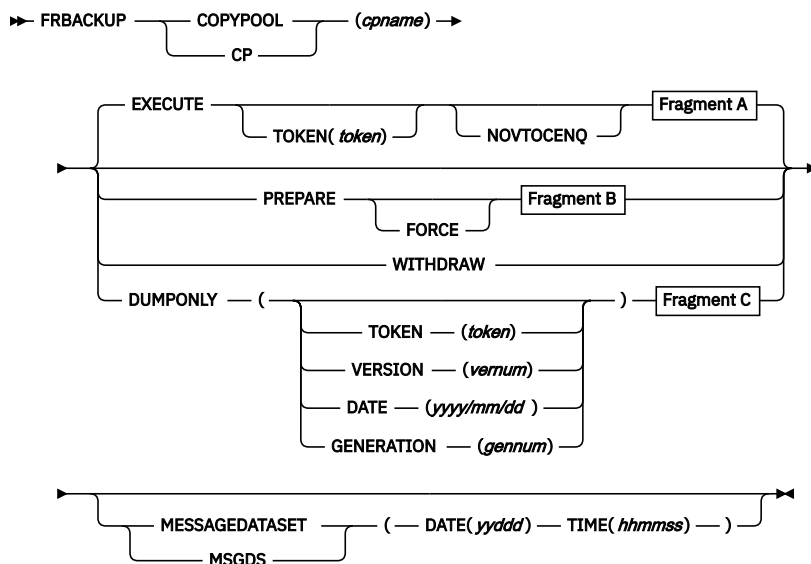
The FIXCDS command is used for maintenance purposes only; for more information, see [*z/OS DFSMSHsm Diagnosis*](#).

Chapter 46. FRBACKUP command: Requesting a fast replication backup or dump version

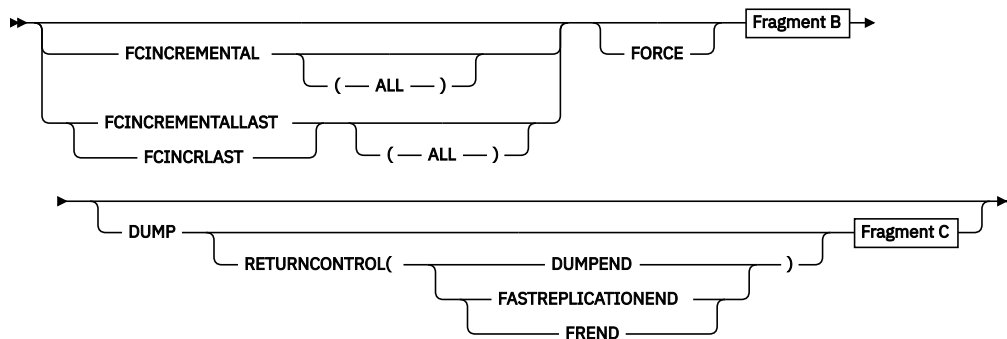
The FRBACKUP command enables you to create a fast replication backup version for sets of storage groups followed by a dump of each fast replication target volume. A set of storage groups is defined through the SMS construct termed 'copy pool'. Storage groups are defined to a copy pool and all of the volumes in those storage groups are backed up or dumped by the fast replication function. For a discussion of the copy pool construct, see [z/OS DFSMSdfp Storage Administration](#).

Note: Because fast replication can be used to make volume-level recoveries, there are important factors to consider to ensure that the backup and recovery process restores your data to a usable state. Before using the fast replication function, see “Managing volume backups with fast replication” on page 254.

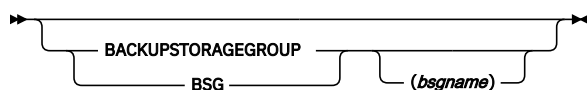
Syntax of the FRBACKUP command



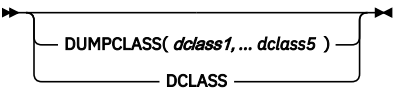
A: EXECUTE Optional Parameters



B: EXECUTE and PREPARE Optional Parameters



C: DUMP and DUMPONLY Optional Parameters



Note: The RETURNCONTROL parameter only applies to the DUMP parameter. It does not apply to the DUMPONLY parameter.

Required parameter of the FRBACKUP command

This topic describes the required parameter of the FRBACKUP command.

COPYPOOL: Specifying the name of the copy pool for backup

Explanation: **COPYPOOL**(*cpname*) is a required parameter that is used to identify the copy pool for backup. For *cpname*, substitute the name of the copy pool for which to create a backup version.

Defaults: None.

Optional parameters of the FRBACKUP command

This topic describes the optional parameters of the FRBACKUP command.

BACKUPSTORAGEGROUP: Overriding the copy pool backup storage group with an alternate storage group

Explanation: **BACKUPSTORAGEGROUP** is an optional keyword that can be used with the EXECUTE and PREPARE keywords. It is a copy pool backup storage group that is defined within the pool storage group.

You can use the **BACKUPSTORAGEGROUP** parameter to override the copy pool backup storage group with an alternate storage group. Only one alternate copy pool backup storage group may be used at a time. If more than one alternate copy pool backup storage group is specified on the command, the other copypool backup storage group is ignored and no error message is issued.

When **BACKUPSTORAGEGROUP** is specified with an alternate copy pool backup storage group (CPBSG) name, DFSMSHsm will attempt to override the existing CPBSG specifications in the copy pool with the alternate. When **BACKUPSTORAGEGROUP** is specified without a storage group name, DFSMSHsm will attempt to use the CPBSG defined in the pool type storage group(s). For more information, see [Chapter 6, “Availability management of SMS-managed storage,”](#) on page 165.

DUMP: Creating a fast replication DASD version followed by a dump

Explanation: **DUMP** is an optional parameter that first causes DFSMSHsm to establish a Fast Replication relationship between the source and target volumes. When all relationships are successfully established, DFSMSHsm starts dumping the target DASD copies to tape. If the copy pool is defined with **VERSIONS=0**, DFSMSHsm withdraws the source and target relationship after creating the dump copies for the source volume. For a copy pool defined with **VERSIONS** not equal to zero, the hardware automatically withdraws the relationship as the background DASD copies finish.

Subparameter	Explanation
RETURNCONTROL	DUMP RETURNCONTROL specifies when DFSMSHsm is to return control to the application or user for WAIT type requests.

The following subparameters are mutually exclusive, required parameters that you must use with the **RETURNCONTROL** parameter.

Subparameter	Explanation
DUMPEND	RETURNCONTROL(DUMPEND) specifies to DFSMSHsm to return control after all volumes are dumped.
FASTREPLICATIONEND	RETURNCONTROL(FASTREPLICATIONEND) specifies to DFSMSHsm to return control just before the dumping processing starts.

Defaults: DUMPEND is the default for the RETURNCONTROL parameter.

DUMPCCLASS: Specifying unique dump classes for fast replication dump processing

Explanation: **DUMPCCLASS** is an optional parameter that allows you to specify up to five unique dump classes per copy pool. Fast Replication dump processing produces one dump copy of each volume for each dump class. You can mark a dump class as required or optional in the dump class definition. If all required dump classes successfully complete, DFSMSHsm marks the dump version as complete.

On the FRBACKUP DUMP command, for *dclass1*, ... *dclass5*, up to five dump class names, each separated by a comma. If a dump class name is listed multiple times, DFSMSHsm fails the command.

Note: If DUMPCCLASS is specified on the FRBACKUP command that initiated a dump for that version, that dump class is marked as required, by default, and all other dump classes associated with the copy pool are ignored.

Defaults: None.

EXECUTE, PREPARE, WITHDRAW, and DUMPONLY: Specifying the backup or dump action

Explanation: **EXECUTE**, **PREPARE**, **WITHDRAW**, and **DUMPONLY** are mutually exclusive, optional parameters that indicate to DFSMSHsm the type of backup or dump action that you are initiating.

EXECUTE: The EXECUTE parameter indicates that DFSMSHsm is to create a fast replication backup for each volume defined in the specified copy pool. EXECUTE is the default action.

PREPARE: When you use this parameter, DFSMSHsm assigns specific target volumes to each source volume for each version of the copy pool.

This parameter does not verify that there is a sufficient amount of tape volumes to successfully dump the version. If you specified VERSIONS=0 for the copy pool, DFSMSHsm does not preassign DASD backup volumes to the copy pool source volumes but PREPARE processing checks that sufficient target volumes are available.

This PREPARE processing enables validation of the fast replication environment outside the backup window. The PREPARE processing does not create a backup of the copy pool.

WITHDRAW: The WITHDRAW parameter indicates that all outstanding FlashCopy relationships are withdrawn for the most recent valid version. DFSMSHsm marks that DASD version invalid if any FlashCopy relationships are withdrawn.

Any successful dump copies for the withdrawn version will remain but any incomplete dumps for the version are not resumable.

You should use the WITHDRAW parameter when it is necessary to withdraw one or more relationships established by FRBACKUP. If a version other than the most recent is invalidated, then that version is also deleted because it is no longer usable. Withdraw of a relationship can only be performed at the copy pool level. If any part of a copy pool backup version is invalidated, then the entire version is invalidated.

Note: If you withdraw the only backup version, recovery cannot be performed. Use the LIST COPYPOOL command or the ARCXTRCT macro to ensure that the backup version you wish to withdraw is not the only backup version.

If you specify the WITHDRAW parameter for a version that is being dumped, DFSMSHsm will not process any volumes that have not yet been dumped. For those volumes that are actively being dumped, after the dump completes, DFSMSHsm marks the dump as a failure. The dump copy for the withdrawn version will be marked as being partially complete. Since the DASD copy has been withdrawn, the dump version cannot be completed.

DUMPONLY: You can use the DUMPONLY parameter when a valid DASD backup version exists and you need to create or complete a dump copy for that version. The DUMPONLY parameter does not initiate a DASD copy.

The following special considerations exist between the DUMPONLY and DUMPCLASS parameters:

- If you specify the DUMPONLY parameter for a version that has an incomplete dump and you have not specified the DUMPCLASS parameter, DFSMSHsm resumes the dump process for any dump classes not successfully dumped.
- If you specify the DUMPONLY parameter and the DUMPCLASS parameter for a dump class that previously failed the dump process, DFSMSHsm resumes the dump process for that dump class.
- If you specify the DUMPONLY parameter and the DUMPCLASS parameter and the dump of a dump class was not previously attempted and less than five dump classes exist for the version, DFSMSHsm creates a new dump copy.

Note: If five successful dump versions already exist, DFSMSHsm fails the dump.

- If you specify the DUMPONLY parameter without the DUMPCLASS parameter and successful dump copies already exist for that version, DFSMSHsm fails the command.

The following subparameters are mutually exclusive, optional parameters that you can use with the DUMPONLY parameter to indicate to DFSMSHsm which specific DASD backup version to dump.

Subparameter	Explanation
DATE	DUMPONLY(DATE (<i>yyyy/mm/dd</i>)) indicates that you want to dump a backup version from a particular date. For <i>yyyy/mm/dd</i> , substitute the creation date for the backup version.
GENERATION	DUMPONLY(GENERATION (<i>gennum</i>)) indicates that you want to dump a particular generation of a backup version. For <i>gennum</i> , substitute the generation number of the backup version.
TOKEN	DUMPONLY(TOKEN (<i>token</i>)) indicates that you want to dump a backup version with a particular token. For <i>token</i> , specify the token for the backup version.
VERSION	DUMPONLY(VERSION (<i>vernum</i>)) indicates that you want to dump a particular version of a backup version. For <i>vernum</i> , specify the specific version number.

Defaults: If you do not specify any of these optional parameters with the FRBACKUP command, the default is EXECUTE. If you do not specify any optional subparameters with the DUMPONLY parameter, the default is GENERATION(0).

FCINCREMENTAL: Requesting DFSMSHsm to create a copy pool version using incremental FlashCopy

Explanation: FCINCREMENTAL is an optional keyword that can be used with EXECUTE. It can be specified with the optional subparameter ALL.

Volumes in an incremental backup version have persistent relationships.

FCINCREMENTAL(ALL) indicates that an incremental copy pool version is to be created, and all subsequent copy pool backup copies should also be incremental.

FCINCREMENTAL without the ALL subparameter indicates the version that is created is an incremental version.

Note:

1. Creating and maintaining multiple incremental backup versions causes each source volume in the copy pool to have multiple incremental targets. Each volume is allowed to have up to 12 incremental targets. That is, when you specify FCINCREMENTAL(ALL), the copy pool VERSIONS must be set to a value between 1 and 12.
2. You can specify the FCINCREMENTAL keyword (without the ALL subparameter) to designate multiple incremental copies. When a version designated as incremental is deleted, the indication of that copy as incremental is lost and the FCINCREMENTAL keyword must be re-specified on a future FRBACKUP command to create a replacement incremental backup copy.
3. If an incremental copy is desired for each copy pool backup, it is recommended that the FCINCREMENTAL(ALL) keyword (instead of multiple FCINCREMENTAL requests) be used. If all backup copies of the copy pool are deleted (both disk and tape copies), the FCINCREMENTAL(ALL) parameter must be reissued when creating the first new backup.
4. The Multiple Incremental FlashCopy target function must be enabled when attempting to create multiple incremental copy pool copies or to recover a backup version created with the Multiple Incremental FlashCopy target support. Otherwise, the FRBACKUP or FRRECOV command will fail. The Multiple Incremental FlashCopy target function can be disabled with MULTINCRFLC=NO in the DEVSUPxx member of parmlib.

FCINCREMENTALLAST: Requesting a FlashCopy incremental copy with a subsequent withdraw

Explanation: FCINCREMENTALLAST is an optional keyword that can be used with EXECUTE. It can be specified with the optional subparameter ALL.

FCINCREMENTALLAST(ALL) causes a new incremental of the oldest copy to be taken and the persistent relationships of that copy to be withdrawn. Relationships of all other backup copies are withdrawn after the next increment is taken of each of those copies. Relationships of any new incremental copies created subsequent to the command being issued are persistent, that is, they are not impacted by the previous FCINCREMENTALLAST(ALL) request.

FCINCREMENTALLAST without the ALL subparameter causes a new increment to be taken of the oldest backup copy, and then the persistent relationships of only that copy to be withdrawn. If the FCINCREMENTAL(ALL) keyword was previously specified for the same copy pool then when this copy rolls off, it is replaced with a new incremental copy.

FORCE: Replacing a DASD backup copy that has an incomplete dump copy

Explanation:

Usually, if a new DASD copy replaces an existing DASD copy that has an incomplete dump copy, the creation of the new DASD copy is failed. FORCE is an optional parameter that allows you to force a DASD copy with an incomplete dump copy to be rolled-off and replaced with a new DASD copy.

The FORCE parameter is required when the creation of a new dump version causes the roll of a DASD backup version before a dump completes.

Defaults: None.

MESSAGEDATASET: Specifying the date and time used in the message data set name

Explanation: MESSAGEDATASET is an optional parameter that allows you to specify the date and time used in the name of the fast replication command message data set. Specifying the date and time ensures that you can identify the message data set that corresponds with the FRBACKUP request. If you don't specify the date and time with the MESSAGEDATASET parameter, DFSMSHsm uses the date and time when the message data set name is generated.

The value for the DATE parameter must be a five-digit numeric value, *yyddd*.

The value for the TIME parameter must be a six-digit numeric value, *hhmmss*.

If a duplicate message data set already exists with the specified date and time, DFSMSShsm issues message ARC1873I and continues without capturing the fast replication messages in a message data set.

Use of the message data set is controlled with the MESSAGEDATASET parameter of the SETSYS FASTREPLICATION command. Refer to [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSShsm control parameters,”](#) on page 1057 for more information.

Defaults: None.

NOVTOCENQ: Requesting DFSMSShsm to not serialize the VTOC of processed volumes

Explanation: **NOVTOCENQ** is an optional parameter that you use to request DFSMSShsm to not serialize on the VTOCs of the volumes being processed. Do not use this parameter unless you have taken specific action to ensure that no update activity will occur to the volumes being processed.

Defaults: None.

TOKEN: Specifying an identifier for each copy pool backup version

Explanation: **TOKEN(token)** is an optional parameter that is a string (up to 40 bytes) that DFSMSShsm maintains with each backup version of the copy pool. The token may be used to identify a specific backup version to dump or recover. The token is returned as part of the LIST and ARCXTRCT data for the copy pool backup version. If the value specified for TOKEN is less than 40 bytes, the TOKEN value is left justified and binary zeros fill the remainder of the positions to fill the 40 byte space. If TOKEN is not specified, binary zeros are returned with the data.

Defaults: None.

Examples of how to code the FRBACKUP command

The following examples present different ways to code the FRBACKUP command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Creating a fast replication dump copy

In this example, DFSMSShsm will initiate a fast replication DASD copy for each volume and subsequently dump the target DASD volumes to a onsite dump class and an offsite dump class.

```
FRBACKUP COPYPOOL(CPDATABAS4) DUMP DCLASS(ONSITE,OFFSITE)
```

Dumping an existing valid DASD version

In this example, DFSMSShsm will initiate dump processing for an existing DASD version.

```
FRBACKUP COPYPOOL(CPDATABAS5) DUMPONLY
```

Validating the current environment

In this example, DFSMSShsm submits a request for validation of the current environment. DFSMSShsm assigns specific target volumes to each source volume for each version.

```
FRBACKUP COPYPOOL(CPDATABASE3) PREPARE
```


Creating a fast replication backup for each volume in the copy pool's storage groups

In this example, DFSMSHsm creates a fast replication backup version.

```
FRBACKUP COPYPOOL(CPDATABASE2) EXECUTE TOKEN(FRBACKUP_JULYDATA)
```

Withdrawing copy pool relationships

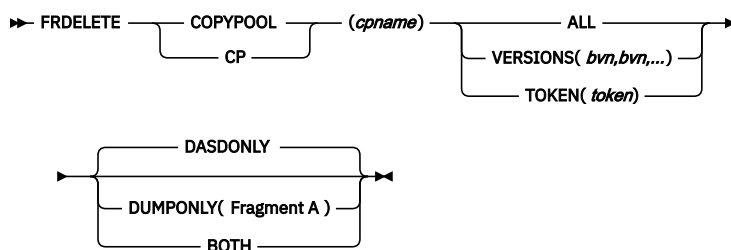
In this example, DFSMSHsm submits a request to withdraw outstanding FlashCopy relationships that were established for the previous backup version. This invalidates the previous backup version.

```
FRBACKUP COPYPOOL(CPDATABASE1) WITHDRAW
```

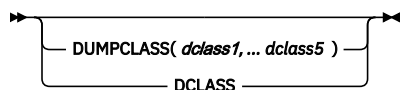

Chapter 47. FRDELETE command: Deleting fast replication backup or dump versions

The FRDELETE command enables you to delete one or more fast replication backup or dump versions. This is useful when you rename a copy pool, when you no longer need a copy pool, or when you no longer need an individual backup or dump version. Before a version is deleted, any outstanding FlashCopy relationships are withdrawn.

Syntax of the FRDELETE command



A: DUMPONLY Optional Parameters



Note: You must specify either ALL, TOKEN, or VERSIONS with each FRDELETE command.

Required parameters of the FRDELETE command

This topic describes the required parameters of the FRDELETE command.

COPYPOOL: Specifying the name of the copy pool for deletion

Explanation

COPYPOOL(*cpname*) is used to identify the copy pool for which backup versions are deleted. For *cpname*, substitute the name of the copy pool for which to delete the backup version.

Defaults

None

ALL, TOKEN, and VERSIONS: Specifying the delete action

Explanation

ALL, TOKEN, and VERSIONS are mutually exclusive parameters that indicate to DFSMSHsm the type of delete action that you are initiating.

ALL

DFSMSHsm is to delete all fast replication backup versions.

TOKEN(*token*)

The token of the backup version to delete. If the token is not unique, then DFSMSHsm deletes the most recent backup version with the specified token.

token

Alphanumeric or hexadecimal token.

VERSIONS(*bvn,bvn,...*)

The backup versions to delete.

bvn,bvn,...

Number of the backup versions.

Defaults

None.

Optional parameters of the FRDELETE command

This topic describes the optional parameters of the FRDELETE command.

DASDONLY, DUMPONLY, and BOTH: Specifying which copies to delete

Explanation

DASDONLY, DUMPONLY, and BOTH are mutually exclusive, optional parameters that indicate to DFSMSHsm which copies to delete.

DASDONLY

DFSMSHsm is to delete only the DASD copies for the specified copy pool.

DUMPONLY

DFSMSHsm is to delete only the dump copies for the specified copy pool.

DUMPCCLASS**DCLASS**

Optional parameter that specifies to DFSMSHsm which specific dump classes to delete.

dclass1, ... dclass5

Up to five dump class names, each separated by a comma. If a dump class name is listed multiple times, DFSMSHsm fails the command.

BOTH

DFSMSHsm is to delete both DASD and dump copies for the specified copy pool.

Defaults

If you do not specify DUMPONLY, DASDONLY, or BOTH, the default is DASDONLY.

Usage notes:

1. Unlike autodump expiration, the FRDELETE command does not keep the last indicated version.
2. If the DUMPCCLASS parameter is not specified, DFSMSHsm deletes all dump copies associated with the specification of the VERSIONS, TOKEN, or ALL parameters.

Examples of how to code the FRDELETE command

The following examples present different ways to code the FRDELETE command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Deleting all versions of a fast replication backup

In this example, DFSMSHsm deletes all fast replication backup versions for both DASD and dump copies for the specified copy pool.


```
FRDELETE COPYPOOL(CPDATABASE1) ALL BOTH
```

Deleting one version of a fast replication backup by version number

In this example, DFSMSHsm deletes one fast replication backup version for a copy pool. The specific version number is specified.

```
FRDELETE COPYPOOL(CPDATABASE1) VERSIONS(2)
```

Deleting one version of a fast replication backup by token

In this example, DFSMSHsm deletes the fast replication backup version for a copy pool. The specific token is specified.

```
FRDELETE COPYPOOL(CPDATABASE1) TOKEN(53800)
```

Deleting a dump class from a copy pool dump version

In this example a single dump class is removed from a copy pool dump version.

```
FRDELETE COPYPOOL(CPDATABAS1) TOKEN(53801) DUMPNLY(DUMPCCLASS(DCL1))
```

Deleting only the DASD version for two copy pool versions

In this example, version 5 and 6 DASD copy of a copy pool will be deleted

```
FRDELETE COPYPOOL(CPDATABAS1) VERSIONS(5,6) DASDONLY
```


Chapter 48. FREEVOL command: Freeing DFSMSHsm-owned volumes

The FREEVOL command allows you to empty a migration or backup DASD volume in preparation for new equipment or to replace old or partially damaged volumes. If the intended use is to make more space available on an ML1 volume, use the MIGRATE ML1 command instead. In addition, FREEVOL removes selected data sets from a volume based on management class attributes or age. FREEVOL moves valid backup versions and migration copies from DFSMSHsm-owned DASD volumes to either tape or DASD.

The FREEVOL command moves migration copies of SMS-managed data sets from:

- A migration level 1 volume based on each data set's management class attribute values
- A migration level 2 volume based on the age specified on the AGE parameter of the FREEVOL command.

The FREEVOL command moves all migration copies of non-SMS-managed data sets meeting a specified age criterion from DFSMSHsm migration volumes. Migration copies meeting the specified age criterion are:

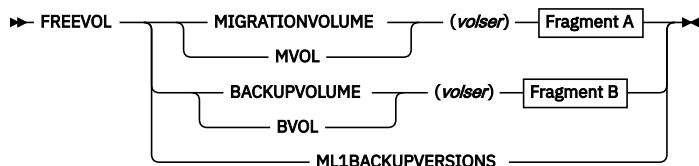
- Moved from a specified migration level 1 volume to other migration level 1 or level 2 volumes
- Moved from a specified migration level 2 DASD volume to other migration level 2 volumes.

The FREEVOL command moves all backup versions of SMS-managed and non-SMS-managed data sets meeting a specified age criterion from DFSMSHsm daily backup volumes. Backup versions are moved from a specified backup volume to spill DASD or tape volumes. A parameter can be specified to prevent the movement of the most recent backup version for each data set.

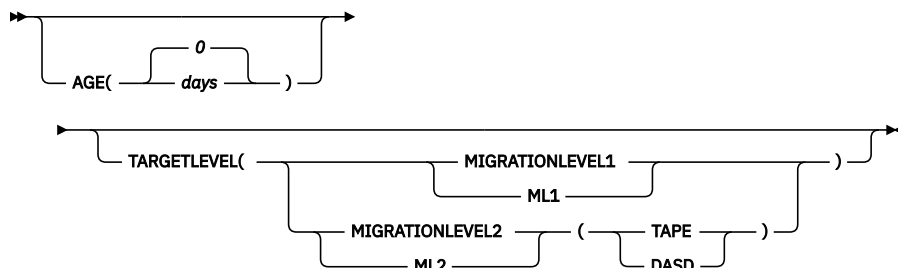
Note:

1. FREEVOL can be used to move migration copies from one unit type to another unit type.
2. Each time automatic backup processing runs, backup copies are moved off migration level 1 volumes. FREEVOL ML1BACKUPVERSIONS provides this same function. However, FREEVOL of a single migration volume does not move backup versions.
3. FREEVOL does not move user data sets, SYS1.VVDS, or the VTOC index data set or user catalogs.
4. FREEVOL moves catalog information data sets to another ML1 volume, regardless of the target level of the FREEVOL command.
5. FREEVOL of a migration level 1 volume moves VTOC copy data sets on the freed volume to other migration level 1 volumes, regardless of the target level of the FREEVOL command.
6. Unless you want DFSMSHsm to continue to use the freed *migration* volume as a target for migration, you should either delete the volume or add it again with the DRAIN attribute.
7. Unless you want DFSMSHsm to continue to use the freed *backup* volume as a target for backup, you should delete the volume.
8. For migrated SMS data sets, the values specified for the management class attribute of *COMMAND-OR-AUTO-MIGRATE* are ignored.
9. FREEVOL does not move any SDSP data sets, although it can move or delete any data sets that are in an SDSP data set. Small data sets that move to another ML1 volume may go into an SDSP data set. Refer to [“Managing SDSP data sets” on page 555](#) for removing an SDSP.
10. When FREEVOL AGE(0) is specified, BACKDS backup versions and data sets that have migrated but are awaiting automatic backup are not moved off the volume. Running automatic backup on the DFSMSHsm primary host performs this function. When you are planning to remove an ML1 volume, run autobackup on the primary host shortly before running the FREEVOL function with the AGE(0) parameter.

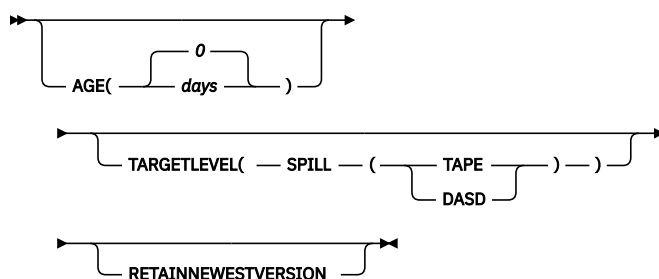
Syntax of the FREEVOL command



A: MIGRATIONVOLUME optional parameters



B: BACKUPVOLUME optional parameters



Required parameters of the FREEVOL command

This topic describes the required parameters of the FREEVOL command.

MIGRATIONVOLUME and BACKUPVOLUME: Specifying a volume to free

Explanation: **MIGRATIONVOLUME(volser)** | **BACKUPVOLUME(volser)** are mutually exclusive, required parameters. Issue one or the other with the FREEVOL command when a volume serial number is specified. For (volser), substitute the serial number of the migration or daily backup volume you want processed.

Defaults: None.

Note:

1. When you are using MIGRATIONVOLUME, all data set movement statistics appear as SUBSEQUENT MIGS on the REPORT summary.
2. When you are using BACKUPVOLUME, all data set movement statistics appear as SUBSEQUENT BACKUP on the REPORT summary.

ML1BACKUPVERSIONS: Moving backup versions off migration volumes

Explanation: **ML1BACKUPVERSIONS** is mutually exclusive with the MIGRATIONVOLUME and BACKUPVOLUME parameters. There are no subparameters specified with this parameter.

When this parameter is specified, backup versions are moved from all migration level 1 volumes to daily backup volumes. This processing occurs on a primary or secondary host, but does not allow processing from two hosts concurrently. While this processing occurs, volume backups are not permitted to start

on this host. This parameter does not support wait processing in the HSEND CMD command or the ARCHSEND invocation macro.

All data set movement statistics appear as SUBSEQUENT BACKUP on the report summary.

Defaults: None.

Optional parameters of the FREEVOL command

This topic describes the optional parameters of the FREEVOL command.

AGE: Specifying data set use

Explanation: **AGE(days)** is an optional parameter specifying the age criterion for moving data sets.

- When processing a migration volume, substitute a decimal value from 0 to 999 for the number of days since data sets were last referred to.
- When processing a backup volume, substitute a decimal value from 0 to 999 for the number of days since the backup version was created. If the RETAINNEWESTVERSION subparameter is specified, the newest versions created are retained on the volume independent of the age criterion specified.

If AGE(0) is entered for a migration level 1 volume, migration copies of all SMS-managed data sets are moved from the volume and placed on other migration level 1 volumes or migration level 2 volumes, depending on each data set's management class. If the management class has a value specified for the LEVEL-1-DAYS-NONUSAGE and the age is met, the data set migrates to a migration level 2 volume. If the management class has a value specified for LEVEL-1-DAYS-NONUSAGE and the data set age does not meet the criterion or if the data set has an attribute of NOLIMIT, the data set migrates to another migration level 1 volume.

If an AGE other than zero is specified for migration level 1 volumes, the management class for each SMS data set is used to determine whether the migration copy is moved to another volume.

If AGE (0–999) is entered for a migration level 2 volume, the age specified applies to all of the data sets on the volume, because there are no management class attributes that apply to SMS data sets on migration level 2 volumes.

If AGE (0–999) is entered for a backup volume, the age specified applies to all of the data sets on the volume. Expiration processing of backup versions is not performed during FREEVOL.

Defaults: If you do not specify AGE, the default is zero. A default of zero clears the volume of all migrated data except the ones in need of backup, or it clears the backup volume if the RETAINNEWESTVERSION subparameter is not specified.

TARGETLEVEL: Specifying the target to which data sets are moved

Explanation: **TARGETLEVEL** specifies the level and the unit type for migration copies that are eligible to be moved.

For SMS-Managed Migrated Data Sets: The target levels of MIGRATIONLEVEL1 and MIGRATIONLEVEL2 do not apply because their target levels are determined by management class as described in the AGE parameter description. The device category of DASD or TAPE applies only to those SMS-managed data sets migrating to ML2:

Subparameter	Explanation
DASD	TARGETLEVEL (MIGRATIONLEVEL2 (DASD)) specifies that the eligible data sets migrate from MIGRATIONVOLUME(<i>volser</i>) to DASD migration level 2 volumes.
TAPE	TARGETLEVEL (MIGRATIONLEVEL2 (TAPE)) specifies that the eligible data sets migrate from MIGRATIONVOLUME(<i>volser</i>) to tape migration level 2 volumes.

For Non-SMS-Managed Migrated Data Sets: The **TARGETLEVEL (MIGRATIONLEVEL1 | MIGRATIONLEVEL2 (TAPE | DASD))** parameters apply to these migrated data sets.

Subparameter	Explanation
MIGRATIONLEVEL1	TARGETLEVEL (MIGRATIONLEVEL1) specifies that volumes defined as migration level 1 receive migrated data sets from <code>MIGRATIONVOLUME(volser)</code> .
MIGRATIONLEVEL2	<p>TARGETLEVEL (MIGRATIONLEVEL2) specifies that volumes defined for migration level 2 receive migrated data sets from <code>MIGRATIONVOLUME(volser)</code>.</p> <p>TARGETLEVEL (MIGRATIONLEVEL2 (DASD)) specifies that the eligible data sets migrate from <code>MIGRATIONVOLUME(volser)</code> to DASD migration level 2 volumes.</p> <p>TARGETLEVEL (MIGRATIONLEVEL2 (TAPE)) specifies that the eligible data sets migrate from <code>MIGRATIONVOLUME(volser)</code> to tape migration level 2 volumes.</p>

For BACKUP Data Sets: The **TARGETLEVEL(SPILL(TAPE | DASD))** parameters are used to specify the level and the unit type for backup versions eligible to be moved. The device category applies to all data sets moving to SPILL backup volumes:

Subparameter	Explanation
DASD	TARGETLEVEL(SPILL(DASD)) specifies that the eligible data sets move from <code>BACKUPVOLUME(volser)</code> to DASD spill backup volumes.
TAPE	TARGETLEVEL(SPILL(TAPE)) specifies that the eligible data sets move from <code>BACKUPVOLUME(volser)</code> to tape spill backup volumes.

Defaults: If you do not specify either DASD or TAPE, the default is the type of migration set up by the SETSYS command. For migration volumes, the TAPEMIGRATION parameter is used to set up this default. For backup volumes, the SPILL parameter is used.

Note:

1. If `MIGRATIONVOLUME` and `TARGETLEVEL` are the same level, the current source volume (`MIGRATIONVOLUME(volser)`) is excluded from the target volume selection process.
2. If the source volume you specified is a migration level 2 volume, the specification of a target level migration level 1 volume is invalid.
3. If you do not specify `TARGETLEVEL`, migration copies of non-SMS-managed data sets are moved to migration level 2 volumes.
4. If you specify either DASD or TAPE, the one selected overrides the environment specified with the SETSYS TAPEMIGRATION command or the SETSYS SPILL command.
5. After a FREEVOL command has been processed against a migration level 2 volume with an associated key range, that volume must be further processed by a DELVOL and an ADDVOL command in order to become eligible again as a target volume for that key range.

RETAINNEWESTVERSION: Preventing movement of backup versions

Explanation: **RETAINNEWESTVERSION** specifies that the FREEVOL process not move the most recent backup version of a data set from the volume.

If both AGE and RETAINNEWESTVERSION are specified, the most recent backup versions are retained on the current volume, regardless of the age criterion.

Defaults: If you do not specify this parameter, all backup versions are selected by the number of days specified by the AGE parameter.

Examples of how to code the FREEVOL command

Coding of the FREEVOL command is demonstrated in the examples that follow.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Moving migration copies from one migration volume to other migration volumes

In this example, migration copies of SMS-managed data sets are moved from a specific migration volume to migration level 1 or migration level 2 DASD volumes based on their management class attributes. Non-SMS-managed data sets are moved from a specific migration volume to migration level 2 DASD volumes.

```
FREEVOL MIGRATIONVOL(MIG100) TARGETLEVEL(ML2(DASD))
```

Moving backup versions from one backup volume to tape

In this example, backup versions are moved from a specific backup volume to spill tape volumes.

```
FREEVOL BACKUPVOL(BACK01) TARGETLEVEL(SPILL(TAPE))
```

Moving backup versions selectively

In this example, backup versions are moved from a specific backup volume to the type of volumes set up by the SETSYS SPILL parameter. Backup versions are moved if they were created 30 or more days ago. Backup versions that are the most recently created copies of their source data sets remain on the volume and are not affected by the AGE parameter.

```
FREEVOL BACKUPVOL(BACK02) AGE(30) RETAINNEWEST
```

Moving backup versions from ML1 volumes

In this example, backup versions are moved from all ML1 volumes to daily backup volumes.

```
FREEVOL ML1BACKUPVERSIONS
```

Chapter 49. FRRECOV command: Requesting a fast replication recovery

The FRRECOV command can be used to recover a copy pool or individual volumes and data sets from the managed copy pool copies. The backup copy to be recovered can reside on either DASD or tape. If the backup copy resides on both DASD and tape, the default is to use the DASD backup copy.

To restrict the recovery to only backup copy versions residing on DASD or tape, use the FROMDASD or FROMDUMP options, respectively. If the backup copy version is not found on either DASD or tape, the recovery request will fail.

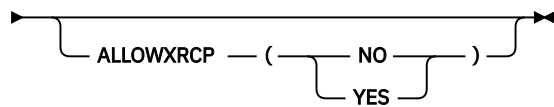
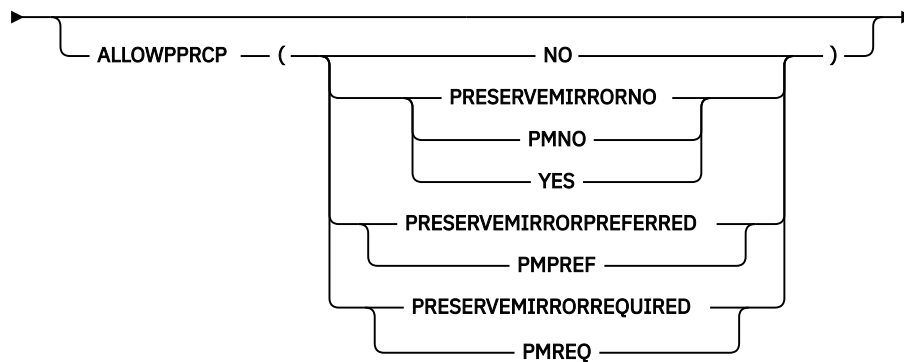
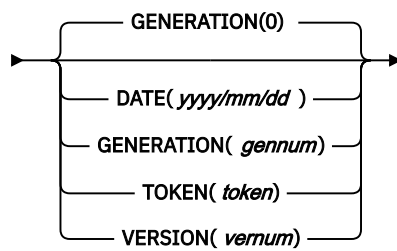
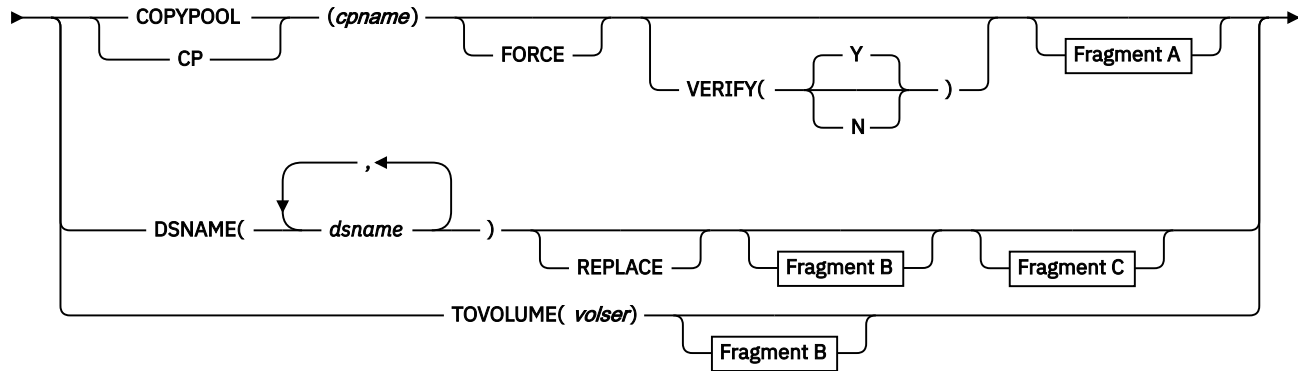
When DATE, GENERATION, TOKEN, or VERSION is specified, the corresponding backup copy will be recovered. If no specific backup copy is specified, an attempt to recover generation zero will occur. If no valid backup copy (either the indicated or implicit) is found, on DASD or tape, the recovery request will fail.

A specific dump class to recover the version from can be specified when recovering from a dump copy on tape. When recovery is performed at the copy pool level, and the dump copy to recover is a partial dump, the recovery request will fail unless the PARTIALOK option is specified.

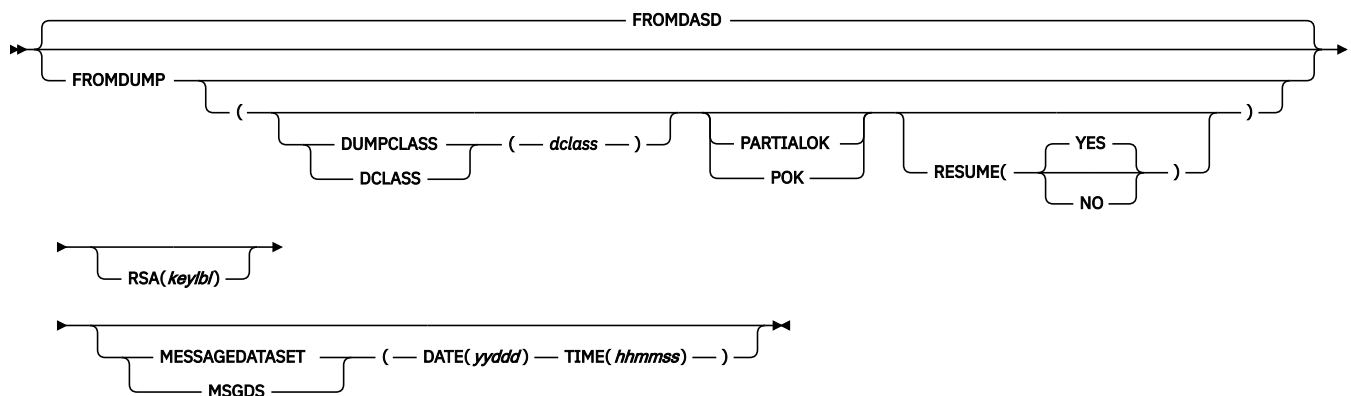
Note: Because fast replication can be used to make volume-level recoveries, there are important factors to consider to ensure that the backup and recovery process restores your data to a usable state. Before using the fast replication function, see [“Managing volume backups with fast replication”](#) on page 254.

Syntax of the FRRECOV command

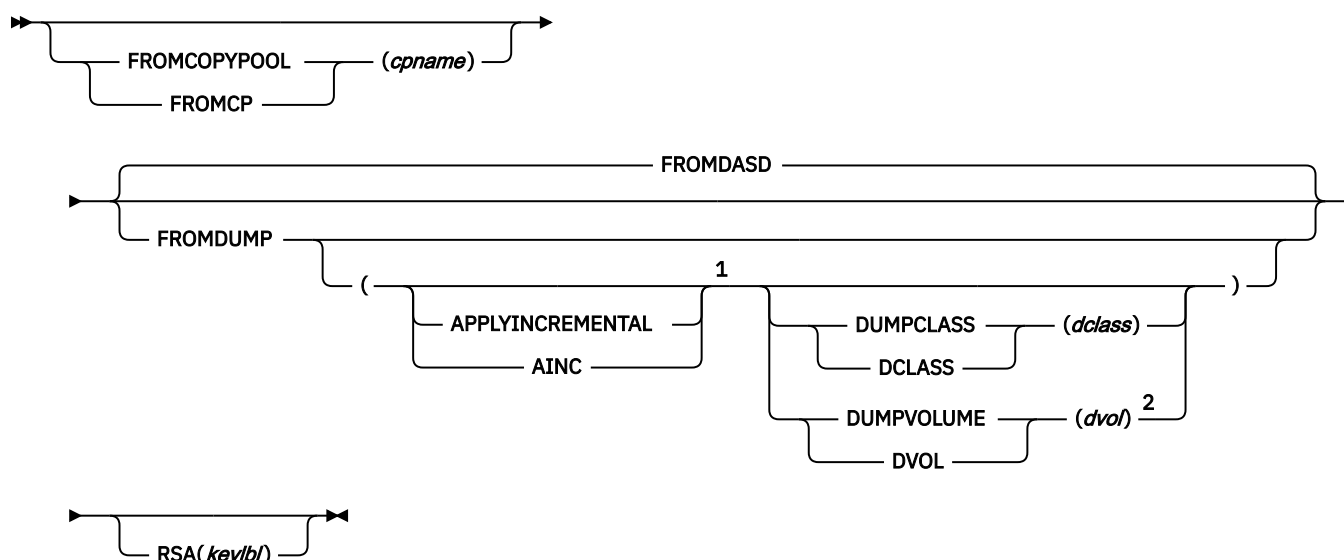
➤ FRRECOV ➤



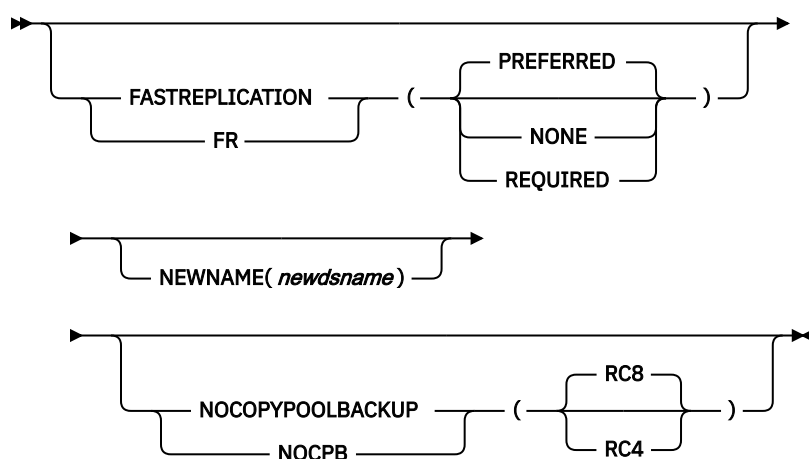
A: COPYPOOL and TOVOLUME optional parameters



B: DSNAME and TOVOLUME optional parameters



C: DSNAME only optional parameters



Notes:

¹ APPLYINCREMENTAL is valid only with the TOVOLUME keyword and is applicable only when the recovery is from a dump volume.

² FROMDUMP(DUMPVOLUME(dvol)) is mutually exclusive with the DATE, GENERATION, TOKEN, and VERSION parameters.

Required parameters of the FRRECOV command

This topic describes the required parameters of the FRRECOV command.

COPYPOOL: Specifying the copy pool for which to recover all backup volumes

Explanation: COPYPOOL is a required parameter that is mutually exclusive with the DSNAME and TOVOLUME parameters. The COPYPOOL parameter indicates to DFSMSHsm to recover all source volumes associated with the named copy pool.

For *cpname*, substitute the name of the copy pool that you want to recover. DFSMSHsm recovers all volumes that are associated with this copy pool.

Defaults: None.

DSNAME: Specifying the data sets to recover from a copy pool backup version

Explanation: DSNAME is a required parameter that is mutually exclusive with the COPYPOOL and TOVOLUME parameters. The DSNAME parameter indicates to DFSMSHsm to recover individual data sets from a copy pool backup version.

For *dsname*, substitute one or more fully-qualified or partially-qualified names of data sets you want to recover (separate each name with a comma). Deleted data sets must be fully-qualified. If you specify a partially-qualified data set name, every cataloged data set in the standard search order that matches the filtering criteria is processed. If one or more data sets are not found, processing continues.

The high-level qualifier of the specified data set name must be fully-qualified or partially-qualified. A double wildcard ("**") is used in place of a full qualifier and a single wildcard ("*") is used within a qualifier. You cannot specify a double wildcard as the high-level qualifier or as the only qualifier. You can specify a single wildcard in the high-level qualifier position if it is not the first character specified. These restrictions help to prevent an unmanageable number of data sets from being processed. The remainder of the data set name can include any arrangement of wildcard characters.

When the optional NEWNAME(*newdsname*) parameter is specified, *dsname* must be a single, fully-qualified data set name.

When GDG data sets are recovered, their bases are serialized. When using a wildcard to recover multiple GDG generations with the same base, they will fail due to serialization errors. It is recommended that multiple GDG generation data sets are recovered individually.

Defaults: None.

TOVOLUME: Specifying a single volume within a copy pool to recover

Explanation: TOVOLUME is a required parameter that is mutually exclusive with the DSNAME and COPYPOOL parameters. The TOVOLUME parameter indicates to DFSMSHsm to recover a single volume within a copy pool.

For *volser*, specify the name of the source volume to be recovered. If you specify a source volume that is contained within a storage group that is defined within multiple copy pools, then you must also specify FROMCOPYPOOL(*cpname*).

You can use the LIST PRIMARYVOLUME BCDS command to determine which copy pools a volume is contained within. You can then use the LIST COPYPOOL(*cpname*) command to determine which of those copy pools contains the version that you want recovered.

Note: Do not use the TOVOLUME operation to recover individual volumes in a copy pool that are defined with fast reverse restore capability.

Defaults: None.

Optional parameters of the FRRECOV command

This topic describes the optional parameters of the FRRECOV command.

ALLOWPPRCP: Specifying Metro Mirror and Global Mirror primary volumes and preserve mirror options for a FlashCopy operation

Explanation: ALLOWPPRCP is an optional parameter that specifies whether a Metro Mirror or Global Mirror primary volume is allowed to become a target of a FlashCopy operation, and whether a preserve mirror operation is required, preferred, or not desired when the target volume is a Metro Mirror primary volume during fast replication recovery. When you specify ALLOWPPRCP, you must specify one of the following options:

NO

Specifies that a Metro Mirror or Global Mirror primary volume is not to become a FlashCopy target. This is the default.

PRESERVEMIRRORNO**YES**

Specifies that a Metro Mirror or Global Mirror primary volume is allowed to become a FlashCopy target, and the FlashCopy is to be performed without considering a preserve mirror operation. If the target volume is not a Metro Mirror or Global Mirror primary volume, this parameter has no effect.

PRESERVEMIRRORPREFERRED

Specifies that a Metro Mirror or Global Mirror primary volume is allowed to become a FlashCopy target and it would be preferable that the Metro Mirror pair does not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, the FlashCopy operation is still to be attempted. If the intended FlashCopy target volume is not a Metro Mirror primary volume, this parameter has no effect.

The IBM DS8880 series and newer IBM storage systems do not support the Preserve Mirror Preferred option for Metro Mirror. If you are using one of those storage systems, and you want to preserve your disk mirror or copy, specify PRESERVEMIRRORREQUIRED.

PRESERVEMIRRORREQUIRED

Specifies that a Metro Mirror or Global Mirror primary volume is allowed to become a FlashCopy target. If the FlashCopy target volume is a Metro Mirror primary device, a Preserve Mirror option is required when performing FlashCopy recovery. If the target volume is not a Metro Mirror or Global Mirror primary volume, this parameter has no effect.

Note:

1. ALLOWPPRCP overrides the stored preserve mirror indicator for the copy pool. The change is not persistent and applies to one invocation of the FRRECOV command only.
2. ALLOWPPRCP is ignored when recovering from a tape.
3. The combined use of fast reverse restore and the preserve mirror operation is not supported. When a copy pool is defined allowing fast reverse restore, and the ALLOWPPRCP(PMREQ) or ALLOWPPRCP(PMPREF) parameter is specified, the FRRECOV command is failed.
4. ALLOWPPRCP options PRESERVEMIRRORREQUIRED, PRESERVEMIRRORPREFERRED, PRESERVEMIRRORNO, and YES cannot be specified in combination with the FASTREPLICATION option of NONE for data set recoveries.

Defaults: If you do not specify ALLOWPPRCP on the FRRECOV command, DFSMSHsm uses the currently stored preserve mirror indicator for the copy pool. If no preserve mirror indicator has been set for the copy pool, the default is NO.

Related reading: For more information about IBM Remote Pair FlashCopy, Metro Mirror, also known as synchronous Peer-to-Peer Remote Copy (PPRC), and other copy services functions, see [z/OS DFSMS Advanced Copy Services](#).

ALLOWXRC: Specifying an extended remote copy primary volume option for a FlashCopy operation

Explanation:

ALLOWXRC is an optional parameter that specifies whether an extended remote copy (XRC) primary volume is allowed to become a target of a FlashCopy operation during fast replication recovery. When you specify ALLOWXRC, you must specify one of the following options:

NO

Specifies that an XRC primary volume is not to become a FlashCopy target. This is the default.

YES

Specifies that an XRC primary volume is allowed to become a FlashCopy target. If the target volume is not an XRC volume, this parameter has no effect.

Note:

1. The ALLOWXRCP parameter overrides the specified setting for the SMS copy pool. The override is not persistent and applies to the current invocation of the FRRECOV command only.
2. ALLOWXRCP is ignored when recovering from a tape.
3. ALLOWXRCP(YES) cannot be specified in combination with the FASTREPLICATION option of NONE for data set recoveries.
4. ALLOWXRCP(YES) cannot be specified in combination with the fast reverse restore option.

Defaults: If you do not specify ALLOWXRCP on the FRRECOV command, DFSMSHsm uses the currently specified option for the SMS copy pool. If no ALLOWXRCP option has been designated for the copy pool, the default is NO.

Related reading: For more information about IBM remote pair FlashCopy, XRC, and other copy services functions, see [z/OS DFSMS Advanced Copy Services](#).

DATE, GENERATION, TOKEN, VERSION: Specifying a specific backup version to recover

Explanation: DATE, GENERATION, TOKEN, and VERSION are mutually exclusive, optional parameters that indicate to DFSMSHsm a specific backup copy to recover.

DATE

Specify DATE(*yyyy/mm/dd*) to recover a backup copy from a particular date. If more than one copy exists for the specified date, then DFSMSHsm recovers the most recent copy from that date. The FRRECOV command fails if no copy exists for the specified date. A copy may not exist because either no valid backup copy exists or the copy exists, but it is invalid. You do not need to include a leading zero to specify single digit months and days.

GENERATION

Specify GENERATION(*gennum*) to recover a backup copy of a particular generation. The generation number can be from 0 to 84. The most recent copy is generation 0, the next is 1, and so on.

TOKEN

Specify TOKEN(*token*) to recover a backup copy with a particular token. If the token is not unique, then the most recent backup copy with the specified token is recovered. If the specified token is less than 40 characters, then the token is left justified and padded with binary zeroes before the backup copy search. You can specify a hexadecimal token using the X'...' syntax.

VERSION

Specify VERSION(*vernum*) to recover a backup of a particular copy. The copy number can be from 1 to 999.

Defaults: If you do not specify DATE, GENERATION, TOKEN, or VERSION, DFSMSHsm uses the most recent backup copy (generation zero) to perform the recovery.

FASTREPLICATION: Indicating a recovery preference

Explanation: FASTREPLICATION is an optional parameter that indicates whether the use of fast replication for data set recovery is required, preferred, or not desired.

The following are parameters of the FASTREPLICATION parameter:

PREFERRED

Indicates that fast replication should be used when possible. If fast replication cannot be used, the data set is recovered through traditional copy methods.

REQUIRED

Indicates that if fast replication cannot be used, the recovery is to fail.

NONE

Indicates that only traditional copy methods are to be used.

Defaults: None.

Notes:

1. Using the FASTREPLICATION parameter overrides the FASTREPLICATION value specified using the SETSYS FASTREPLICATION command. If not specified, the value of SETSYS FASTREPLICATION is used.
2. FASTREPLICATION is valid only for recovery from disk. If the recovery is from tape, the parameter is ignored.
3. FASTREPLICATION is valid with DSNAME only. It is not valid with the COPYPOOL or TOVOLUME parameters.
4. To recover data sets only when fast replication can be used, specify FROMDASD with FASTREPLICATION(REQUIRED).

FORCE: Allowing recovery from a DASD backup copy that has an incomplete dump copy

Explanation: FORCE is an optional parameter that allows you to force fast replication recovery of a copy pool from a DASD backup copy with an in-progress dump to tape or otherwise incomplete dump copy. Any in-progress dump to tape of the DASD backup copy is terminated. In addition, the FORCE parameter also allows the recovery of a copy pool to proceed when the recovery operation (such as fast reverse restore) will invalidate the DASD backup copy. In a situation where the DASD backup copy is invalidated, the incomplete dump cannot be resumed.

Defaults: None.

FROMCOPYPOOL: Recovering a volume or data set that belongs to multiple copy pools

Explanation: FROMCOPYPOOL(*cpname*) is an optional parameter that you must specify if the volume or data set you want DFSMSHsm to recover belongs to a storage group that is defined within multiple copy pools. For *cpname*, substitute the name of the copy pool that contains the volume. When performing a data set recovery, DFSMSHsm checks the catalog to determine the volumes on which the data set resides. Based on the first volume, DFSMSHsm determines which copy pool the volume belongs to. If the volume belongs to more than one copy pool, you must specify the FROMCOPYPOOL(*cpname*) keyword. The FROMCOPYPOOL is a required keyword for deleted data sets. If the data set is uncataloged, but exists on a volume, you must specify both FROMCOPYPOOL and REPLACE keywords in order to recover the data set.

Defaults: None.

FROMDASD: Recovering a volume from DASD

Explanation: FROMDASD is an optional parameter that indicates to DFSMSHsm to recover the specified volume from DASD.

Defaults: FROMDASD is the default. If a volume version exists both on DASD and tape, DFSMSHsm recovers the DASD copy.

FROMDUMP: Recovering a volume from a dump tape

Explanation: FROMDUMP is an optional parameter that indicates to DFSMSHsm to recover the specified volume from a dump tape.

The following options can be used with the FROMDUMP parameter:

APPLYINCREMENTAL

Specifies that an incremental volume recovery process is to follow immediately after the volume restore and is to be part of the single request to recover a specific volume.

DUMPCCLASS

Specifies which dump class to use for the volume recovery.

dclass

Is the name of the dump class to use for the volume recovery.

DUMPVOLUME

Specifies which specific dump copy to use for the volume recovery.

dvol

Is the name of the dump volume to use for volume recovery.

PARTIALOK

Allows recovery of a backup copy from a partial dump version. If the requested dump version is complete, that is, not a partial dump version, the PARTIALOK parameter is ignored.

RESUME

Specifies whether resuming a previously failed recovery is attempted. If the previous recovery was successful, the RESUME parameter is ignored.

NO

Specifies a previously failed recovery should not be resumed. A full recovery will be attempted instead.

YES

Specifies a previously failed recovery should be resumed. This is the default.

Note:

1. APPLYINCREMENTAL can be specified only when recovering a specific volume (TOVOLUME) from a dump tape copy.
2. PARTIALOK and RESUME can be specified only when recovering a copy pool from a dump tape copy.

Defaults: If the version to be recovered resides on a dump tape only, DFSMSHsm recovers the version from tape. If the version exists both on DASD and tape, DFSMSHsm recovers the DASD copy.

MESSAGEDATASET: Specifying the date and time used in the message data set name

Explanation: MESSAGEDATASET is an optional parameter that allows you to specify the date and time used in the name of the fast replication command message data set. Specifying the date and time ensures that you can identify the message data set that corresponds with the FRRECOV COPYPOOL or TOVOLUME command.

The value for the DATE parameter must be a five-digit numeric value, *yyddd*.

The value for the TIME parameter must be a six-digit numeric value, *hhmmss*.

When DATE and TIME are specified on the FRRECOV COPYPOOL or TOVOLUME command and a duplicate message data set already exists with the specified date and time, DFSMSHsm issues message ARC1873I and continues without capturing the fast replication messages in a message data set.

When the DATE and TIME keywords are not specified on the FRRECOV command, DFSMSHsm obtains the date and time that message data set name was generated.

Use of the message data set is controlled with the MESSAGEDATASET parameter of the SETSYS FASTREPLICATION command. Refer to [Chapter 65, “SETSYS command: Establishing or changing the values of DFSMSHsm control parameters,” on page 1057](#) for more information.

Defaults: None.

NEWNAME: Specifying a new data set name for the recovered data set

Explanation: NEWNAME(*newdsname*) is an optional parameter specifying a new, fully qualified data set name for the recovered backup version or dump copy. For *newdsname*, specify a new, fully-qualified data set name for the recovered data set. The originally-named data set will be preserved for analysis or other purposes.

If a cataloged data set exists with the same name as the new name you are specifying, you must specify the REPLACE parameter to replace the existing cataloged data set.

Note:

1. The NEWNAME parameter applies only to data set recovery. If you specify NEWNAME when it does not apply, DFSMSHsm ignores it.
2. You must specify NEWNAME when a data set exists with the same data set name as the data set that you are recovering, and you want to keep both versions of the data set.
3. A GDG base cannot be recovered using fast replication data set recovery. If you specify the NEWNAME keyword when you recover a generation data set (GDS) to a new name, the new GDG base name must have already been defined.
4. During FRRECOV, when NEWNAME is used, DFSMS ACS routines are not called. The data set will be recovered to a volume in the same storage group by using the source data set's DFSMS constructs.

Defaults: None.

NOCOPYPOOLBACKUP: Setting an alternate return code

Explanation: NOCOPYPOOLBACKUP is an optional parameter that indicates an alternate return code value for failed data set recovery requests.

If you specify a partially qualified data set name on DSNNAME and the copy pool backup version cannot be found for a data set that matches the filtering criteria, the default return code is eight. In this case, however, you might not expect all of the data sets to be found in a copy pool backup version. If so, you can specify NOCOPYPOOLBACKUP(RC4) to have a lower return code for this situation (return code four instead of eight).

NOCOPYPOOLBACKUP is valid with DSNNAME only. It is not valid with the COPYPOOL or TOVOLUME parameters.

Defaults: RC8.

REPLACE: Recovering a cataloged data set

Explanation: REPLACE is an optional parameter, but required for recovering cataloged data sets. If the data set is uncataloged, but exists on a volume, you must specify both the REPLACE and FROMCOPYPOOL parameters in order to recover the data set.

If you specify the NEWNAME(*newdsname*) parameter and the new name is the same as the name of an existing data set, you must specify the REPLACE parameter. If REPLACE is not specified, DFSMSHsm fails the FRRECOV DSNNAME command.

Defaults: None.

RSA: Overriding the RSA key label

Explanation: RSA is an optional parameter used to override the RSA key label that DFSMSdss would otherwise use to restore an encrypted dump.

The following variable is required when using the RSA parameter:

keylbl

Is the key label, up to 64 characters in length, used to restore an encrypted dump. The first character must be alphabetic or national.

Defaults: None.

VERIFY: Verifying that no unexpected FlashCopy relationship exists for a volume in a copy pool

Explanation: VERIFY is an optional parameter that specifies whether verification, of conditions which might prevent a successful fast replication recovery, is performed *before* starting a fast replication recovery operation. Performing verification reduces the risk of a fast replication recovery failure. The following condition is verified:

1. The copy pool is defined allowing fast reverse restore, but one or more DASD are in the process of being dumped to tape.

The following conditions are verified when the devices do not support cascaded FlashCopy:

1. The copy pool is not defined allowing fast reverse restore, but one or more volumes are in an existing FlashCopy relationship.
2. The copy pool is defined allowing fast reverse restore, but there are multiple FlashCopy targets, including non-DFSMSHsm managed targets, associated with a source volume in the copy pool. Additional FlashCopy targets need to be removed before attempting fast reverse restore when cascaded FlashCopy is not supported.
3. There are multiple FlashCopy Incremental FlashCopy targets associated with a source volume in the copy pool.

The following options can be used with the VERIFY parameter:

Y

specifies that source volumes require verification. This option is the default.

N

specifies that source volumes do not require verification. Use VERIFY(N) only when all volumes are SnapShot capable, or if the QUERY COPYPOOL command has been issued to verify that there are not any unexpected FlashCopy relationships.

Note:

1. VERIFY(N) cannot be specified when a copy pool is defined allowing fast reverse restore unless a previous FRRECOV COPYPOOL operation has determined the copy pool backup version is no longer eligible for fast reverse restore and regular fast replication recovery can be used.

Defaults: If you do not specify the VERIFY parameter, the default is Y (verification is performed).

Examples of how to code the FRRECOV command

The following examples present different ways to code the FRRECOV command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Recovering a volume from a dump copy for dump class DCL1

In this example a single volume is recovered from dump class DCL1.

```
FRRECOV TOVOLUME(SRC001) FROMDUMP DUMPCLASS(DCL1) GENERATION(0)
```

Recovering a single volume is recovered from a specific dump tape

In this example, a specific volume from a specific copy pool from a specific dump volume is recovered.

```
FRRECOV TOVOLUME(SRC002) FROMCOPYPOOL(CPDATABASE1) FROMDUMP(DUMPVOLUME(TDUMP1))
```


Recovering all data sets for multiple users

In this example, DFSMSHsm requests a fast replication recovery of all data sets for three specific users.

```
FRRECOV DSNNAME(USER1.** ,USER2.** ,USER3.** ) REPLACE
```

Recovering each volume in a copy pool

In this example, DFSMSHsm requests a fast replication recovery for each volume in a copy pool named CPDATABASE1. The most recent versions are recovered since a specific version is not requested.

```
FRRECOV COPYPOOL(CPDATABASE1) VERIFY(Y)
```

Recovering each volume in a copy pool from a specific date

In this example, DFSMSHsm requests a fast replication recovery for each volume in a copy pool named CPDATABASE1 from a specific date.

```
FRRECOV COPYPOOL(CPDATABASE1) DATE(2002/1/15)
```

Recovering volumes in a copy pool with a specific token

In this example, DFSMSHsm requests a fast replication recovery of each volume in a copy pool named CPDATABASE1 with a specific token.

```
FRRECOV COPYPOOL(CPDATABASE1) TOKEN(FRBACKUP_JULYDATA)
```

Recovering a specific source volume

In this example, DFSMSHsm requests a fast replication recovery of the most recent fast replication backup version to a specific source volume.

```
FRRECOV TOVOLUME(VOL186)
```

Recovering a specific source volume from a specific copy pool

In this example, DFSMSHsm requests a fast replication recovery for a specific source volume from the copy pool named CPDATABASE1.

```
FRRECOV TOVOLUME(VOL186) FROMCOPYPOOL(CPDATABASE1)
```

Recovering a specific generation of a source volume from a specific copy pool

In this example, DFSMSHsm requests a fast replication recovery for a specific generation of a source volume from the copy pool named CPDATABASE1.

```
FRRECOV TOVOLUME(VOL186) FROMCOPYPOOL(CPDATABASE1) GENERATION(1)
```

Resuming a failed copy pool recovery from dump

In this example, the previous recovery of a copy pool named CPDATABASE1 from dump class DCL1 failed. The failed recovery is resumed.

```
FRRECOV COPYPOOL(CPDATABASE1) FROMDUMP(DUMPCCLASS(DCL1) RESUME(YES))
```

Chapter 50. HOLD command: Preventing DFSMSHsm functions from running

The HOLD command prevents all or some DFSMSHsm functions from running. For example, you specify HOLD when a temporary software or hardware problem occurs that can be compounded if data movement continues in DFSMSHsm.

Migration, backup, dump, recovery, and restore have an optional keyword related to stopping volume processing after the current data set (EOD) or delaying until the end of processing the user volume (EOV).

Note:

- HOLD ALL will hold the sub-function of AUTOEXPIRATION of data sets beginning with PTF UA99661 in 2.4 and above.
- HOLD MIGRATION or HOLD MIGRATION(AUTO) will not hold AUTOEXPIRATION. You must use HOLD AUTOEXPIRATION if you want this function held.

The WAIT and NOWAIT parameters of the HSEND CMD command have implications to HOLD command processing that are function specific. Refer to the specific processing function for an explanation.

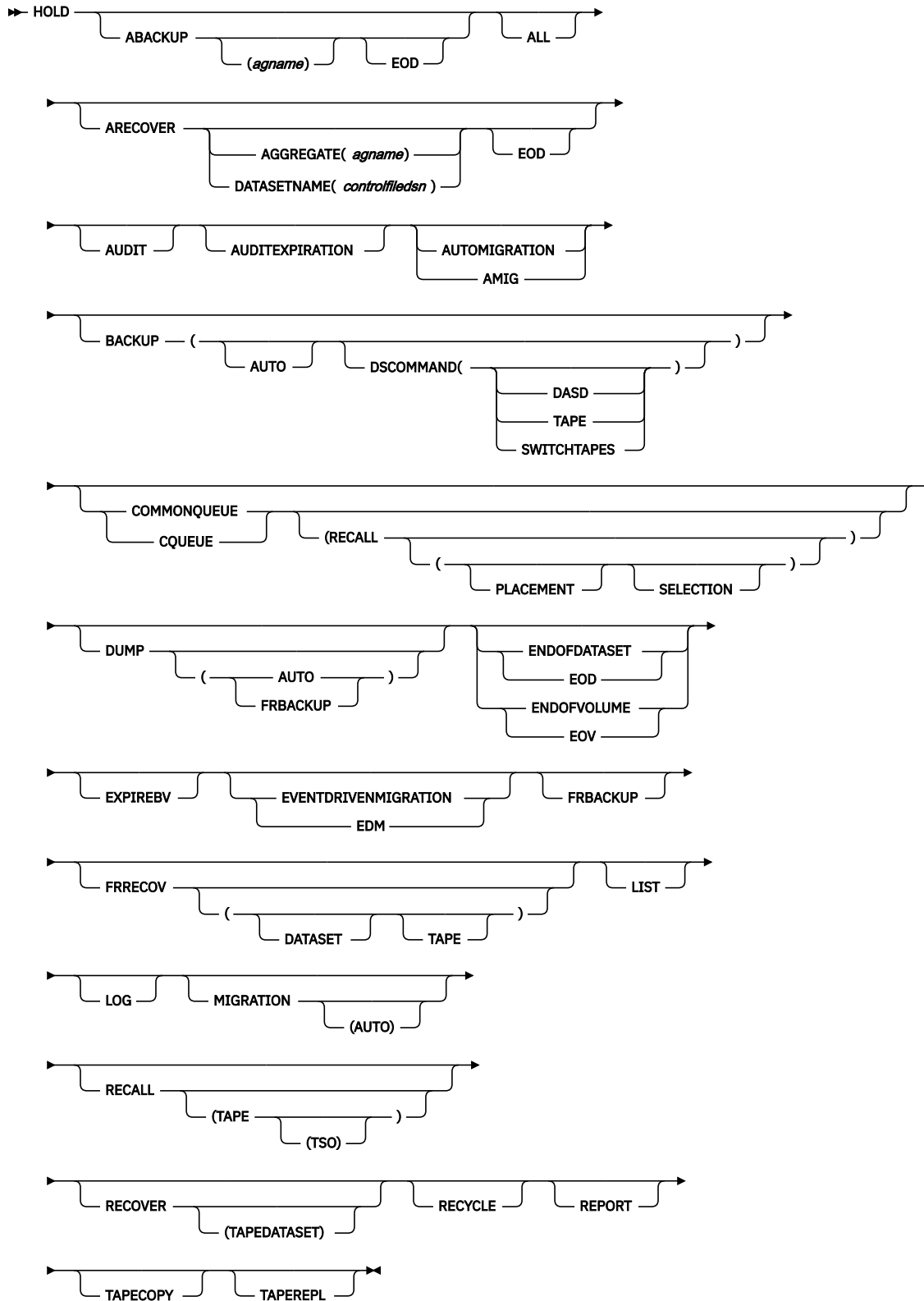
All nonWAIT-type requests remain on the processing queue when the appropriate HOLD command is in effect.

If the HOLD command is issued while the TAPECOPY or TAPER EPL commands are processing more than one volume, either from a volume list from the command or from an input data set, processing stops after the current volume.

You can issue the RELEASE command to release for processing all or some of the DFSMSHsm functions held with the HOLD command.

Functions held from processing by the HOLD command are no longer prevented from processing if DFSMSHsm is reinitialized.

Syntax of the HOLD command



Note:

1. The ENDOFDATASET and ENDOFVOLUME optional parameters affect the following optional parameters:

- AUTOMIGRATION
 - BACKUP
 - BACKUP(AUTO)
 - DUMP
 - DUMP(AUTO)
 - DUMP(FRBACKUP)
 - MIGRATION
 - MIGRATION(AUTO)
 - RECOVER
 - RECOVER(TAPEDATASET)
2. Although the HOLD command has no required parameters, you must specify at least one optional parameter if you want to prevent a DFSMSHsm function from processing.
 3. In a multiple address space for DFSMSHsm environment, ABARS commands, like HOLD ABACKUP or HOLD ARECOVER, can only be issued by the host identified as the main host. DFSMSHsm ignores the ABARS commands if directed to a host identified as HOSTMODE=AUX.

Optional parameters of the HOLD command

This topic describes the optional parameters of the HOLD command.

ABACKUP: Preventing aggregate backup

Explanation: **ABACKUP** is an optional parameter specifying that the process of backing up aggregated data sets stops after completion of any currently running aggregate backup. Any ABACKUP commands not processed remain in the queue until ABACKUP is released.

Subparameter	Explanation
<i>agname</i>	ABACKUP(<i>agname</i>) is an optional parameter specifying an aggregate group. For <i>agname</i> , substitute the name of the SMS aggregate group.

Subparameter	Explanation
EOD	<p>ABACKUP EOD If the EOD option is specified with a HOLD ABACKUP command, the currently executing aggregate backup is stopped at the end of the data set currently processing, and the ABACKUP command is purged from the queue. All open data sets and tapes are closed and all allocated data sets and tapes are deallocated before the termination. The ABACKUP output files are uncataloged and the tape volume serial numbers are removed from the HSMABR tape volume set if they were indicated to be protected. During ABACKUP processing, if the control file has already been written and a HOLD EOD is issued, ABACKUP is allowed to complete normally. However, if ABACKUP was issued with the MOVE option, ABACKUP will be held. If an aggregate group name was specified with the HOLD ABACKUP command, the above processing only occurs when the specified name matches a currently running or pending ABACKUP command.</p> <p>If the ABACKUP parameter is specified with the HOLD command and without the EOD option, the currently executing ABACKUP commands run to completion. If an aggregate group name has been specified, subsequent commands for that name are held. When an aggregate group name was specified, the above processing only occurs if the specified name matches a currently running or pending ABACKUP command.</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. If you issue an ABACKUP command with a WAIT option when ABACKUP is held, that request fails and an error message is issued. 2. If you issue an ABACKUP command with a NOWAIT option when ABACKUP is held, that request is queued and processed when ABACKUP is released. 3. To ensure data integrity, you should HOLD ABACKUP from processing aggregated data sets whose data is being heavily used, or when automatic functions are running.

Defaults: None.

ALL: Preventing all DFSMSHsm functions

Explanation: **ALL** is an optional parameter specifying that all functions controlled by this command, except logging, are prevented.

If the EOD or EOV parameter is specified, then all functions (except ABACKUP, ARECOVER, FRBACKUP, and FRRECOV) are held using that parameter. If neither EOD nor EOV is specified, then each function is held according to its default: BACKUP, MIGRATION, and RECOVER are held as EOD, and DUMP is held as EOV. ABACKUP and ARECOVER ignore the EOD parameter.

Note: HOLD ALL will hold the sub-function of AUTOEXPIRATION of data sets during Auto Migration.

Defaults: None.

ARECOVER: Preventing aggregate recovery

Explanation: **ARECOVER** is an optional parameter specifying that the process of recovering aggregated data sets stop after completion of any currently running aggregate recovery. Any ARECOVER commands not processed remain in the queue until ARECOVER is released.

Subparameter	Explanation
AGGREGATE	<p>ARECOVER AGGREGATE(<i>agname</i>) is an optional parameter specifying an aggregate group. For <i>agname</i>, substitute the name of the SMS aggregate group against which you want DFSMSHsm to hold processing.</p>

Subparameter	Explanation
DATASETNAME	ARECOVER DATASETNAME(<i>controlfiledsn</i>) is an optional parameter specifying a control file name. For <i>controlfiledsn</i> , substitute the name of the control file against which you want DFSMSHsm to hold processing.
EOD	<p>ARECOVER EOD If the EOD option is specified with a HOLD ARECOVER command, the currently executing aggregate recovery is stopped at the end of the data set currently processing, and the ARECOVER command is purged from the queue. All open data sets and tapes are closed and all allocated data sets and tapes are deallocated before the termination. If an aggregate group name or control file data set name has been specified with the HOLD ARECOVER command, the above processing only occurs when the specified name matches a currently running or pending ARECOVER command. During ARECOVER processing, the restart data set is written when the HOLD EOD command is selected so that if the ARECOVER is reissued, data sets already processed are skipped.</p> <p>If the ARECOVER parameter is specified with the HOLD command and without the EOD option, the currently executing ARECOVER commands run to completion. If AGGREGATE(<i>agname</i>) or DATASETNAME(<i>controlfiledsn</i>) has been specified, subsequent commands for that name will be held. When AGGREGATE(<i>agname</i>) or DATASETNAME(<i>controlfiledsn</i>) has been specified, the above processing only occurs if the specified name matches a currently running or pending ARECOVER command.</p> <p>Note:</p> <ol style="list-style-type: none"> 1. If you issue an ARECOVER command with a WAIT option when ARECOVER is held, that request fails and an error message is issued. 2. If you issue an ARECOVER command with a NOWAIT option when ARECOVER is held, that request is queued and processed when ARECOVER is released.

Defaults: None.

AUDIT: Ending the audit function

Explanation: **AUDIT** is an optional parameter specifying that DFSMSHsm stop the audit function after it finishes processing the current data set. WAIT-type command requests fail, and NOWAIT-type command requests remain queued.

Defaults: None.

Note: When you issue the HOLD command, you lose any part of the AUDIT command that has not finished running.

AUTOEXPIRATION: Preventing data sets from expiring

Explanation: **AUTOEXPIRATION** is an optional parameter specifying that DFSMSHsm stop the automatic expiration of data sets during space management.

Defaults: None.

Note: Use caution with this command, as this can lead to volumes being filled to capacity. Short hand for the command is 'AEXP'.

AUTOMIGRATION: Preventing automatic volume and automatic secondary space management

Explanation: **AUTOMIGRATION** is an optional parameter specifying that automatic volume and automatic secondary space management are prevented. If automatic volume and automatic secondary space management are running when you issue the HOLD command, automatic volume and automatic secondary space management stop. The point at which processing stops is controlled by the ENDOFDATASET and ENDOFVOLUME parameters.

Defaults: None.

Note:

1. When you specify **AUTOMIGRATION**, DFSMSHsm does not prevent the running of any space management functions that were issued by command.
2. **HOLD AUTOMIGRATION** has the same meaning as **HOLD MIGRATION(AUTO)**.
3. **HOLD AUTOMIGRATION** will not **HOLD AUTOEXPIRATION**. You must use **HOLD AUTOEXPIRATION** if you want this function held.

BACKUP: Preventing backups

Explanation: **BACKUP** is an optional parameter that specifies whether you want DFSMSHsm to prevent all backup processing or to prevent only the following processing:

- All data set backup requests
- All data set backup requests to either DASD or tape
- Data set backups to DASD
- Data set backups to tape
- All automatic backups

The **BACKUP** parameter also specifies whether you want DFSMSHsm to demount tapes that are used by command backup.

The **BACKUP** parameter, without any subparameters, prevents all backup processing tasks and demounts all tapes mounted for command data set backup. No more tasks are started for backup processing. Any command requests that specified the **WAIT** keyword are failed.

Subparameter	Explanation
AUTO	<p>BACKUP(AUTO) specifies that automatic backup processing is prevented. No more automatic backup processing tasks are started. You can still perform command backup processing tasks.</p> <p>The ENDOFDATASET and ENDOFVOLUME parameters control volume backup processing. Data set processing is always ENDOFDATASET.</p>

Subparameter	Explanation
DSCOMMAND	<p>BACKUP(DSCOMMAND) specifies that all data set backups by command (BACKDS, HBACKDS, ARCHBACK, and ARCINBACK) and all INUSE RETRY backups from autobackup are prevented. All WAIT-type requests fail, and control returns to the requester.</p> <p>BACKUP(DSCOMMAND(DASD)) specifies that only data set backups by command that are directed to ML1 DASD are prevented. All WAIT-type requests fail that specifically target DASD, and the requester is returned control.</p> <p>BACKUP(DSCOMMAND(TAPE)) specifies that only data set backups by command that are directed to tape are prevented. All WAIT-type requests that specifically target tape or are too large for DASD fail, and the requester is returned control. Based on the setting used in the SETSYS DSBACKUP (DASDSELECTIONSIZE) parameter, DFSMSHsm directs some WAIT-type requests to tape. DFSMSHsm fails these WAIT-type requests when you specify DSCOMMAND(TAPE). To redirect these requests to DASD, you can do one of the following tasks:</p> <ul style="list-style-type: none"> • Issue SETSYS DSBACKUP(TAPE(TASKS(0))). This directs to DASD all but those requests that specifically target tape. DFSMSHsm fails those BACKDS requests that specifically target tape. • Increase the SETSYS DSBACKUP(DASDSELECTIONSIZE) settings. This directs to DASD all larger data sets. <p>BACKUP(DSCOMMAND(SWITCHTAPES)) specifies that command data set backup demounts any mounted tapes at the end of the current data set or immediately, if the mounted tape is waiting for additional work. When command data set backup has additional work, processing continues with new tapes. This SWITCHTAPES type of HOLD does not require the use of a RELEASE command.</p> <p>Use the SWITCHTAPES parameter for the following tasks:</p> <ul style="list-style-type: none"> • You have tapes that are continuously mounted, and the need arises to demount a tape so that it or its alternate tape can be taken to another site for disaster backup purposes • To introducing new tape devices to the installation, these continuously mounted tapes also need to be demounted and the drives deallocated <p>If you want to change the PARTIALTAPE options from what was originally defined in the DEFINE SWITCHTAPES(DSBACKUP) command, you must do so before you issue the HOLD command.</p>

Defaults: Issuing the HOLD BACKUP command with no parameters prevents all backup processing.

Note: If you are using the SETSYS BACKUP command or a data set backup installation exit to direct DFSMSHsm to retry backup requests that fail because data sets are in use, and if DFSMSHsm has scheduled any such retries during automatic backup, those backup retries will not be affected by the HOLD BACKUP(AUTO) command. You will need to use HOLD BACKUP or HOLD ALL.

COMMONQUEUE: Preventing common queue functions

Explanation: **COMMONQUEUE** is an optional parameter that specifies that DFSMSHsm prevent all common queue processing. If you specify COMMONQUEUE without any parameters, a host neither places recall requests in the CRQ, nor selects recall requests from the CRQ.

Subparameter	Explanation
RECALL	<p>COMMONQUEUE(RECALL) specifies that a host neither places recall requests in the CRQ, nor selects recall requests from the CRQ.</p> <p>COMMONQUEUE(RECALL(PLACEMENT)) specifies that the host does not place additional requests in the CRQ. All new requests will be placed on the host's local recall queue.</p> <p>COMMONQUEUE(RECALL(SELECTION)) specifies that the host does not select recall requests from the CRQ.</p>

Note: Specifying HOLD COMMONQUEUE without parameters does not affect the common dump queue or recover queue functions.

For more information about using HOLD COMMONQUEUE, see [“Preventing DFSMSHsm functions from running” on page 596.](#)

Defaults: None.

DUMP: Preventing dumps

Explanation: **DUMP** is an optional parameter specifying whether DFSMSHsm prevents all dump processing (command and automatic) or prevents only automatic dump processing. The **DUMP** parameter, without any subparameters, specifies that all dump processing tasks are prevented. No more dump processing tasks are started. Also, any command requests that specified the **WAIT** keyword are failed.

Subparameter	Explanation
AUTO	DUMP(AUTO) specifies that automatic dump processing is prevented. No additional automatic dump processing tasks are started. You can still perform command dump processing tasks. HOLD DUMP(AUTO) also prevents auto dump tasks for fast replication copy pool volumes from starting. This setting does not affect dump tasks started through the FRBACKUP command.
FRBACKUP	DUMP(FRBACKUP) specifies that dump processing of FRBACKUP target volumes is prevented. No additional FRBACKUP DUMP tasks or FRBACKUP DUMPONLY tasks are started.

The point at which DFSMSHsm stops dump processing is controlled by the **ENDOFDATASET** and **ENDOFVOLUME** parameters.

When a **HOLD** command is submitted that interrupts a dump of a copy pool, if it is a **NOWAIT** request, the MWE for the dump remains queued until canceled or the **HOLD** is released. All **WAIT** requests are purged and must be resubmitted once the **HOLD** is released for the dump to take place.

When the **HOLD** command is issued from a host that is connected to a common dump queue (CDQ), the effect is different than in a non-CDQ environment. The actions taken by this host include:

- A **HOLD DUMP** does not fail **WAIT** type requests if another host in the CDQ is not held. This is because this host's resources are those of the group.
- Requests submitted to the CDQ from this host that are active or queued are allowed to complete as long as work did not get assigned to the host issuing the **HOLD DUMP**.
- A **HOLD DUMP** on a CDQ host may cause the work currently on that host for processing to be failed, based on whether **EOD** or **EOV** is specified on the **HOLD**.
- If the **HOLD** was issued from the master scheduler host, the scheduling of CDQ work is prevented.

Consider using **SETSYS MAXDUMPTASKS(0)** from the master scheduler, instead of this **HOLD**, if you want the master scheduler to schedule CDQ work but not use its resources for CDQ dump processing.

When the **HOLD** command is issued from all hosts connected to a common dump queue, which changes the master scheduler status to **DISABLED**, the actions taken by each host include:

- New WAIT request types from this host are failed locally
- New NOWAIT request types from this host are queued locally as the master scheduler status is DISABLED
- If EOD is specified on the HOLD, all requests are failed. If EOVS is specified, the current volume in a stack is completed and the remaining are failed.
- Queued WAIT request types in the CDQ master scheduler host are failed

HOLD FRBACKUP, DUMP, and AUTODUMP all have the same impact in a common dump queue (CDQ) for selection of a host for dump processing. Any one of these being held prevents a host from being selected for all CDQ dump processing. HOLD of FRBACKUP, DUMP, and/or AUTODUMP also impact the master scheduler's ability to perform scheduling of CDQ work. Instead of the HOLD, consider using the SETSYS MAXDUMPTASKS(0) command from all of the hosts connected to a CDQ. This prevents the selection of a host's resources for CDQ dump processing. It allows the master scheduler to schedule CDQ work but not select its resources for CDQ dump processing. If a HOLD was used, any takeover of the master scheduler responsibilities by another system are impacted by a previously issued HOLD for DUMP. In this case the HOLD DUMP should only be used to prevent the selection of a host's resources, unless the host is defined to the group with a master scheduler candidate status of NO, with SETSYS COMMONQUEUE(DUMP(MSC(NO))).

In addition, when any variety of the HOLD command is issued from a host that is connected to a common dump queue, its host resources are prevented from being selected to process a request by the common queue, even if the requests did not originate from the system that issued the hold command.

Notes:

1. The MASTERSCHEDULER status for the dump group on this host is not changed by this command. It will continue to have a MASTERSCHEDULER status of ENABLED as long as one host in the group is not held. If all of the group hosts have been held, then no host is eligible to do the work and the MASTERSCHEDULER status will be changed to be DISABLED. In this state, the wait-type requests are failed and no-wait type requests are still placed on the common dump queue.
2. You can further refine the capabilities of a common dump queue host, as follows:
 - a. If you wish a dump common dump host to be able to submit command work, but *not* process dump work, issue the SETSYS MAXDUMPTASKS(0) command on that host. This can be either a master scheduler host or a non-master scheduler host. In the case of a master scheduler host, this does *not* impact the master scheduler duties.
 - b. If you wish to stop automatic dump processing in a common dump queue, issue the HOLD DUMP(AUTO) command on the host that originally requested the automatic dump processing.
 - c. If you wish to stop a host from performing master scheduler duties, issue a SETSYS COMMONQUEUE(DUMP(MSC(NO))) on that host. Another master scheduler candidate host, if available, will take on the master scheduler duties.
 - d. If you wish to prevent a host from running automatic dump processing, while still allowing it to submit command dumps and to process command and auto dumps using the common dump queue, simply do *not* provide an AUTODUMPSTART window for that host.

Defaults: If AUTO is not specified, command dump and automatic dump processing are prevented.

ENDOFDATASET: Specifying when aggregate group processing should stop

Explanation: ENDOFDATASET is an optional parameter that you can use with the ABACKUP and ARECOVER parameters to specify when processing stops during aggregate backup or aggregate recovery.

Refer to “ABACKUP: Preventing aggregate backup” on page 897 and “ARECOVER: Preventing aggregate recovery” on page 898 for a description of aggregate processing when using the ENDOFDATASET parameter with the ABACKUP or ARECOVER parameters.

Defaults: None.

ENDOFDATASET and ENDOFVOLUME: Specifying when volume processing should stop

Explanation: **ENDOFDATASET** and **ENDOFVOLUME** are optional parameters specifying when to stop volume processing of one of the following:

- Space management (automatic or command)
- Backup (automatic or command)
- Dump (automatic or command, including fast replication dump)
- Recover (including fast replication recovery from a dump volume)
- Restore

Note: This description does not apply to aggregate group processing or fast replication backup or recovery processing. See “[ENDOFDATASET: Specifying when aggregate group processing should stop](#)” on page 903 for information on holding aggregate group processing. See “[FRBACKUP: Preventing fast replication backup or dump processing](#)” on page 905 and “[FRRECOV: Preventing fast replication recovery processing](#)” on page 905 for information on holding fast replication backup or recovery processing.

ENDOFDATASET specifies that currently processing volume tasks stop after the current data set finishes processing. In the case of DUMP, FRBACKUP DUMP, FRBACKUP DUMPONLY, FRRECOV, or FRRECOV(TAPE), the current dump copies being written by each dump task are discarded; if any dump task is stacking output, any dump copies successfully completed by that task are preserved.

ENDOFVOLUME specifies that volume tasks stop after the current user volume finishes processing. In the case of DUMP, FRBACKUP DUMP, or FRBACKUP DUMPONLY, the current dump copies being written by each dump task are allowed to complete; if any dump task is stacking output, no further dump copies are stacked or processed.

When the level functions are running at the time a HOLD EOV command is issued, the command is interpreted to mean that the particular function runs to completion before the stop occurs. **Example:** If a HOLD EOV is entered while the movement of backup versions is running, the movement of the backup versions is allowed to complete all of its requests but the backup of migrated data sets does not start. In other words, HOLD EOV means stop processing a specific function at the end of a volume, or at the end of the movement of backup versions from a migration volume to a daily backup volume, or at the end of the movement of backup versions from one volume to another volume.

Either parameter prevents processing another user volume.

Defaults: If neither ENDOFDATASET nor ENDOFVOLUME is specified, the default is ENDOFDATASET. If the function being held is DUMP, FRBACKUP DUMP, or FRBACKUP DUMPONLY, the default is ENDOFVOLUME.

Note: If you hold dump processing with the ENDOFDATASET parameter while DFSMSHsm is automatically deleting expired dump copies or deleting excess dump VTOC copy data sets, processing is ended after DFSMSHsm finishes processing the current dump generation.

EVENTDRIVENMIGRATION: Preventing Interval and On-demand Migration

Explanation: **EVENTDRIVENMIGRATION** is an optional parameter specifying that DFSMSHsm stop both interval Migration and On-demand Migration. Other automatic migration functions may still run.

Defaults: None.

Note: When you issue the HOLD command, if Interval or On-demand migration have already been started they will run through to completion, and then will be held the next time they try to run.

Note: If automatic migration (HOLD AUTOMIGRATION) has been held, it will also cause Interval and On-demand migration to be held along with other automatic migration functions.

EXPIREBV: Ending the expire backup function

Explanation: **EXPIREBV** is an optional parameter specifying that DFSMSHsm stop the process of deleting old, unwanted backup versions from the BCDS at the end of the current data set and not allow an EXPIREBV command to be started. If the expire backup function is running at the time the HOLD command is entered, the key of the last data set processed is saved, which allows processing to continue from that point if the RESUME parameter is specified on the next EXPIREBV command processed by this host. WAIT-type command requests fail, and NOWAIT-type command requests remain queued.

Defaults: None.

FRBACKUP: Preventing fast replication backup or dump processing

Explanation: **FRBACKUP** is an optional parameter that you specify to prevent new FRBACKUP, FRBACKUP DUMP, or FRBACKUP DUMPONLY commands from processing. When you issue the HOLD FRBACKUP command, DFSMSHsm completes any FRBACKUP, FRBACKUP DUMP, or FRBACKUP DUMPONLY requests that have already started.

When a HOLD command is submitted that interrupts a dump of a copy pool, if it is a NOWAIT request, the MWE for the dump remains queued until canceled or the HOLD is released. All WAIT requests are purged and must be resubmitted once the HOLD is released for the dump to take place.

HOLD ALL prevents FRBACKUP, both dump of target volumes and backup of DASD, from starting. The auto dump of FRBACKUP target volumes is also held. Currently running target dumps and DASD copy pool backups are allowed to finish.

This command will also have the same effect if the system issuing the command is connected to a dump group in the HSMplex. The respective queue in this case is the dump group common queue.

Defaults: None.

Note: HOLD ALL does not control the hold level of the common recall queue (CRQ) placement and selection functions. To place a hold on CRQ placement and selection, specify HOLD COMMONQUEUE. To place a hold on only one of the two functions (placement or selection), specify the appropriate optional subparameter.

FRRECOV: Preventing fast replication recovery processing

Explanation: **FRRECOV** is an optional parameter that you specify to prevent new FRRECOV commands from processing. When you enter the HOLD FRRECOV command, DFSMSHsm completes any FRRECOV requests that have already started.

The description below applies to preventing fast replication recovery and recover processing when running in a common recover queue (CVQ).

When you issue the HOLD command (ALL, RECOVER, FRRECOV, or FRRECOV(TAPE)) from a host that is connected to a CVQ, the effect is different from in a non-CVQ environment. The actions taken by this host include:

- HOLD of any type (ALL, RECOVER, FRRECOV) will prevent a host being selected by the master scheduler as a processing host, even though the request may be for another type that the HOLD was specified for.
- HOLD of any type (ALL, RECOVER, FRRECOV) on a specific host will not prevent that host from submitting requests, including WAIT type requests, to the CVQ. This is because this host's resources are those of the group. It just won't be selected as a processing host (previous item).
- HOLD by the Master Scheduler host for a specific type will affect the assignment of that type by the Master Scheduler host. A HOLD on both (RECOVER, FRRECOV) on the CVQ MS will prevent it from performing its assignment duties. For example, a HOLD RECOVER on the Master Scheduler host will prevent the assignment of only RECOVER requests but will still assign FRR CP requests volumes to processing hosts. The Query Active command message ARC1560I response will indicate this status.
 - Requests submitted to the CVQ from this host that are active or queued are allowed to complete as long as work did not get assigned to the host issuing the HOLD

- A HOLD on a CVQ host may cause the work currently on that host for processing to be failed, based on whether EOD or EOVS is specified on the HOLD.
- Consider using SETSYS MAXDUMPRECOVERTASKS(0) from the master scheduler, instead of A HOLD, if you want the master scheduler to schedule CVQ work but not use its resources for CVQ volume restore processing.
- When the HOLD command (ALL, RECOVER, FRRECOV, or FRRECOV(TAPE)) is issued from all hosts connected to a common recover queue, which changes the master scheduler status to DISABLED, the actions taken by each host include:
 - New WAIT request types from this host are failed locally
 - New NOWAIT request types from this host are queued locally as the master scheduler status is DISABLED
 - If EOD is specified on the HOLD, all requests are failed. If EOVS is specified, the current volume in a stack is completed and the remaining are failed.
 - Queued WAIT request types in the CDQ master scheduler host are failed

When you select a host for volume restore processing, the HOLD ALL, RECOVER (any variety), and FRRECOV(TAPE) commands prevent a host from being selected for CVQ volume restore processing

These holds also impact the ability of the master scheduler to perform scheduling of CVQ work for the specific hold type. Instead of using the HOLD command, consider using the SETSYS MAXDUMPRECOVERTASKS(0) command from all of the hosts connected to a CVQ.

Using the SETSYS MAXDUMPRECOVERTASKS(0) command prevents the selection of host resources for CVQ volume restore processing, and allows volume restore requests to be submitted. It also allows the master scheduler to schedule CVQ work without selecting resources for processing. If a HOLD was used, any transfer of the master scheduler responsibilities to another system is impacted by a previously issued HOLD for RECOVER or FRRECOV(TAPE). In this case, the HOLD can be used only to select resources from a host, unless the host is defined to the group with a master scheduler candidate status of NO, with SETSYS COMMONQUEUE(DUMP(MSC(NO))).

In addition, when you issue any of the HOLD commands from a host that is connected to a CVQ, it prevents the selection of host resources to process a request, even when the requests did not originate from the system that issued the HOLD command.

Note:

The MASTERSCHEDULER status for the common recover group on this host remains ENABLED until one host in the group is not held. If all of the group hosts have been held, then no host is eligible to do the work, and the MASTERSCHEDULER status changes to DISABLED. In this state, the wait-type requests are failed and no-wait type requests are placed on the common recover queue.

You can further refine the capabilities of a common recover queue host, as follows:

- When you want a common recover queue host to be able to submit command work, but not process the work, issue the SETSYS MAXDUMPRECOVERTASKS(0) command on that host. This can be either a master scheduler host or a non-master scheduler host. When you issue the SETSYS MAXDUMPRECOVERTASKS(0) command on master scheduler host, it does not impact the master scheduler responsibilities.

Subparameter	Explanation
TAPE	FRRECOV(TAPE) specifies that recovery processing of FRBACKUP generated dump volumes is prevented. No more recoveries from FRBACKUP dump tasks are started. You can still recover DASD backups.
DATASET	FRRECOV(DATASET) is an optional parameter for HOLD FRRECOV. Data set recoveries are held when HOLD ALL, HOLD FRRECOV or HOLD FRRECOV(DATASET) are specified. The existing TAPE keyword will hold data set recoveries from tape.

Defaults: None.

LIST: Ending the list function

Explanation: **LIST** is an optional parameter specifying that DFSMSHsm stop the list function after it finishes processing the control data set record of the current request. WAIT-type command requests fail, and NOWAIT-type command requests remain queued.

Defaults: None.

Note: When you issue the HOLD command, you lose any part of the LIST command that did not finish running.

LOG: Preventing logging

Explanation: **LOG** is an optional parameter specifying that DFSMSHsm stop recording in the DFSMSHsm log. No records are written in the DFSMSHsm log for those DFSMSHsm functions that continue to run. DFSMSHsm continues recording information in the activity logs.

Defaults: None.

MIGRATION: Preventing space management

Explanation: **MIGRATION** is an optional parameter specifying whether DFSMSHsm should prevent all space management processing or prevent only automatic space management processing. The MIGRATION parameter, without any subparameters, specifies that all space management functions are prevented. No more space management tasks are started.

Subparameter	Explanation
AUTO	MIGRATION(AUTO) specifies that automatic volume and automatic secondary space management processing are prevented. No more automatic volume and automatic secondary space management processing tasks are started. You can still perform command space management.

The point at which DFSMSHsm stops space management processing is controlled by the ENDOFDATASET and ENDOFVOLUME parameters. WAIT-type command requests are failed, and NOWAIT command requests remain queued.

Defaults: If AUTO is not specified, automatic volume space management, automatic secondary space management, and command space management are prevented.

Note: HOLD MIGRATION(AUTO) will not HOLD AUTOEXPIRATION. You must use HOLD AUTOEXPIRATION if you want this function held.

RECALL: Preventing recall and deletion

Explanation: **RECALL** is an optional parameter specifying which recalls (automatic or command) and deletions of a migrated data set are prevented.

The RECALL parameter, without any subparameters, specifies that all recall and data set deletion tasks from all volumes are prevented. No more recall or data set deletion tasks are started.

Subparameter	Explanation
TAPE	<p>RECALL(TAPE) specifies that recall requests from all tape volumes do not process. If you specify RECALL(TAPE) for a host that is participating in a common recall queue, the host selects only recall requests that do not require tape.</p> <p>RECALL(TAPE(TSO)) specifies that TSO recall requests from migration level 2 tapes are prevented. Recall requests from tape submitted from a batch job and all nontape volumes are still processed.</p>

Defaults: None.

Note:

1. All nonWAIT requests remain queued when the appropriate function is held.
2. If TSO tape recalls are held, a WAIT-type recall from tape that was submitted through TSO is changed to a non-WAIT-type request. The user is told that the request remains on the queue to be processed when the function is released.
3. All WAIT recall or deletion requests from batch or from the storage administrator are failed when RECALL is held.
4. All WAIT recall requests from tape, from batch, or from the storage administrator are failed when tape recall is held.

RECOVER: Preventing recovery and restore

Explanation: **RECOVER** is an optional parameter specifying whether DFSMSHsm prevents all recovery and restore processing or prevents only tape data set recovery processing. The RECOVER parameter, without any subparameters, specifies that all recover and restore processing tasks are prevented. No additional recovery and restore processing tasks are started. The RECOVER parameter prevents any request for backup data sets on a mounted tape volume.

When you issue any HOLD (ALL, RECOVER, FRRECOV, or FRRECOV(TAPE)) command from a host that is connected to a common recover queue (CVQ), the HOLD command has a different effect than when it is issued from a non-CVQ environment. See [“FRRECOV: Preventing fast replication recovery processing”](#) on page 905 for details.

Subparameter	Explanation
TAPEDATASET	RECOVER(TAPEDATASET) specifies that tape recover and restore processing is prevented. No additional tape data set processing tasks are started. You can still perform recover processing tasks from DASD.

The point at which DFSMSHsm stops recovery processing is controlled by the ENDOFDATASET and END OFVOLUME parameters. WAIT-type command requests fail, and NOWAIT-type command requests remain queued.

Defaults: If TAPEDATASET is not specified, command recovery for both data sets and volumes is prevented.

Note: If a HOLD RECOVER(TAPEDATASET) has been issued, any request for backup data sets on a mounted backup tape volume is processed, and any requests for the new tape volume mounts are not processed.

RECYCLE: Ending the recycle function

Explanation: **RECYCLE** is an optional parameter specifying that the recycle function ends after DFSMSHsm processes the current backup version or migration copy on each volume being recycled. WAIT-type command requests fail, and NOWAIT-type command requests remain queued.

Defaults: None.

Note: When you issue the HOLD command, you lose any part of the RECYCLE command that has not finished running.

REPORT: Ending the report function

Explanation: **REPORT** is an optional parameter specifying that DFSMSHsm stop the report function after it has finished processing the control data set record for the current request. WAIT-type command requests fail, and NOWAIT-type command requests remain queued. When you issue the HOLD command, you lose any part of the REPORT command that has not finished running.

Defaults: None.

TAPECOPY: Preventing DFSMSHsm from copying a tape

Explanation: **TAPECOPY** is an optional parameter specifying that DFSMSHsm hold all tape copy processes. WAIT-type command requests fail, and NOWAIT-type command requests remain queued.

If DFSMSHsm was processing an explicit volume list when the HOLD command was issued, the remaining volumes in the list may not process when the TAPECOPY command is released.

Defaults: None.

TAPEREPL: Preventing DFSMSHsm from replacing an original tape volume with an alternate

Explanation: **TAPEREPL** is an optional parameter specifying that DFSMSHsm hold all tape replacement processes. WAIT-type command requests fail, and NOWAIT-type command requests remain queued.

Defaults: None.

Examples of how to code the HOLD command

The following examples present different ways to code the HOLD command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Preventing space management and recall

In this example, DFSMSHsm prevents processing of automatic space management, space management by command, and recall and deletion of migrated data sets.

```
HOLD MIGRATION RECALL
```

Preventing all DFSMSHsm functions and logging

In this example, DFSMSHsm prevents processing of all DFSMSHsm functions, including logging.

```
HOLD ALL LOG
```

Ending the report function

In this example, DFSMSHsm stops the report function at the end of the current unit of work. No more REPORT commands are started.

```
HOLD REPORT
```


Ending all recalls from tape

In this example, DFSMSHsm stops all recalls from tape.

```
HOLD RECALL(TAPE)
```

Ending all recalls from tape submitted by an interactive TSO user

In this example, DFSMSHsm stops all recalls from tape submitted by an interactive TSO user. DFSMSHsm allows recalls from tape that are submitted from a batch job to process.

```
HOLD RECALL(TAPE(TSO))
```

Preventing placement and selection of recall requests in the CRQ

In this example, DFSMSHsm prevents the host from placing recall requests in the common recall queue (CRQ) and selecting recall requests from the CRQ.

```
HOLD COMMONQUEUE(RECALL)
```

Preventing placement of recall requests in the CRQ

In this example, DFSMSHsm prevents the host from placing recall requests in the CRQ, but the host selects recall requests for processing from the CRQ.

```
HOLD COMMONQUEUE(RECALL(PLACEMENT))
```

Preventing selection of recall requests from the CRQ

In this example, DFSMSHsm prevents the host from selecting recall requests for processing from the CRQ, but the host places recall requests in the CRQ.

```
HOLD COMMONQUEUE(RECALL(SELECTION))
```

Ending all tape copying

In this example, DFSMSHsm stops all tape copying.

```
HOLD TAPECOPY
```

Ending all tape replacement

In this example, DFSMSHsm stops all tape replacement.

```
HOLD TAPEREPL
```

Preventing backup function from running

In this example, DFSMSHsm prevents the backup function from running.

```
HOLD BACKUP
```


Preventing the fast replication backup function from running

In this example, DFSMSHsm prevents new FRBACKUP commands from processing. When you issue this command, DFSMSHsm completes any FRBACKUP requests that have already started.

```
HOLD FRBACKUP
```

Preventing the fast replication recovery function from running

In this example, DFSMSHsm prevents new FRRECOV commands from processing. When you issue this command, DFSMSHsm completes any FRRECOV requests that have already started.

```
HOLD FRRECOV
```

Preventing the automatic volume and automatic secondary space management functions from running

In this example, DFSMSHsm prevents the automatic volume and automatic secondary space management functions from running. Command migration can still run.

```
HOLD MIGRATION(AUTO)
```

Preventing processing of a specific function at the end of the data set

In this example, DFSMSHsm stops backup processing at the end of the data set currently being processed.

```
HOLD BACKUP ENDOFDATASET
```

Preventing dump processing

In this example, DFSMSHsm prevents the dump function from running. Dumps currently in progress complete.

```
HOLD DUMP
```

Note: This is equivalent to HOLD DUMP ENDOFVOLUME.

Ending dump processing

In this example, DFSMSHsm stops all dumps the next time DFSMSdss reads or writes a record. Dumps in progress are stopped and invalidated.

```
HOLD DUMP ENDOFDATASET
```

Preventing automatic dump processing

In this example, DFSMSHsm prevents the automatic dump function from running. Any automatic dumps currently in progress complete, but no new automatic dumps start. Command dumps can still run.

```
HOLD DUMP(AUTO)
```

Preventing the expire backup versions function from running

In this example, DFSMSHsm stops deleting backup versions at the end of the current data set.


```
HOLD EXPIREBV
```

Preventing aggregate backup

In this example, DFSMSHsm stops backing up aggregated data sets after completion of any currently running aggregate backup.

```
HOLD ABACKUP(PAY1) EOD
```

Preventing aggregate recovery

In this example, DFSMSHsm stops recovering aggregated data sets after completion of any currently running aggregate recovery.

```
HOLD ARECOVER(PAY1.C.C01V0001)
```

Preventing automatic backup and command data set backups to DASD

In this example, DFSMSHsm holds automatic backups and command data set backups directed to ML1 DASD. Command data set backups to tape continue processing.

```
HOLD BACKUP(AUTO DSCOMMAND(DASD))
```

Preventing automatic backup and command data set backups to tape

In this example, DFSMSHsm holds automatic backups and command data set backups directed to tape. Command data set backups to ML1 DASD continue processing.

```
HOLD BACKUP(AUTO DSCOMMAND(TAPE))
```

Interrupting command data set backup to change output tapes

In this example, you have tapes that are scheduled to be taken offsite for disaster backup and you need to switch the output tapes used by command data set backup.

```
HOLD BACKUP(DSCOMMAND(SWITCHTAPES))
```


Chapter 51. HSEND CMD command: Issuing DFSMSHsm authorized-user commands from a TSO terminal

In a RACF FACILITY class environment, permission to issue the HSEND CMD is not checked by DFSMSHsm. Instead, permission to issue the command submitted by HSEND CMD is checked by DFSMSHsm using RACF FACILITY class checking.

In an AUTH environment, if you are an authorized user, you can use the HSEND CMD command to issue any DFSMSHsm operator, storage administrator, or system programmer commands from a TSO terminal, instead of from the system console.

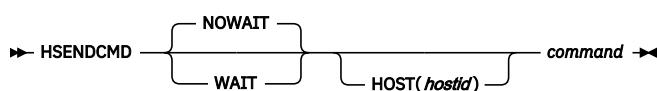
In this AUTH environment, DFSMSHsm allows an unauthorized user to use the HSEND CMD to issue the ALTERDS, BDELETE, and the QUERY commands. The subcommands ALTERDS and BDELETE are processed in the same manner as the HALTERDS and HBDELETE.

In a RACF FACILITY class environment, an unauthorized user can only use the HQUERY, HCANCEL, HBDELETE, and HALTERDS commands. Users no longer have access to any storage administrator commands unless the security administrator has given them access to those command profiles.

Related Reading:

- For a list of the operator commands, refer to [“Storage administrator commands” on page 682](#)
- For a list of the storage administrator commands, refer to [“Operator commands” on page 681](#)
- For a list of the system programmer commands, refer to [“System programmer commands” on page 683](#)
- To authorize users under different environments, refer to [“Submitting DFSMSHsm commands” on page 687](#).

Syntax of the HSEND CMD command



Required parameters of the HSEND CMD command

This topic describes the required parameter of the HSEND CMD command.

command: Specifying DFSMSHsm authorized command

Explanation

command is a required positional parameter specifying the authorized command from a TSO terminal. For *command*, substitute the DFSMSHsm command you want and its associated parameters. You cannot use more than 1024 bytes. The command and its associated parameters are treated as a group of parameters.

Optional parameters of the HSEND CMD command

This topic describes the optional parameters of the HSEND CMD command.

WAIT and NOWAIT: Specifying whether to wait for command completion

Explanation: WAIT and NOWAIT are mutually exclusive, optional parameters to specify whether you want to wait for the HSEND CMD requested function command to complete before control is returned to the user. If WAIT or NOWAIT is used, it must immediately follow HSEND CMD and precede command.

WAIT

Wait for the HSEND CMD requested function command to complete before control is returned to the user. The following DFSMSHsm commands (when used with the HSEND CMD WAIT) cannot ensure that the action will have completed at the time control is returned to the requester, because the processing of the command merely sets a flag or builds a request that is subsequently processed by another task.

- LOG
- QUERY SPACE
- STOP
- SWAPLOG

When you use the WAIT parameter with a command that processes a single item (such as one data set) or a volume as a single unit (such as DUMP), an ARC1000I message is issued if the process is successful, or an ARC1001I message is issued if the process is not successful. The following is a list of these commands:

- ABACKUP
- ARECOVER
- BACKDS
- BACKVOL with DUMP parameter
- DELETE of a migrated data set
- MIGRATE of a data set
- RECALL
- RECOVER of a data set
- RECOVER of a volume from a DFSMSdss physical full-volume copy

Other commands, when issued with the WAIT parameter, do not issue a successful termination message or set an HSEND CMD failing return code; however, they do issue explicit error messages for failing conditions.

If you press the TSO attention key when the WAIT parameter has been specified on a command request, and you confirm your intention to "attention out" of the request, an ARC1800I message is issued. This action makes your terminal available for use but does not cause DFSMSHsm to discontinue processing the request. If the request is for recalling or recovering a data set, a volume mount request (ARC0612I) message can be issued.

NOWAIT

Do not want to wait for the HSEND CMD requested function command to complete before control is returned to the user. The completion message does not occur until the requested function is complete. If the NOWAIT option is being used with the ABACKUP, ARECOVER, BACKDS, BACKVOL, DELETE, MIGRATE, RECALL, or RECOVER commands, an ARC1007I message is issued after the command has been communicated to DFSMSHsm. After DFSMSHsm has processed the request, a completion message (ARC1000I or ARC1001I) is written, indicating respectively success or failure of the request. If the request is for recalling or recovering a data set, a volume mount request (ARC0612I) message is issued.

Defaults: The default is NOWAIT.

HOST: Specifying the DFSMSHsm host that is to receive the command

Explanation: HOST(*hostid*) is an optional parameter to specify the host where the command needs to be issued. The parameter variable is explained:

hostid

The *hostid* is a one-character host identifier that is used to distinguish which active DFSMSHsm (main or auxiliary) host is to receive the command. The specified host ID must match one of the currently active DFSMSHsm host IDs. For requirements of the *hostid* value, see the topic about startup procedure keywords in *z/OS DFSMSHsm Implementation and Customization Guide*.

Defaults If the host parameter is not specified, the command is directed to the main host.

Usage notes:

1. The following are HSEND CMD return codes and their meanings:

Return code**Meaning****0**

With the WAIT option, communication with DFSMSHsm is successful and the function is successful.

With the NOWAIT option, communication with DFSMSHsm is successful. It is not known if the function is or is not successful.

4

With the WAIT option, the function has encountered an error.

8

With the WAIT or NOWAIT option, communication with DFSMSHsm has failed.

2. The HSEND CMD command name must precede each authorized command you issue from a TSO terminal.
3. An HSEND CMD return code of 4 indicates that a functional error occurred during processing when one of the previously identified commands was being performed. Specific errors can be determined by the ARC1001I message and the DFSMSHsm messages accompanying it. However, if the command was BACKVOL CDS, then specific errors can be determined by the ARC0741I, ARC0744E, and ARC0747E messages. Return code 4 means a functional error only when one of these specifically identified commands is issued.

Note: A return code of 4 will be returned from any request when DFSMSHsm cannot recognize the command or all of its parameters.
4. These HSEND CMD return codes are set in Register 15. They can be tested in a CLIST using as variables &LASTCC or &MAXCC; They can also be tested with the COND parameter within JCL when the commands are issued in the TSO batch environment.
5. If you issue multiple HSEND CMD commands at the same time, some specifying the WAIT parameter and others the NOWAIT parameter, all of the commands with the WAIT parameter are processed first, regardless of the order in which the commands were received.
6. The WAIT and NOWAIT parameters of the HSEND CMD command have implications for HOLD command processing that are function specific. Refer to the specific processing function for an explanation.
7. When a command is issued to an inactive host with NOWAIT specified, then the command is queued and ARC1004I message is issued. The command is executed when the host is started. When a command is issued to an inactive host with WAIT specified, then the request is rejected and ARC1005I is issued.

Examples of how to code the HSEND CMD command

The following examples present different ways to code the HSEND CMD command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Sending an authorized command with WAIT from a TSO terminal

In this example, the ADDVOL command is issued from a TSO terminal. Control returns to the user when the command is completed. The syntax of the ADDVOL command does not change when it is used with the HSEND CMD command.

```
HSEND CMD WAIT ADDVOL VOL004 UNIT(3390) +  
PRIMARY(NOAUTOMIGRATION AUTOBACKUP NOAUTORECALL)
```

Sending an authorized command from a TSO terminal

In this example, the ADDVOL command is issued from a TSO terminal. Control is returned to the user before the command completes. The syntax of the ADDVOL command does not change when it is used with the HSEND CMD command.

```
HSEND CMD ADDVOL VOL004 UNIT(3390) +  
PRIMARY(NOAUTOMIGRATION AUTOBACKUP NOAUTORECALL)
```

Sending an authorized command with HOST from TSO terminal

In this example, ADDVOL command is issued to host with *hostid* 2. Control returns to the user before the command completes. The syntax of the ADDVOL command does not change when it is used with the HSEND CMD command.

```
HSEND CMD HOST(2) ADDVOL VOL004 UNIT(3390) +  
PRIMARY(NOAUTOMIGRATION AUTOBACKUP NOAUTORECALL)
```

Chapter 52. HSENDU command: Issuing DFSMSHsm authorized user commands to a Filemode DFSMSHsm host

The HSENDU command is an alias to the HSENCMD command. All parameters valid on the HSENCMD command are acceptable on the HSENDU command. When this command is used, the request is directed to the DFSMSHsm Filemode host. If only a Full Support DFSMSHsm host exists, the request is directed to the Full Support DFSMSHsm host.

Chapter 53. LIST command: Listing information from the MCDS, BCDS, and OCDS

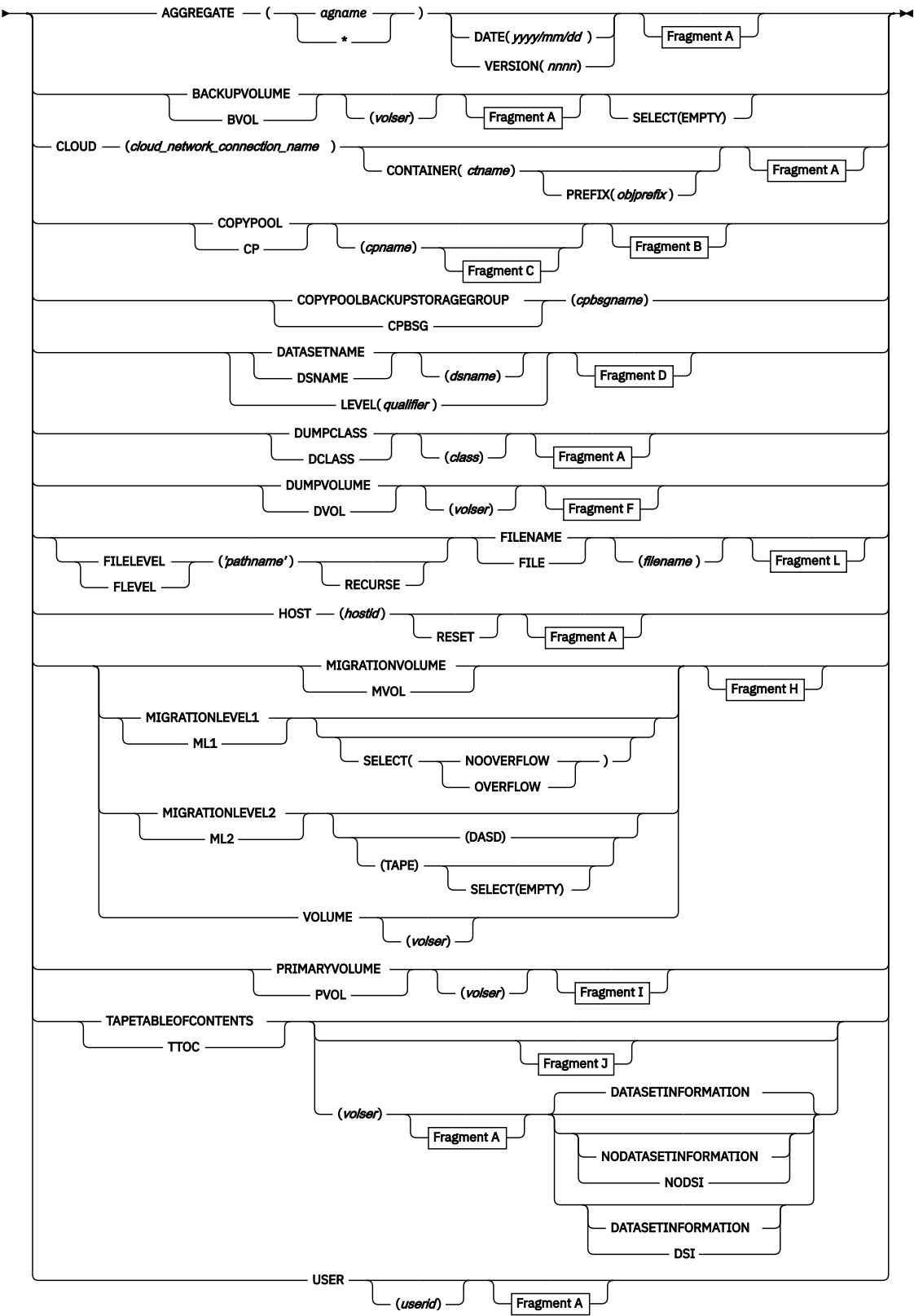
The LIST command lists selected data set, aggregate, user, DFSMSHsm host serialization, and volume information from the MCDS, BCDS, and OCDS. You can list the following categories of information:

Category of Information	See
Aggregate backup and recovery activity information	“AGGREGATE: Listing the aggregate backup and recovery activity information” on page 925
Backup volume information	“BACKUPVOLUME: Listing backup volume entries” on page 925
Cloud container and object information	Cloud storage
Copy pool information	“COPYPOOL: Listing information for each version of a copy pool” on page 926
Copy pool backup storage group information	“COPYPOOLBACKUPSTORAGEGROUP: Listing copy pool information for each volume in the specified copy pool backup storage group” on page 926
Data set information	“DATASETNAME and LEVEL: Listing data set entries” on page 926
Dump class information	“DUMPCCLASS: Listing dump class profiles” on page 927
Dump volume information	“DUMPVOLUME: Listing information about a single dump volume or a set of dump volumes” on page 927
Host information	“HOST: Listing all control data sets serialized by the DFSMSHsm host” on page 928
Migration information	“MIGRATIONVOLUME, MIGRATIONLEVEL1, MIGRATIONLEVEL2, and VOLUME: Listing migration volumes” on page 929
Primary volume information	“PRIMARYVOLUME: Listing primary volumes” on page 929
Tape volume information	“TAPETABLEOFCONTENTS: Listing tape volume information from the OCDS” on page 930
UNIX file information	“FILENAME and FILELEVEL: Listing UNIX file entries” on page 928
User authorization information	“USER: Listing user entries” on page 931

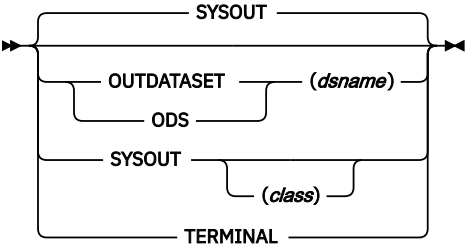
For examples of LIST command output, see [Chapter 76, “Using the LIST command,” on page 1341](#).

Syntax of the LIST command

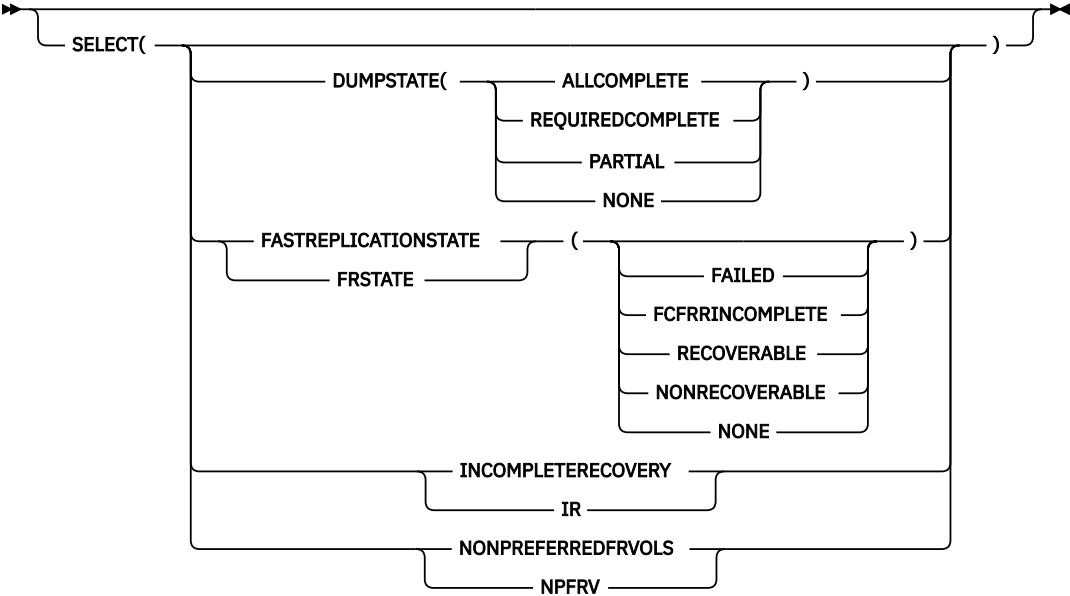
➤ LIST ➔



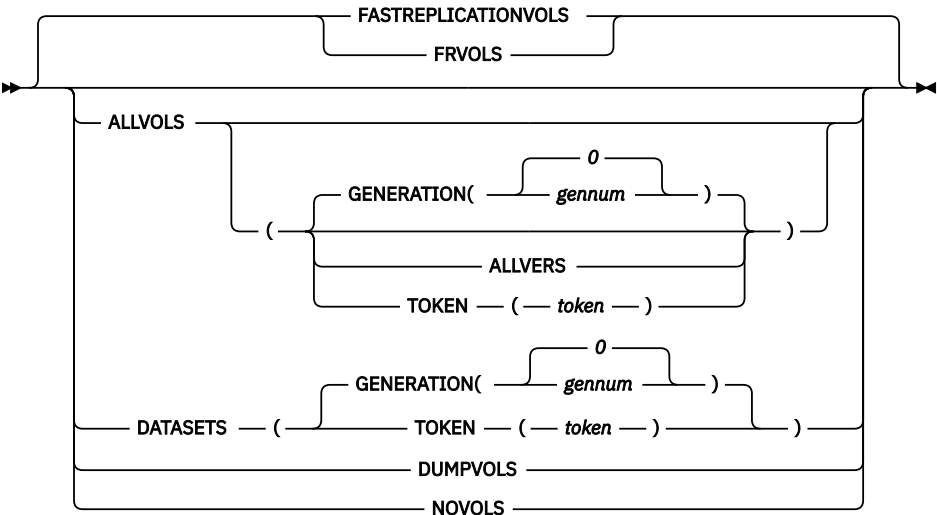
A: LIST optional parameters



B: COPYPOOL optional parameters

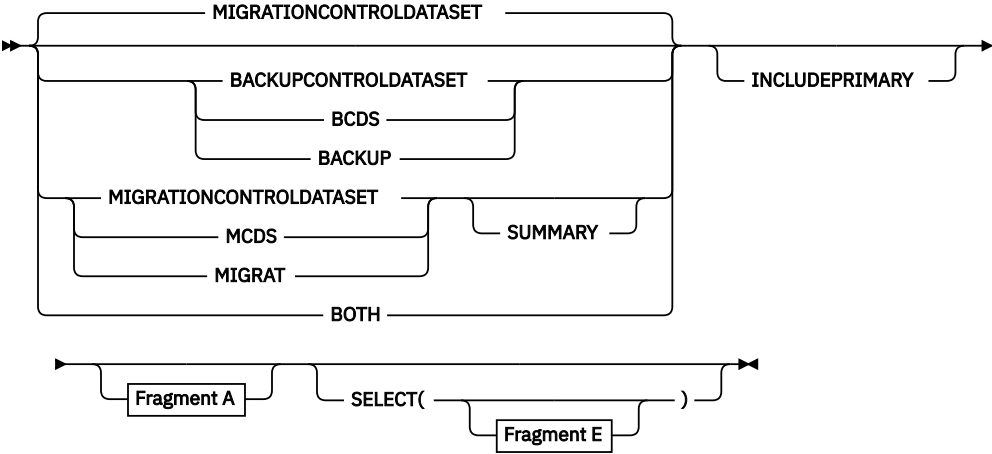


C: COPYPOOL(cpname) optional parameters

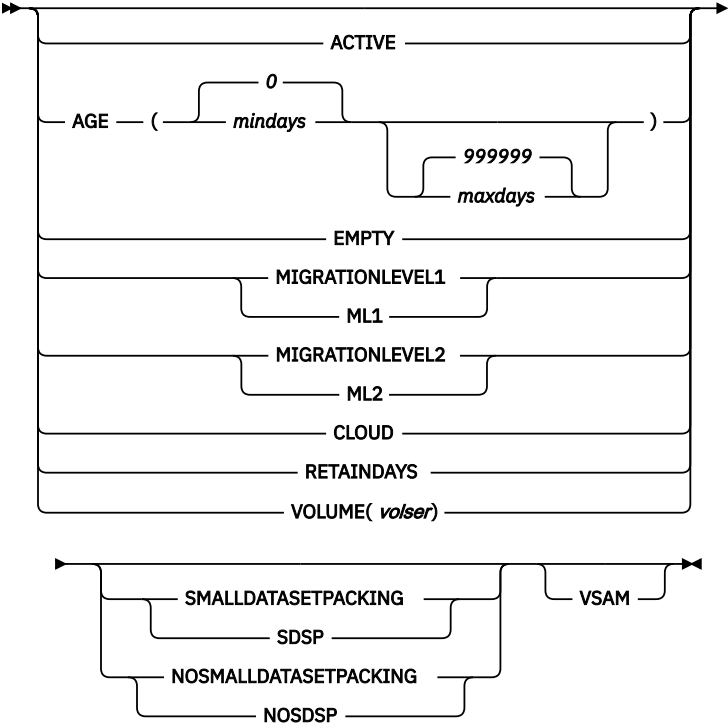


D: DATASETNAME optional parameters

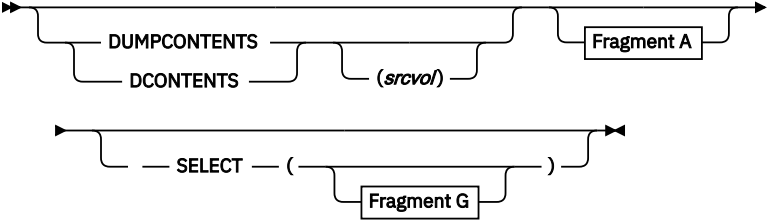
LIST command for DFSMSShsm



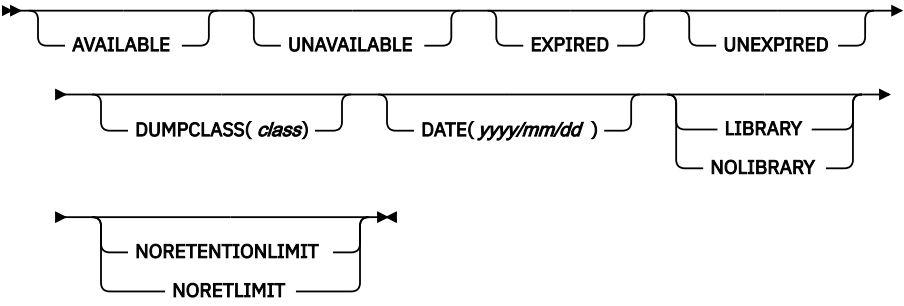
E: DATASETNAME SELECT optional parameters



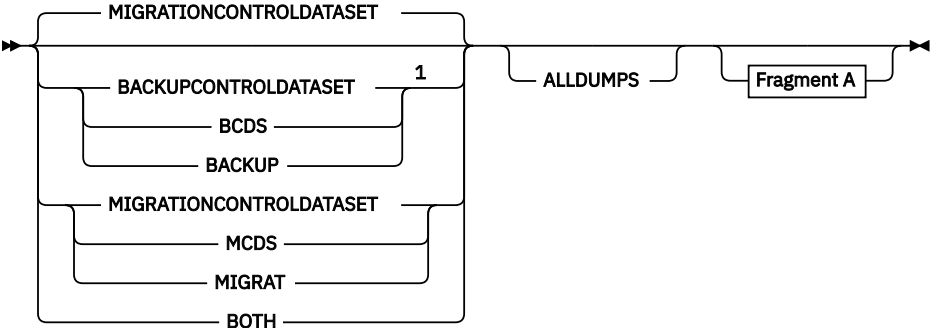
F: DUMPVOLUME optional parameters



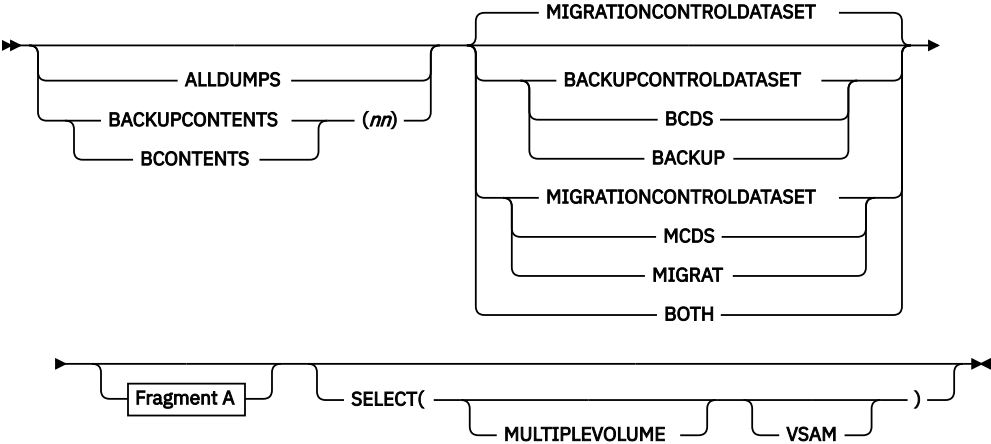
G: DUMPVOLUME SELECT optional parameters



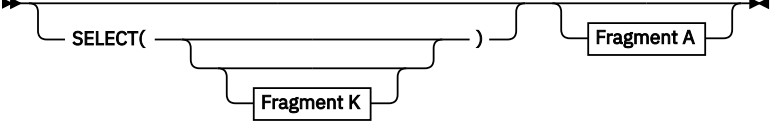
H: MVOL, ML1, ML2, and VOLUME optional parameters



I: PRIMARYVOLUME optional parameters

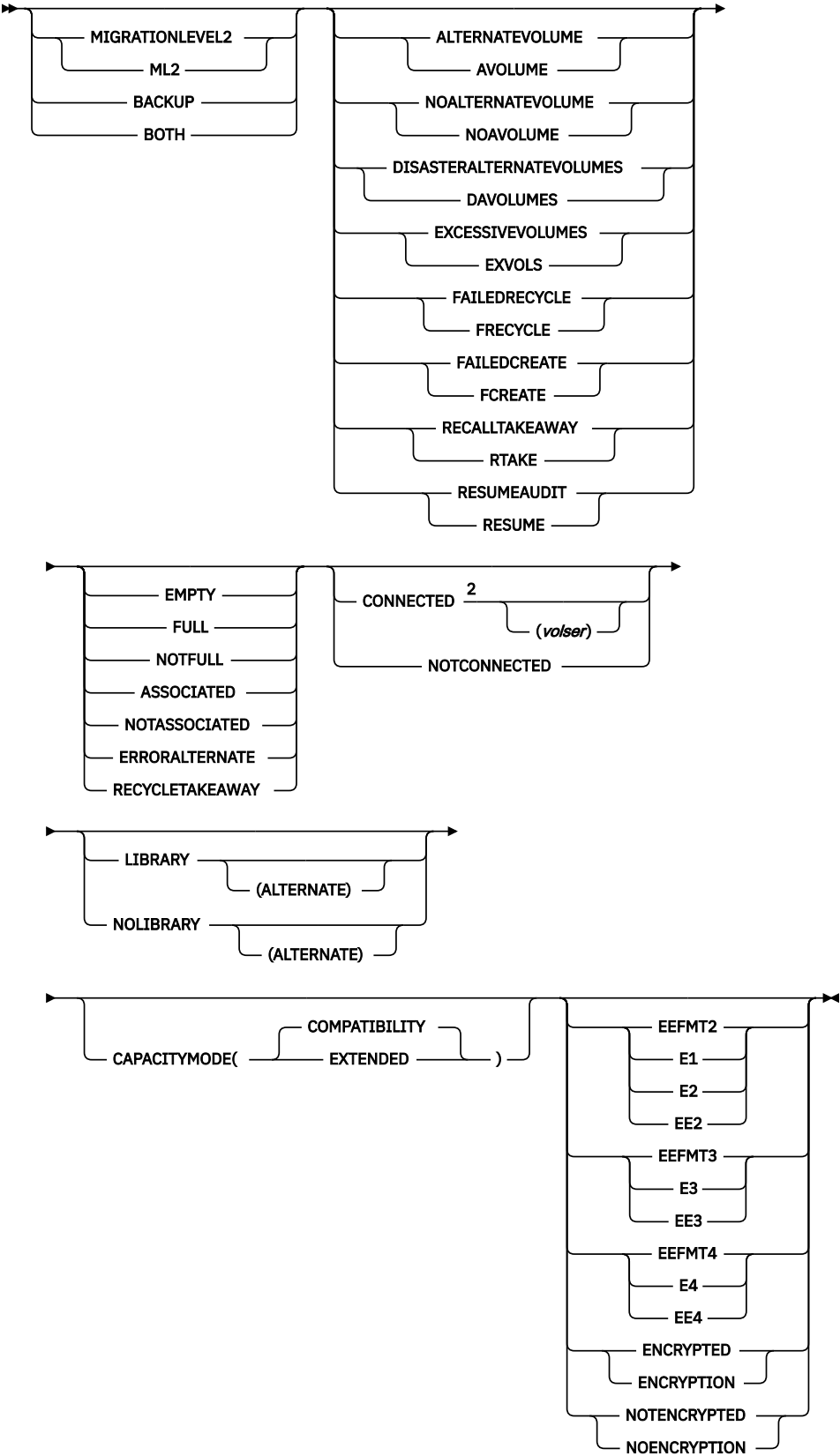


J: TTOC optional parameters

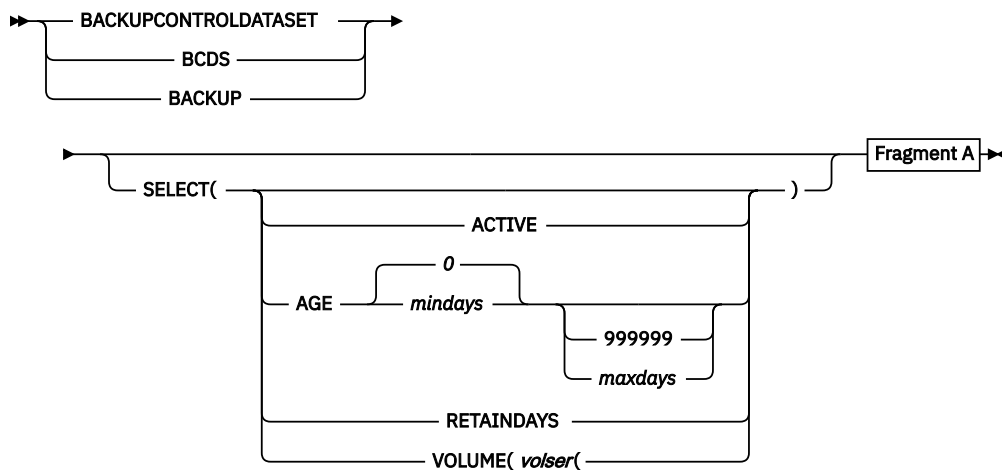


K: TTOC SELECT optional parameters

LIST command for DFSMSHsm



L: FILENAME and FILELEVEL optional parameters



Notes:

- ¹ The BACKUPCONTROLDATASET or BCDS parameter and the BACKUP parameter are not valid with the LIST ML2(TAPE) SELECT(EMPTY) command.
- ² The SELECT(CONNECTED) subparameter of the LIST TTOC command cannot be specified with the SELECT(ML2), SELECT(BACKUP), or SELECT(BOTH) subparameters.

Required parameters of the LIST command

This topic describes the required parameters of the LIST command.

AGGREGATE: Listing the aggregate backup and recovery activity information

Explanation: AGGREGATE is a required parameter you specify to request a list of information for selected backed up and recovered aggregates. This information is contained in the BCDS. The values specified when using the AGGREGATE parameter are explained:

agname

specifies the name of the specific aggregate group whose records are listed.

*

specifies that all records for all aggregate groups are listed.

Defaults: None.

BACKUPVOLUME: Listing backup volume entries

Explanation: BACKUPVOLUME is a required parameter you specify to request a list of selected information from backup volume entries contained in the BCDS.

BACKUPVOLUME specifies that you want a list of all backup volumes.

For *volser*, substitute the serial number of the specific backup volume you want listed.

Defaults: If you specify BACKUPVOLUME without (*volser*), DFSMSHsm lists all backup volume entries contained in the BCDS.

Note:

1. If you specify this parameter, you get a list, but you do not get any additional information about the data sets on the backup volume.

CLOUD: Listing information from cloud storage

Explanation: CLOUD is a required parameter that you specify to retrieve information about containers, and objects in cloud storage. Only containers created using the current HSMplex name will be surfaced in the output.

CLOUD(*cloud_network_connection_name*)

specifies the name of an SMS construct, or a CDA provider file that describes cloud storage to list the DFSMSHsm containers currently existing in the specified cloud.

CONTAINER(*ctname*)

specifies a particular container name within the cloud storage to list the pseudo-directories or prefixes that are found within the requested container. A pseudo-directory is the first part of the object name ending with a forward-slash character (/).

PREFIX(*objprefix*)

specifies to list all of the objects found under a specific prefix or pseudo-directory. It is a maximum 44-byte prefix that can match to specific objects contained within the specified container. Fragment A indicates that output can be written to SYSOUT, OUTDATASET, and TERMINAL.

COPYPOOL: Listing information for each version of a copy pool

Explanation: COPYPOOL is a mutually exclusive required parameter in which you can specify a single copy pool name (*cpname*) or no copy pool name, for a list containing all copy pools.

COPYPOOL

Specify COPYPOOL to request a list of information for each version of each copy pool that has at least one fast replication backup or dump version.

COPYPOOL(*cpname*)

Specify COPYPOOL(*cpname*) to request detailed information for each version of the specified copy pool. This information includes source volume to target volume pairings, DASD versions, dump versions, and catalog information.

Defaults: None.

COPYPOOLBACKUPSTORAGEGROUP: Listing copy pool information for each volume in the specified copy pool backup storage group

Explanation: COPYPOOLBACKUPSTORAGEGROUP is a mutually exclusive required parameter that you specify to request copy pool information for each volume in the specified copy pool backup storage group.

Specify COPYPOOLBACKUPSTORAGEGROUP(*cpbsgname*) to request a list of volumes in the copy pool backup storage group that DFSMSHsm has assigned as a target volume. The corresponding source volumes are also listed.

Defaults: None.

DATASETNAME and LEVEL: Listing data set entries

Explanation: DATASETNAME and LEVEL are mutually exclusive, required parameters that you specify to request a list of data set entries.

DATASETNAME(*dsname*)

is specified to request a data set entry with a specific data set name. For *dsname*, substitute the fully qualified name of the particular data set you want listed. You can specify a data set name of up to 44 characters.

LEVEL(*qualifier*)

is specified to request a list of all data set entries that have the same set of initial characters of the data set name. For *qualifier*, substitute the set of initial characters of the data set name for the data sets you want listed. The set of initial characters can contain imbedded periods. The *qualifier* can end with a period if LEVEL is the first keyword on the command. You can specify a *qualifier* of up to 44 characters.

Defaults: If you specify DATASETNAME without *dsname*, DFSMSHsm lists all data set entries.

Note:

1. If you used the SETMIG command to prevent a data set from migrating, DFSMSHsm does not list the data set.
2. DFSMSHsm does not handle partitioned data set members individually. If you specify a member name of a partitioned data set, DFSMSHsm lists the entire partitioned data set.
3. You can specify DATASETNAME or LEVEL with a LIST COMMAND, but not both. If you specify both parameters, the last parameter specified overrides the previous one.

DUMPCLASS: Listing dump class profiles

Explanation: DUMPCLASS is a mutually exclusive required parameter you specify to request a list of one or more dump-class profiles. For *class*, substitute from 1 to 8 alphanumeric characters for the class to be listed.

The following information is listed for each dump class:

- Class name
- Class disposition
- Retention period
- Required (Y/N)
- Available (Y/N)
- Frequency and/or day
- Automatic reuse option (Y/N)
- Data set restore option (Y/N)
- Change bit reset option (Y/N)
- Unit type
- Class disable (Y/N)
- Maximum stack value
- Minimum stack value
- FRRECOV(AVAILABLEFORMOUNT(Y|N))
- FRDUMP(REQUIRED|OPTIONAL)
- HWCOMPRESS(Y|N)
- Encryption values

Defaults: If you specify DUMPCLASS without *class*, profiles for all dump classes are listed.

DUMPVOLUME: Listing information about a single dump volume or a set of dump volumes

Explanation: DUMPVOLUME is a required parameter you specify to request a list of information about a single dump volume or a set of dump volumes. For *volser*, substitute any *volser* from a set of volumes that collectively represents a dump of a user volume. The following information is listed first about the dump volume:

- Dump volume serial number
- Copy pool (Y|N)
- Volume status
- Dump volume unit type
- Dump class
- Dump expiration date
- IDRC status
- Physical percent full

- Library status
- HWCOMPRESS(Y|N)
- Encryption value
- Tape hardware encryption

The following information is then listed about each valid dump copy that is stacked on the dump volume:

- File sequence number
- Source volume serial number
- SMS status
- Dump creation date and time
- Set of dump volume serial numbers

See [Figure 136 on page 1366](#) for an example of the listed information.

Defaults: None.

Note:

1. If the *volser* specified is not part of a valid dump copy, only the dump class, volume serial, and volume unit type of the specified *volser* is listed.
2. If host based encryption is specified and the SMS data class is set up for tape hardware encryption, the host based encryption is ignored in favor of doing tape hardware encryption. If more than one DUMPCLASS is specified (creating more than one dump copy) and they are set up for host based encryption, but the associated SMS data classes do not agree in their tape hardware encryption settings, the dump function fails.
3. When %CLOUD is specified as the DUMPVOLUME *volser*, the DUMPCONTENTS(*srcvol*) parameter is mandatory for LIST DUMPVOLUME(%CLOUD). SELECT(DDMMYY(*yyyy/mm/dd*)) is optional.

FILENAME and FILELEVEL: Listing UNIX file entries

Explanation: FILENAME and FILELEVEL are mutually exclusive, required parameters that you specify to request a list of UNIX file entries.

FILENAME('filename')

Requests information about an entry with a specific UNIX file name. It allows you to list information about backups for a specific UNIX file. If a UNIX directory is specified, 0 entries are returned. FILE is an acceptable abbreviation. Single quotation marks are required to enclose the absolute UNIX path name.

FILELEVEL('pathname')

Lists information about backups for all files under a specific UNIX directory. If the path is not for a directory, 0 entries are returned. FLEVEL is an acceptable abbreviation. Single quotes are required to enclose the absolute UNIX path name.

HOST: Listing all control data sets serialized by the DFSMSHsm host

Explanation: HOST is a required parameter you use in a multiple-DFSMSHsm-host environment to request a list of all control data set records currently serialized by the specified DFSMSHsm host. The serialized control data set records are:

- Backup cycle volume record (BVR)
- Copy pool dump version record (FRD)
- MCDS data set record (MCD)
- MCDS volume record (MCV)
- Tape table of contents record (TTOC)

You can issue the LIST HOST command to receive information about SMS-managed volumes that have MCV records serialized with a host-ID. For *hostid*, substitute the identification character of the DFSMSHsm host whose serialization information you want listed.

Defaults: None

MIGRATIONVOLUME, MIGRATIONLEVEL1, MIGRATIONLEVEL2, and VOLUME: Listing migration volumes

Explanation: MIGRATIONVOLUME, MIGRATIONLEVEL1, MIGRATIONLEVEL2, and VOLUME are mutually exclusive, required parameters you specify to request a list of selected information for migration volumes. These parameters can also be used as subparameters of the SELECT parameter.

MIGRATIONVOLUME

specifies that you want a list of the volume entries for all migration volumes.

MIGRATIONLEVEL1

specifies that you want a list of volume entries for migration level 1 volumes.

You can narrow down the kind of migration level 1 volumes you list using the optional SELECT parameter:

SELECT(NOOVERFLOW)

lists the volume entries for all NOOVERFLOW ML1 volumes.

SELECT(OVERFLOW)

lists the volume entries for all OVERFLOW ML1 volumes.

Defaults: If you specify MIGRATIONLEVEL1 alone (without the SELECT subparameter), which is the default, you will get a list of all migration level 1 volumes.

MIGRATIONLEVEL2

specifies that you want a list of the volume entries for all migration level 2 volumes.

DASD

lists the volume entries for all DASD migration level 2 volumes is generated.

TAPE

lists the volume entries for all tape migration level 2 volumes is generated.

VOLUME

specifies that you want a list of all primary and migration volumes. VOLUME(*volser*) specifies that you want a list of a particular primary or migration volume. For *volser*, substitute the serial number of the specific volume you want listed. Both SMS and non-SMS volumes can be listed.

When using VOLUME(*volser*) as a subparameter of SELECT and when *volser* is specified as a backup volume, the output contains a list of the latest backup versions on the volumes.

Defaults: If you specify VOLUME without *volser*, DFSMSHsm lists all the primary and migration volumes it owns or manages. If you specify PRIMARYVOLUME without *volser*, DFSMSHsm lists all primary volumes it owns or manages.

Note:

1. When you specify any of these parameters, DFSMSHsm does not list information about the data sets on the primary or migration volume, and DFSMSHsm does not give you any information about the amount of space left on the volume. To find out how much space is left on a primary or migration level 1 volume, use the SPACE parameter of the QUERY command.

PRIMARYVOLUME: Listing primary volumes

Explanation: PRIMARYVOLUME is a required parameter you can specify to list selected information for primary volumes.

PRIMARYVOLUME

specifies that you want a list of the volume entries for all primary volumes.

PRIMARYVOLUME(*volser*)

specifies that you want a list for a particular primary volume. For *volser*, substitute the serial number of the specific volume you want listed. Both SMS and non-SMS volumes can be listed.

Defaults: If you specify PRIMARYVOLUME without *volser*, DFSMSHsm lists all primary volumes it owns or manages except when the optional ALLDUMPS parameter is specified.

Note:

1. The PRIMARYVOLUME parameter returns a list of copy pools that the volume is a part of when BCDS or BOTH are specified.
2. The PRIMARYVOLUME(*volser*) parameter returns a list of copy pool dump generations that the volume is a part of when ALLDUMPS and BCDS are both specified.
3. When you specify PRIMARYVOLUME, DFSMSHsm does not list information about the data sets on the primary volume, and DFSMSHsm does not give you any information about the amount of space left on the volume. To find out how much space is left on a primary volume, use the SPACE parameter of the QUERY command.

TAPETABLEOFCONTENTS: Listing tape volume information from the OCDS

Explanation: TAPETABLEOFCONTENTS is a required parameter you can specify to list tape volume information from an OCDS, without listing all the data-set-related information for the volume.

You can direct the output of the list command using the optional parameters OUTDATASET(*dsname*) | SYSOUT(*class*) | TERMINAL.

Defaults: LIST TAPETABLEOFCONTENTS defaults to a condition of NODATASETINFORMATION. When you require data set information, use the LIST TAPETABLEOFCONTENTS(*volser*) command, which is discussed in the following topic.

TAPETABLEOFCONTENTS(*volser*): Listing tape volume information and data set names from the TTOC

Explanation: TAPETABLEOFCONTENTS(*volser*) is a required parameter that can be specified to list migration level 2 or backup tape volume information from the OCDS. At your option, you can choose whether or not to list data set information for the volume. For *volser*, substitute the serial number of the specific tape volume you want listed. If you specify a *volser* shorter than 6 characters, you must enclose it in quotes and pad it to the right with blanks.

If the volume specified in the LIST command is added to DFSMSHsm's control (ADDVOL) as an esoteric unit, the field, UNIT NAME, that appears in the LIST output, contains the esoteric unit name until DFSMSHsm uses the volume. Once DFSMSHsm has used the volume, this UNIT NAME field will contain the generic unit name.

You can direct the output of the list command using the optional parameters OUTDATASET(*dsname*) | SYSOUT(*class*) | TERMINAL.

You can specify NODATASETINFORMATION | DATASETINFORMATION. When you specify NODATASETINFORMATION, you will get a list of volume information, without data set information. When you specify DATASETINFORMATION, you will get a list of volume information and data set information for the volume.

If you use extended TTOCs at your installation and you do not use 3390s for SYSALLDA, consider patching MCVTODS_SECONDARY(PATCH .MCVT.+4b2) to a be X'A0' (160 tracks).

Defaults: LIST TAPETABLEOFCONTENTS(*volser*) defaults to DATASETINFORMATION. You can change this to NODATASETINFORMATION when you do not require data set information for the volume.

USER: Listing user entries

Explanation: USER is a required parameter you specify to request a list of DFSMSHsm authorized users. For *userid*, substitute a string of 1-8 alphanumeric characters for the identification of the user whom you want listed.

Defaults: If you specify USER without *userid*, all user entries are listed from the MCDS.

Note:

1. The USER list does not include any information about data sets associated with specified users.

Optional parameters of the LIST command

When you issue the LIST command, you may specify one or more of the optional parameters. Refer to the syntax diagrams at the beginning of this topic for the recommended use of the LIST command and its sub-parameters.

ALLDUMPS and BACKUPCONTENTS: Determining existence of dumps or listing incremental backup contents

Explanation: ALLDUMPS and BACKUPCONTENTS(*nn*) are optional parameters you specify to determine the existence of all full-volume dumps and to determine the contents of the user volume at the time of incremental backup.

ALLDUMPS

is specified to determine the existence of all volume dumps and on which dump volumes the dumps exist for a single level 0 or migration level 1 volume. The following information is provided for each selected dump:

- Source volume serial number
- Copy pool dump generations
- Dump class
- Dump date
- Dump time
- Dump relative generation number
- Dump expiration date
- Volume serial numbers of the set of volumes containing the dump

BACKUPCONTENTS(*nn*)

when used with the PRIMARYVOLUME(*volser*) parameter, is specified to list the contents of the user volume when the latest or next-to-latest backup of (*volser*) was made. For *nn*, substitute a 0 or 1 to indicate whether the latest or the next to latest backup volume table of contents (VTOC) is listed. A 0 (zero) indicates the latest. A 1 (one) indicates the next latest.

The listing is from the requested VTOC data set. If 0 is specified, the listing represents the contents of the user volume at the time when the last incremental backup was done.

BACKUPCONTENTS, when used with the PRIMARYVOLUME(*volser*) parameter, is not processed for volumes that have not been incrementally backed up by DFHSM 2.3.0 or a subsequent release. This happens because the VTOC copy data set is not on a migration level 1 volume for incremental volume backups performed by previous releases of DFSMSHsm. With the PRIMARYVOLUME, the listing can include data sets that are not eligible for incremental backup by DFSMSHsm. In addition to information provided for this variation of the command, the following information is provided:

- Data set name
- Data set organization
- Date of last reference to data set

- Data set creation date

Defaults: If you do not specify a 0 or 1 for *nn*, the default is 0 (list the latest backup VTOC copy data set). If you specify any value other than 0 or 1, the default is 0.

Note:

1. BACKUPCONTENTS is meaningful only when *volser* is specified with the PRIMARYVOLUME parameter.
2. The BACKUPCONTENTS parameter is not equivalent to the DUMPCONTENTS parameter.
3. BACKUPCONTENTS is ignored when DUMPVOLUME is specified.
4. ALLDUMPS is valid with PRIMARYVOLUME and VOLUME when specified with *volser* and with BCDS.
5. Since DFSMSHsm does not handle partitioned data set members individually, the entire partitioned data set will be listed.

ALLVOLS, DATASETS, DUMPVOLS, FASTREPLICATIONVOLS, and NOVOLS: Listing a specific type of copy pool information

Explanation: ALLVOLS, DATASETS, DUMPVOLS, FASTREPLICATIONVOLS, and NOVOLS are mutually exclusive, optional parameters that you can use to obtain specific information about copy pools, as follows:

ALLVOLS

requests a list of DASD source and target pairs, and all dump volume information.

ALLVERS

specifies that DFSMSHsm is to list all DASD and dump versions.

GENERATION(*gennum*)

specifies that DFSMSHsm is to list a particular generation.

The default with ALLVOLS is GENERATION(0). It lists only the most recent DASD and dump copies.

TOKEN(*token*)

specifies that DFSMSHsm is to list the version with the specified token.

DATASETS

requests a list of data sets associated with a given version, including the data sets that were backed up, but have been deleted. DATASETS returns information only if the catalog information about a backup version is captured at the time of backup.

GENERATION(*gennum*)

specifies that DFSMSHsm is to list data sets associated with a particular generation. The default, DATASETS GENERATION(0), lists only the most recent DASD and dump copies.

TOKEN(*token*)

specifies that DFSMSHsm is to list the data sets associated with the version with the specified token.

Note: You must specify either GENERATION or TOKEN to identify the version.

DUMPVOLS

requests a list of source and dump volume serial numbers for all dump versions.

FASTREPLICATIONVOLS

FRVOLS

requests a list of source and target pairs.

NOVOLS

requests a list of dump class information, if a dump copy version exists. This parameter does not list any data for volume pairs.

Defaults: FASTREPLICATIONVOLS is the default. The default when ALLVOLS or DATASETS is specified is GENERATION(0).

BACKUPCONTROLDATASET, MIGRATIONCONTROLDATASET, and BOTH: Controlling the source of information

Explanation: BACKUPCONTROLDATASET, MIGRATIONCONTROLDATASET, and BOTH are mutually exclusive, optional parameters. When the information to be listed can exist in the MCDS and BCDS, you can tell DFSMSHsm where to look for the information. These parameters apply to data set entries, UNIX file entries, primary volume entries, and migration volume entries.

BACKUPCONTROLDATASET

is specified to request a list of the BCDS entries for a specific data set, level, UNIX file, UNIX directory, or volume, or for migration volumes, or primary volumes. BACKUP is allowed as an alias for BACKUPCONTROLDATASET. Currently information for UNIX files exists in the BCDS only. If MCDS is specified or defaulted to, 0 entries will be returned and ARC0138I will be issued.

MIGRATIONCONTROLDATASET

is specified to request a list of the MCDS entries for a specific data set, level, or volume, or for migration volumes, or primary volumes. MIGRAT is allowed as an alias for MIGRATIONCONTROLDATASET.

BOTH

is specified to request a list of the MCDS and BCDS data set entries.

Defaults: The default is MIGRATIONCONTROLDATASET.

DATASETINFORMATION and NODATASETINFORMATION: Including or not including data set information

Explanation: DATASETINFORMATION and NODATASETINFORMATION are mutually exclusive, optional parameters that can be used only when you specify LIST TAPETABLEOFCONTENTS(*volser*).

DATASETINFORMATION

When you specify DATASETINFORMATION, you get a list of volume-related information and data set information for the volume.

NODATASETINFORMATION

When you specify NODATASETINFORMATION, you get a list of only volume-related information. Data set information for the volume is not included.

Defaults: LIST TAPETABLEOFCONTENTS(*volser*) defaults to DATASETINFORMATION.

DATE and VERSION: Specifying the date or version of the aggregate group

Explanation: DATE and VERSION are optional parameters of AGGREGATE.

DATE(*yyyy/mm/dd*)

specifies that all versions of the aggregate group created on that date are listed:

- *yyyy* is the 4-digit year (1992)
- *mm* is the month (01-12)
- *dd* is the day of the month (01-31)

VERSION(*nnnn*)

specifies the version number of a specific aggregate group version to be listed. VERSION is not valid when AGGREGATE(*) is specified.

Defaults: If you do not specify either DATE or VERSION with the LIST AGGREGATE command, you receive a listing of all active versions of an aggregate.

DUMPCONTENTS: Listing contents of a set of dump volumes

Explanation: DUMPCONTENTS is an optional parameter. You can specify a contents list for a set of dump volumes. The listing is from a full-volume dump VTOC copy data set and represents the contents of the user volume at the time the dump was done.

For any dump volume that is stacked with more than one valid dump copy, you must specify the *srcvol* subparameter to indicate which dump contents you desire.

The following information is listed for each data set on the source volume:

- Data set name
- Data set organization
- Multivolume or single volume
- Data set creation date
- Date data set was last referred to
- Data set expiration date
- RACF indicated
- Password protected
- Change indicated

Defaults: None.

Note:

1. DUMPCONTENTS is meaningful only when *volser* is specified with the DUMPVOLUME parameter.
2. The DUMPCONTENTS parameter is not equivalent to the BACKUPCONTENTS parameter.
3. DUMPCONTENTS is ignored when PRIMARYVOLUME is specified.
4. DUMPCONTENTS applies only to dumps made from level 0 volumes. VTOC copies are not made for migration level 1 volumes. If this parameter is specified for a migration level 1 dump copy, it is ignored.
5. When %CLOUD is specified as the DUMPVOLUME *volser*. The DUMPCONTENTS(*srcvol*) parameter is mandatory for LIST DUMPVOLUME(%CLOUD). SELECT(DDNAME(%CLOUD)) is optional.

INCLUDEPRIMARY: Listing entries for any data set that has migrated

Explanation: INCLUDEPRIMARY is an optional parameter requesting that a list of data set entries include data set entries from the MCDS, even though DFSMSHsm might have already recalled the data sets. A recalled data set can still have an entry in the MCDS.

Defaults: If you do not specify INCLUDEPRIMARY, DFSMSHsm does not list entries for recalled data sets.

Note:

1. The INCLUDEPRIMARY parameter applies only to the DATASETNAME and LEVEL parameters and only when DFSMSHsm lists information from the MCDS. If you specify INCLUDEPRIMARY when it does not apply, DFSMSHsm ignores it.

OUTDATASET, SYSOUT, and TERMINAL: Specifying the location of output for the list

Explanation: OUTDATASET, SYSOUT, and TERMINAL are mutually exclusive, optional parameters specifying the output location for the list.

OUTDATASET(*dsname*)

specifies the name of the data set where DFSMSHsm is to write the output data. For *dsname*, substitute the fully qualified name of the data set to receive the LIST command output.

If the data set does not exist, DFSMSHsm dynamically allocates and catalogs an output data set with the following characteristics:

- Data set name specified (*dsname*)
- Record format of fixed-blocked with ANSI control characters (FBA)
- Logical record length of 121
- Data set is system reblockable
- Primary allocation of 20 tracks (see third note)
- Secondary allocation of 50 tracks (see third note)
- Unit of SYSALLDA (see third note)

If the data set already exists, DFSMSHsm will use the data set. The data set must have the following characteristics:

- The data set must be cataloged and on DASD.
- The data set record format must be FBA, and the logical record length must be 121.
- The data set must be system reblockable.
- The user chooses the primary space allocation.
- DFSMSHsm uses a secondary space allocation of 50 tracks, if DFSMSHsm needs additional extents after the primary space allocation. (See third note.)
- If the data set does not contain data, DFSMSHsm starts writing output data at the beginning of the data set.
- If the data set contains data, DFSMSHsm writes the output data after the existing data.

SYSOUT(class)

specifies that the list is printed to the specified system output class. For *class*, substitute the alphanumeric character for the system output class you want.

TERMINAL

specifies that the list is printed at your terminal.

Defaults: The default is SYSOUT, and *class* defaults to the class specified with the SETSYS SYSOUT command. If you did not specify the type of class with the SETSYS command, the default is class A.

Note:

1. DFSMSHsm does not handle partitioned data set members individually. If you specify a partitioned data set with a member name as the output data set, the list can be written over existing data.
2. If you specify an output data set that does not have the required characteristics, I/O errors can occur in your output data set.
3. If you select the OUTDATASET option, you can use the PATCH command to change the unit name, primary allocation, and secondary allocation. If you select the SYSOUT option, you can use the PATCH command to change whether, and how, DFSMSHsm limits the lines of SYSOUT output. See [*z/OS DFSMSHsm Implementation and Customization Guide*](#) for a description of the PATCH commands available.

RECURSE: Examine subdirectories of the specified UNIX file path

Explanation: RECURSE is optional parameter accepted with the FILELEVEL parameter that requests that subdirectories in the DFSMSHsm inventory be traversed recursively to list backup versions of UNIX files.

Defaults: DFSMSHsm does not use RECURSE.

RESET: Requesting that the host ID field indicate no serialization

Explanation: RESET is an optional parameter specifying that DFSMSHsm reset the host ID field in the records to indicate no serialization by the DFSMSHsm host specified on the command. You use this parameter in a multiple-DFSMSHsm-host environment. Then, if one of the DFSMSHsm hosts becomes inoperative, the serialization done by that DFSMSHsm host remains in the control data set records. No

other DFSMSHsm host can access the serialized resource. DFSMSHsm issues message ARC0817I to show whether the record is reset successfully.

Defaults: None.

Note:

1. The RESET parameter applies only to the HOST parameter. If you do not specify HOST, DFSMSHsm ignores the RESET parameter.

SELECT: Listing only those data set or volume entries that meet selection criteria

Explanation: SELECT, along with its subparameters, is used to specify that only selected entries are listed. SELECT and each of its subparameters is optional. Refer to the syntax diagrams at the beginning of this topic for the recommended use of SELECT and its subparameters.

The optional subparameters of the SELECT parameter are listed in alphabetical order and can be found at the following locations:

Subparameter	See
ACTIVE and RETAIN_DAYS	“SELECT (ACTIVE and RETAIN_DAYS): Displaying information about active and retained backup copies” on page 938
AGE	“SELECT (AGE): Specifying data set use by age” on page 938
ALTERNATEVOLUME and NOALTERNATEVOLUME	“SELECT (ALTERNATEVOLUME and NOALTERNATEVOLUME): Listing tape volumes that have alternate tape volumes” on page 938
AVAILABLE	“SELECT (AVAILABLE): Listing available dump volumes for dump output” on page 939
CAPACITYMODE	“SELECT (CAPACITYMODE): Identifying tape volumes by their ability to use available tape capacity” on page 939
CONNECTED and NOTCONNECTED	“SELECT (CONNECTED and NOTCONNECTED): List volumes that contain spanning data sets ” on page 939
DISASTERALTERNATEVOLUMES	“SELECT (DISASTERALTERNATEVOLUMES): Listing tape volumes that have disaster alternate volumes” on page 940
DUMPCLASS	“SELECT (DUMPCLASS)” on page 940
EFMT1, EFMT2, EEFMT2, EFMT3, EEFMT3, EFMT4, EEFMT4, ENCRYPTED, and NOTENCRYPTED	“SELECT (EFMT1, EFMT2, EEFMT2, EFMT3, EEFMT3, EFMT4, EEFMT4, ENCRYPTED, and NOTENCRYPTED): Recording format” on page 941
EMPTY, FULL, NOTFULL, ASSOCIATED, NOTASSOCIATED, and RECYCLETAKEAWAY	“SELECT (EMPTY, FULL, NOTFULL, ASSOCIATED, NOTASSOCIATED, ERRORALTERNATE, and RECYCLETAKEAWAY): Identifying tape volume capacity” on page 941
ERRORALTERNATE	“SELECT (EMPTY, FULL, NOTFULL, ASSOCIATED, NOTASSOCIATED, ERRORALTERNATE, and RECYCLETAKEAWAY): Identifying tape volume capacity” on page 941

Subparameter	See
EXCESSIVEVOLUMES	“SELECT (EXCESSIVEVOLUMES): Listing volumes with data sets spanning four or more volumes” on page 942
EXPIRED	“SELECT (EXPIRED): Listing expired dump volumes” on page 942
FAILEDCREATE	“SELECT (FAILEDCREATE): Listing all tape volumes that have incomplete data set information saved in the OCDS TTOC record” on page 943
FAILEDRECYCLE	“SELECT (FAILEDRECYCLE): Listing only those volumes that have failed recycle” on page 943
FASTREPLICATIONSTATE, DUMPSTATE, INCOMPLETERECOVERY and NONPREFERREDFRVOLS	“SELECT (DUMPSTATE, FASTREPLICATIONSTATE, INCOMPLETERECOVERY, and NONPREFERREDFRVOLS): Listing specific information about copy pools” on page 940
LIBRARY and NOLIBRARY	“SELECT (LIBRARY and NOLIBRARY): Listing tape volume library status” on page 943
MIGRATIONLEVEL1, MIGRATIONLEVEL2, VOLUME, BACKUP, and BOTH	“SELECT (MIGRATIONLEVEL1, MIGRATIONLEVEL2, VOLUME(volser), BACKUP, and BOTH): Requesting specific information for volumes” on page 944
MULTIPLEVOLUME	“SELECT (MULTIPLEVOLUME): Listing multiple-volume non-VSAM data sets from backup or dump VTOC copy data sets” on page 944
NORETENTIONLIMIT	“SELECT (NORETENTIONLIMIT): List dump volumes with no retention period” on page 945
OVERFLOW and NOOVERFLOW	“SELECT (OVERFLOW and NOOVERFLOW): List overflow or non-overflow ML1 volumes” on page 945
RECALLTAKEAWAY	“SELECT (RECALLTAKEAWAY): Listing ML2 volumes taken away by recall or ABACKUP functions” on page 945
RESUMEAUDIT	“SELECT (RESUMEAUDIT): Listing tape volumes that are capable of being resumed after AUDIT MEDIACONTROLS was interrupted” on page 945
SMALLDATASETPACKING and NOSMALLDATASETPACKING	“SELECT (SMALLDATASETPACKING and NOSMALLDATASETPACKING): Specifying whether data sets have migrated to SDSP” on page 946
UNAVAILABLE	“SELECT (UNAVAILABLE): Listing dump volumes with invalid contents” on page 946
UNEXPIRED	“SELECT (UNEXPIRED): Listing dump volumes that have not reached their expiration date” on page 946
VSAM	“SELECT (VSAM): Specifying the data set organization” on page 947

Defaults: None.

SELECT (ACTIVE and RETAIN_DAYS): Displaying information about active and retained backup copies

Explanation: ACTIVE and RETAIN_DAYS are optional subparameters of SELECT that specify what information is to be displayed about active and retained backup copies

ACTIVE

displays the backup information about the active copies of the data set or UNIX files and the total number of backup copies.

RETAIN_DAYS

displays the backup information about the active and retained backup copies that have RETAIN_DAYS specified at the time of backup. Version and generation number do not apply to retained backup copies, and therefore are not displayed in the LIST output.

Defaults: If you specify SELECT without specifying either RETAIN_DAYS or ACTIVE, DFSMSHsm displays information about both active and retained backup copies. The output contains zeros for generation and version numbers for retained copies, and "*****" in the field of the RETAIN_DAYS value for copies that do not have RETAIN_DAYS specified during backup.

SELECT (AGE): Specifying data set use by age

Explanation: AGE is an optional subparameter of SELECT, specifying a list of entries for those data sets in the MCDS that have been referred to within the specified range of days. AGE is also used for those data sets in the BCDS that have been backed up within the specified range of days. The values to specify when using the AGE parameter are explained:

mindays

specifies the minimum number of days since the data sets have been referred to or backed up. For *mindays*, substitute a decimal number from 0 to 999999.

maxdays

specifies the maximum number of days since the data sets have been referred to or backed up. Specify a *maxdays* value greater than or equal to the *mindays* value. For *maxdays*, substitute a decimal number from 0 to 999999.

Defaults: The default for *mindays* is 0. The default for *maxdays* is 999999. The default range causes entries to be listed for all data sets that meet the data set or volume-selection criteria. If *maxdays* is less than *mindays*, *maxdays* defaults to the same value as *mindays*.

Note:

1. AGE applies to information from the MCDS and BCDS.

SELECT (ALTERNATEVOLUME and NOALTERNATEVOLUME): Listing tape volumes that have alternate tape volumes

Explanation: ALTERNATEVOLUME and NOALTERNATEVOLUME are mutually exclusive optional subparameters used with LIST TAPETABLEOFCONTENTS when *volser* is not specified.

ALTERNATEVOLUME

lists only those tape volumes that are marked full or partially full and that have alternate tape volumes. This list provides you with the capability to identify the full or partially full backup and migration tape volumes that have alternate tapes. In a duplex tape environment, you want to keep those alternate tapes in the library because they can be extended. In a nonduplex tape environment, you may want to EJECT those alternate tapes from the library or initially not put them in the library.

NOALTERNATEVOLUME

lists only those tape volumes that are marked full or partially full and have neither alternate tape volumes nor DISASTERALTERNATEVOLUMES.

Defaults: None.

SELECT (AVAILABLE): Listing available dump volumes for dump output

Explanation: AVAILABLE is an optional subparameter of SELECT specifying a list of each dump volume that is available for dump output use. An available volume is one that has been added to DFSMSHsm's inventory but never used or that has been used but has since been invalidated and remains under DFSMSHsm ownership. The following information is listed for each available dump volume:

- Volume serial number
- Unit type
- Dump class

Defaults: None.

SELECT (CAPACITYMODE): Identifying tape volumes by their ability to use available tape capacity

Explanation: CAPACITYMODE is an optional subparameter of SELECT. The parameters of the CAPACITYMODE parameter are explained:

COMPATIBILITY

lists all nonempty tape volumes that are written by an IBM 3590 in 3490 emulation mode that are not written in CAPACITYMODE(EXTENDED) mode. Data on these tapes is accessible to IBM 3590 drives emulating 3490 drives, regardless of CAPACITYMODE switching support.

EXTENDED

lists all nonempty tape volumes that are written by an IBM 3590 in 3490 emulation mode that were written in CAPACITYMODE(EXTENDED) mode. Data on these tapes is accessible only to tape units capable of CAPACITYMODE switching. On input, DFSMSHsm dynamically switches temporarily to CAPACITYMODE(EXTENDED) to read the tape even if the unit name is in CAPACITYMODE(COMPATIBILITY).

Defaults: None.

SELECT (CONNECTED and NOTCONNECTED): List volumes that contain spanning data sets

Explanation: CONNECTED and NOTCONNECTED are mutually exclusive optional subparameters of SELECT.

CONNECTED

specifies a list of tape volumes that contain spanning data sets. The tapes are grouped based on which volumes are associated with their spanning data sets. CONNECTED lists all connected sets where each set is shown in the connected order. The order of the sets is in collating order of the first tape volser in each set. The collection of sets lists all tapes that contain spanning, valid data sets.

CONNECTED(volser)

specifies that the tape volume set that is associated with *volser* be listed. All tapes in the connected set associated with this *volser* are listed.

NOTCONNECTED

specifies a list of tape volumes that excludes all that contain spanning data sets. Any tape in this list can be moved into or out of a library without having to find a tape containing another part of a spanning data set. Tapes that are not connected can be full or partially full (empty tapes are not included). The SELECT subparameters FULL or NOTFULL may be used to further restrict the listing. Output from the NOTCONNECTED subparameter identifies those tape volumes that can be individually inserted into or ejected from a tape library.

Defaults: None.

Note:

1. Output from the CONNECTED subparameter identifies which tape volumes should be inserted or ejected from a tape library at the same time.
2. Do not use any other subparameters on SELECT when specifying CONNECTED(*volser*).
3. Do not use the EMPTY, FULL, NOTFULL, ALTERNATE, NOALTERNATE, or DIASSTERALTERNATEVOLUMES subparameters when specifying CONNECTED.

SELECT (DIASSTERALTERNATEVOLUMES): Listing tape volumes that have disaster alternate volumes

Explanation: DIASSTERALTERNATEVOLUMES is an optional subparameter of SELECT to list TAPETABLEOFCONTENTS. DIASSTERALTERNATEVOLUMES will list only tape volumes whose TTOCs have been flagged as having a disaster alternate volume.

Defaults: None.

SELECT (DUMPCLASS)

DUMPCLASS can be used as an optional parameter of the LIST command and as a subparameter of SELECT. DUMPCLASS is explained in [“DUMPCLASS: Listing dump class profiles”](#) on page 927.

SELECT (DUMPSTATE, FASTREPLICATIONSTATE, INCOMPLETERECOVERY, and NONPREFERREDFRVOLS): Listing specific information about copy pools

Explanation: DUMPSTATE, FASTREPLICATIONSTATE, INCOMPLETERECOVERY, and NONPREFERREDFRVOLS are mutually exclusive optional subparameters of SELECT.

DUMPSTATE

requests information about the FRBACKUP DUMP.

The following parameters are mutually exclusive, optional parameters with the DUMPSTATE parameter.

ALLCOMPLETE

indicates that all volumes have been successfully dumped to the dump classes specified.

REQUIREDCOMPLETE

indicates that all volumes have been successfully dumped to at least the required dump classes.

PARTIAL

indicates that some volumes of a copy pool have been successfully dumped.

NONE

indicates that no dumps are associated with the copy pool.

FASTREPLICATIONSTATE

requests information about the FRBACKUP DASD copy.

The following parameters are mutually exclusive, optional parameters of the FASTREPLICATIONSTATE parameter.

FAILED

indicates the DASD version is not recoverable due to one of the following reasons:

- a failure occurred during an FRBACKUP or WITHDRAW operation before all volumes in the copy pool were successfully copied.
- the backup version was invalidated during an FRRECOV operation following a successful fast reverse restore.

FCFRRINCOMPLETE

indicates that the copy pool has a DASD version that was not completely recovered using fast reverse restore. Some volumes have been recovered while other volumes have remained unrecovered. This state typically indicates that either the recovery is in process or ended without recovering all volumes.

RECOVERABLE

indicates that the copy pool has a DASD copy that can be recovered.

NONRECOVERABLE

indicates that the copy pool has a DASD version that was created with the VERSIONS=0 and ALLOWFCFRR=N copy pool values.

NONE

indicates a DASD copy does not exist.

INCOMPLETERECOVERY

requests information about copy pool FRBACKUP versions that have not been recovered completely. This refers to either DASD recovery using fast reverse restore or recovery from tape.

NONPREFERREDFRVOLS

requests information about DASD source and target pairs that are recognized by DFSMSHsm as non-preferred.

During fast replication environment setup, DFSMSHsm assigns eligible copy pool backup storage group volumes to the source volumes and prefers target volumes that are in the same cluster as the source volumes on applicable devices. When such a volume is not available, any volumes eligible for fast replication are assigned.

With NONPREFERREDFRVOLS, DFSMSHsm lists only the non-preferred source and target volume pairs. The preferred pairs are omitted. Volume pairs that were assigned without enabling the volume preferencing function are not listed. Information for non-preferred pairs is available when all of the following conditions are met:

- Volume preferencing function was used when the fast replication environment was set up and one or more pairs are non-preferred
- ALLVOLS or FRVOLS is specified, or FRVOLS is in effect by default
- A specific copy pool name is specified to request detailed information
- The specified copy pool is defined with one or more backup versions.

Defaults: None.

SELECT (EFMT1, EFMT2, EEFMT2, EFMT3, EEFMT3, EFMT4, EEFMT4, ENCRYPTED, and NOTENCRYPTED): Recording format

Explanation: EFMT1, EFMT2, EEFMT2, EFMT3, EEFMT3, EFMT4, EEFMT4, ENCRYPTED, and NOTENCRYPTED are mutually exclusive optional subparameters of SELECT.

Defaults: None.

SELECT (EMPTY, FULL, NOTFULL, ASSOCIATED, NOTASSOCIATED, ERRORALTERNATE, and RECYCLETAKEAWAY): Identifying tape volume capacity

Explanation: EMPTY, FULL, NOTFULL, ASSOCIATED, NOTASSOCIATED, ERRORALTERNATE, and RECYCLETAKEAWAY are mutually exclusive optional subparameters of SELECT. The FULL, NOTFULL, ASSOCIATED, NOTASSOCIATED, ERRORALTERNATE, and RECYCLETAKEAWAY subparameters apply only to the LIST TTOC command. The EMPTY subparameter applies to the following commands:

- LIST BACKUPVOLUME
- LIST DATASETNAME
- LIST MIGRATIONLEVEL2(TAPE)
- LIST TTOC

EMPTY

specifies a list of empty tape backup volumes or empty tape level 2 migration volumes. If you specify SELECT(EMPTY) with LIST DATASETNAME MCDS, then DFSMSHsm will list only those migrated data sets that were empty when they were migrated.

FULL

specifies a list of tape volumes that are not available for selection (marked full). This subparameter provides a means of identifying the full backup and migration tape volumes.

NOTFULL

specifies a list of tape volumes that are available for output selection including those that are empty, as well as those that have some data but are not marked full. By using NOTFULL, you have the capability of identifying those tapes that could be the first ones added to a library and made available for backup or migration.

ASSOCIATED

specifies a listing of only those ML2 partial (not empty and not full) tape volumes currently associated for output by specific migration or recycle tasks.

NOTASSOCIATED

specifies a listing of only those ML2 partial tape volumes available for an initial allocation and not currently associated for output by any migration or recycle task.

ERRORALTERNATE

displays the volumes that encountered errors and were subsequently marked full when the SETSYS DUPLEX (MIGRATION(Y (ERRORALTERNATE(MARKFULL) environment was specified.

RECYCLETAKEAWAY

displays the volumes that were not completely recycled because they were taken away.

Defaults: None.

Note:

1. These subparameters cannot be used when *volser* is part of the command.
2. The FULL and NOTFULL subparameters are ignored when used with the RECALLTAKEAWAY subparameter of SELECT.
3. When EMPTY is specified with the BACKUPVOLUME parameter, DASD backup volumes are not listed. EMPTY is ignored when it is specified with MIGRATIONLEVEL2(DASD). You must use the TAPE parameter for the EMPTY subparameter to work.
4. When EMPTY is specified with the DATASETNAME MIGRATIONCONTROLDATASET parameter, the list excludes PDS and PDSE data sets.
5. The ASSOCIATED and NOTASSOCIATED subparameters are ignored when listing TTOCs for backup tape volumes.

SELECT (EXCESSIVEVOLUMES): Listing volumes with data sets spanning four or more volumes

Explanation: EXCESSIVEVOLUMES is an optional parameter used with the LIST TAPETABLEOFCONTENTS command when no *volser* is specified. The EXCESSIVEVOLUMES parameter lists only the volumes that contain a data set that spans four or more volumes.

Defaults: None.

SELECT (EXPIRED): Listing expired dump volumes

Explanation: EXPIRED is an optional subparameter of SELECT specifying a list of each dump volume that is part of a valid dump copy and has reached or passed its expiration date. The following information is listed for each dump volume:

- Source volume serial number
- Dump volume serial number

- Unit type
- Expiration date
- Dump class

Defaults: None.

Note:

1. If DUMPCCLASS(*class*) is also specified in the set of SELECT options, only dump volumes assigned to the specified *class* are listed.
2. The EXPIRED subparameter of the SELECT parameter is relevant when *volser* is not specified with the DUMPVOLUME parameter. The EXPIRED subparameter can be used in combination with the subparameters AVAILABLE, DUMPCCLASS, NORETENTIONLIMIT, UNAVAILABLE, or UNEXPIRED to get a list of volumes that satisfies all the selection criteria.

SELECT (FAILEDCREATE): Listing all tape volumes that have incomplete data set information saved in the OCDS TTOC record

Explanation: FAILEDCREATE is an optional parameter used when you specify LIST TAPETABLEOFCONTENTS without the *volser*.

FAILEDCREATE lists only the volumes that have not been created.

If DFSMSHsm issues message ARC0378I, it indicates the TTOC for a volume is incomplete. Use LIST TAPETABLEOFCONTENTS SELECT(FAILEDCREATE) to list all volumes that have incomplete data set information saved in the OCDS TTOC record.

Defaults: None.

Note:

1. Multihost Consideration: As an example, consider two hosts, 'A' and 'B'. If host 'A' has experienced an outage and has not been restarted, you *cannot* use host 'B' to issue the FAILEDCREATE parameter to list all tape volumes that have incomplete data set information saved in the OCDS TTOC record.
2. If DFSMSHsm has issued message ARC0378I, indicating that there is a mismatch between the OCDS TTOC and the tape media, we recommend using the AUDIT MEDIACONTROLS FIX command to resolve the problem. See [“MEDIACONTROLS VOLUMES, MEDIACONTROLS\(SDSP\) VOLUMES and MEDIACONTROLS\(CLOUD\): Requesting an audit of control information” on page 748](#) for more information. Also see Chapter 75, “Using the AUDIT command,” on page 1255 for more information on using AUDIT MEDIACONTROLS. An explanation of how to correct the mismatch can be found in the [“How to handle inconsistencies between tape media contents and OCDS TTOC records” on page 550](#).

SELECT (FAILEDRECYCLE): Listing only those volumes that have failed recycle

Explanation: FAILEDRECYCLE is an optional parameter used when you specify LIST TAPETABLEOFCONTENTS without the *volser*.

FAILEDRECYCLE lists only the volumes that have failed recycle and which are not selected for processing by a generic RECYCLE command.

Defaults: None.

Note:

1. Input tapes are marked in error when one of the following occurs: an I/O error on the input tape or an OPEN failure for the input tape. These errors prevent a successful recycle of the tape.

SELECT (LIBRARY and NOLIBRARY): Listing tape volume library status

Explanation: LIBRARY and NOLIBRARY are mutually exclusive optional subparameters of SELECT.

LIBRARY

lists only the tape volumes that are in a tape library. LIBRARY(ALTERNATE) lists only the tape volumes that have alternate tape volumes.

NOLIBRARY

lists only the tape volumes that are *not* in a tape library. NOLIBRARY(ALTERNATE) lists only the tape volumes that have alternate tape volumes not in a library.

When using the ALTERNATE subparameter, the listing is independent of whether the ORIGINAL volume is in a library and applies to both BACKUP and ML2 volumes. The field under the heading, LIB, applies to the alternate volume not the original volume. If you specify SELECT(NOALTERNATEVOLUME), the specification of LIBRARY(ALTERNATE) or NOLIBRARY(ALTERNATE) is ignored.

Defaults: None.

Note:

1. If you select LIBRARY with no other parameter, all volumes in your library will be listed. To reduce the run time of the LIST command, narrow the scope of the command with other parameters.

SELECT (MIGRATIONLEVEL1, MIGRATIONLEVEL2, VOLUME(*volser*), BACKUP, and BOTH): Requesting specific information for volumes

Explanation: MIGRATIONLEVEL1, MIGRATIONLEVEL2, VOLUME(*volser*), BACKUP, and BOTH are mutually exclusive, optional parameters you specify to request a list of selected information for migration or backup volumes.

MIGRATIONLEVEL1

specifies that you want a list of the volume entries for all migration level 1 volumes. This parameter is used only with DATASETNAME and LEVEL.

MIGRATIONLEVEL2

specifies that you want a list of the volume entries for all migration level 2 volumes.

VOLUME

specifies that you want a list for a particular volume. For *volser*, substitute the serial number of the specific volume you want listed. When you specify VOLUME(*volser*) to list a backup volume, the output contains a list of the latest backup versions on the volume.

BACKUP

is used only with the LIST TAPETABLEOFCONTENTS with no *volser* specified. You specify BACKUP to list all backup volumes.

BOTH

is specified to list both migration and backup volumes.

Defaults: If the SELECT subparameters RECALLTAKEAWAY, ASSOCIATED, or NOTASSOCIATED is specified, the default is MIGRATIONLEVEL2. Otherwise, the default is BOTH.

SELECT (MULTIPLEVOLUME): Listing multiple-volume non-VSAM data sets from backup or dump VTOC copy data sets

Explanation: MULTIPLEVOLUME is an optional subparameter of SELECT, specifying a list of only multiple-volume non-VSAM data sets. DFSMSHsm treats a non-VSAM data set as multiple-volume if the data set VTOC entry indicates that the data resides on more than one volume. DFSMSHsm does not use the catalog in determining if a non-VSAM data set resides on multiple volumes.

Defaults: None.

Note:

1. If both the MULTIPLEVOLUME subparameter and the VSAM subparameter are specified as SELECT options, all VSAM data sets and all multiple-volume non-VSAM data sets are listed.

SELECT (NORETENTIONLIMIT): List dump volumes with no retention period

Explanation: NORETENTIONLIMIT is an optional subparameter of SELECT specifying a list of each dump volume that is part of a valid dump copy and has no retention date. These dump volumes contain dump copies that DFSMSHsm invalidates automatically only if the dump copy is the 100th generation and a new dump is performed. The following information is listed for each dump volume:

- Source volume serial number
- Dump volume serial number
- Unit type
- Dump class

Defaults: None.

Note:

1. If DUMPCLASS(*class*) is also specified in the set of SELECT options, only dump volumes assigned to the specified *class* are listed.
2. The NORETENTIONLIMIT subparameter of SELECT is relevant only when *volser* is not specified with the DUMPVOLUME parameter. The NORETENTIONLIMIT subparameter can be used in combination with the subparameters AVAILABLE, DUMPCLASS, EXPIRED, UNAVAILABLE, or UNEXPIRED to get a list of volumes that satisfy all the selection criteria.

SELECT (OVERFLOW and NOOVERFLOW): List overflow or non-overflow ML1 volumes

Explanation: OVERFLOW and NOOVERFLOW are mutually exclusive optional subparameters of SELECT used with LIST MIGRATIONLEVEL1 specifying a list of NOOVERFLOW or OVERFLOW ML1 volumes.

NOOVERFLOW

specifies a list of all NOOVERFLOW ML1 volumes.

OVERFLOW

specifies a list of all OVERFLOW ML1 volumes.

Defaults: None.

Note:

1. If OVERFLOW or NOOVERFLOW is not specified, all ML1 volumes are listed.

SELECT (RECALLTAKEAWAY): Listing ML2 volumes taken away by recall or ABACKUP functions

Explanation: RECALLTAKEAWAY is an optional subparameter of SELECT that specifies a list of all ML2 volumes taken away from use as migration tape output because either RECALL or ABACKUP needed a data set on the volume.

Defaults: None.

SELECT (RESUMEAUDIT): Listing tape volumes that are capable of being resumed after AUDIT MEDIACONTROLS was interrupted

Explanation: RESUMEAUDIT is an optional subparameter used when you specify LIST TAPETABLEOFCONTENTS without the *volser*.

RESUMEAUDIT lists only the volumes that were interrupted during AUDIT MEDIACONTROLS and are capable of being resumed.

Defaults: None.

Notes:

1. LIST TTOC SELECT(RESUMEAUDIT) only lists volumes that were previously interrupted during AUDIT MEDIACONTROLS because the volume was held. DFSMSHsm was stopped or emergency mode was specified. RESUME can be specified on the AUDIT MEDCTLVOLUMES(*tape volser*) FIX command to resume audit of a volume at the point it was interrupted.

SELECT (SMALLDATASETPACKING and NOSMALLDATASETPACKING): Specifying whether data sets have migrated to SDSP

Explanation: SMALLDATASETPACKING and NOSMALLDATASETPACKING are mutually exclusive, optional subparameters of SELECT specifying a list of those data sets that have migrated to small-data-set-packing data sets or a list of those data sets that have not migrated to small-data-set-packing data sets. If you do not specify either subparameter, the list contains entries for all migrated data sets.

SMALLDATASETPACKING

specifies a list of entries for those data sets that have migrated to small-data-set-packing data sets.

NOSMALLDATASETPACKING

specifies a list of entries for those data sets that have not migrated to small-data-set-packing data sets.

Defaults: None.

Note:

1. Small-data-set-packing data sets can exist only on migration level 1 volumes. You can specify the MIGRATIONLEVEL1 subparameter of the SELECT parameter and the SMALLDATASETPACKING or NOSMALLDATASETPACKING subparameter of the SELECT parameter in the same LIST command.

Do not specify MIGRATIONLEVEL2 and SMALLDATASETPACKING in the same LIST command. Also, do not specify a migration level 1 volume (VOLUME) and SMALLDATASETPACKING in the same LIST command.

2. These parameters apply only to information in the MCDS.

SELECT (UNAVAILABLE): Listing dump volumes with invalid contents

Explanation: UNAVAILABLE is an optional subparameter of SELECT specifying a list of each dump volume that has had its contents invalidated, but was not eligible to be automatically reused. The following information is listed for each dump volume:

- Dump volume serial number
- Unit type
- Dump class

Defaults: None.

Note:

1. If DUMPCLASS(*class*) is also specified in the set of SELECT options, only dump volumes assigned to the specified *class* are listed.
2. The UNAVAILABLE subparameter of the SELECT parameter is relevant when *volser* is not specified with the DUMPVOLUME parameter. The UNAVAILABLE subparameter can be used in combination with the subparameters AVAILABLE, DUMPCLASS, EXPIRED, NORETENTIONLIMIT, or UNEXPIRED to get a list of volumes that satisfy all the selection criteria.

SELECT (UNEXPIRED): Listing dump volumes that have not reached their expiration date

Explanation: UNEXPIRED is an optional subparameter of SELECT specifying a list of each dump volume that is part of a valid dump copy and has not reached or passed its expiration date. Dump volumes with a retention period of NONE are not included. The following information is listed for each dump volume:

- Source volume serial number for each valid dump copy
- Dump volume serial number
- Unit type
- Expiration date
- Dump class

Defaults: None.

Note:

1. If DUMPCCLASS(*class*) is also specified in the set of SELECT options, only dump volumes assigned to the specified *class* are listed.
2. The UNEXPIRED subparameter of the SELECT parameter is relevant only when *volser* is not specified with the DUMPVOLUME parameter. The UNEXPIRED subparameter can be used in combination with the subparameters AVAILABLE, DUMPCCLASS, EXPIRED, NORETENTIONLIMIT, or UNAVAILABLE to get a list of volumes that satisfy all the selection criteria.

SELECT (VSAM): Specifying the data set organization

Explanation: VSAM is an optional subparameter of SELECT specifying a list of only migrated VSAM data sets. The list contains standard data set information for the MCDS data set records and any VSAM object names that you can use to automatically recall the data set.

Defaults: None.

Note:

1. The VSAM subparameter applies only to data set information from the MCDS.

SUMMARY: Requesting a summary

Explanation: SUMMARY is an optional parameter specifying a count of data sets, tracks, and bytes of the selected data sets.

Defaults: None.

Note:

1. When you specify this parameter, DFSMSHsm does not list information about each data set. This parameter applies only to information from the MCDS.
2. The SUMMARY information is issued when you specify the LEVEL parameter or the DATASETNAME parameter (with no data set name specified) and the MCDS information is requested and available. The SUMMARY parameter applies only to the DATASETNAME parameter (with no data set name specified) and the LEVEL parameter.

Examples of how to code commonly used list requests

The examples as presented below show how to use the LIST command to request various types of commonly desired information. Chapter 76, “Using the LIST command,” on page 1341 contains examples of output produced by the LIST command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Listing migrated data sets contained in the SDSP

In this example, you issued the DATASETNAME command to list the migrated data sets contained in the SDSP data set on a particular migration level 1 (ML1) volume and the results of the request are written to an output data set.


```
LIST DATASETNAME MIGRATIONCONTROLDATASET SELECT(VOLUME(L1VOLA) SDSP)
OUTDATASET(fully.qualified.dsn)
```

Listing migrated data sets using abbreviations

In this example, you used the standard abbreviations to list the migrated data sets.

```
LIST DSN MCDS SEL(VOL(L1VOLA) SDSP)
```

Listing the number and total space of migrated data sets

This example shows you how to list the number and total space of the migrated data sets contained in the SDSP data set on a particular ML1 volume.

```
LIST DATASETNAME MIGRATIONCONTROLDATASET SELECT(VOLUME(L1VOLA) SDSP) SUMMARY
```

Listing migrated data sets not referenced in 180 days

This example shows you how to list the migrated data sets, not referenced in 180 days, on an ML1 volume and how to write the results of the request to the terminal.

```
LIST DSN MCDS SEL(VOL(L1VOLA) AGE(180)) TERM
```

Listing migrated data sets with a high-level qualifier

This example shows you how to list on ML1 volumes the migrated data sets having a common high-level qualifier that have not been referenced in 180 days.

```
LIST LEVEL(L364204) MCDS SELECT(ML1 AGE(180))
```

Listing specific migrated data sets

This example shows you how to request a specific ML1 volume and how to write the results of the request to the terminal.

```
LIST LEVEL(L364204) MCDS SELECT(VOLUME(L1VOLB) AGE(180)) TERM
```

Listing migrated data sets with a high-level and a common secondary qualifier

This example shows you how to list on ML1 volume the migrated data sets with a common high-level qualifier and a common secondary qualifier and how to write the results of the request to the terminal.

```
LIST LEVEL(L364204.TEST) MCDS SELECT(VOLUME(L1VOLB)) TERM
```

Listing copy pool information for all copy pools

In this example, DFSMSHsm lists information for all copy pools..

```
LIST COPYPOOL
```

See [Figure 116 on page 1350](#) for an example of the output.

Listing copy pool information for a specific copy pool

In this example, DFSMSHsm lists detailed information for each backup version of the CPDATABASE1 copy pool.

```
LIST COPYPOOL(CPDATABASE1)
```

See [Figure 116 on page 1350](#) for an example of the output.

Listing specified copy pool storage group information

In this example, DFSMSHsm lists information for each volume in the CPBSGDATABASE4 storage group and the corresponding source volume for each volume in that copy pool backup storage group. This list only returns those volumes that DFSMSHsm has selected as target volumes. For those target volumes that are listed, the source volume is also listed.

```
LIST COPYPOOLBACKUPSTORAGEGROUP(CPBSGDATABASE4)
```

See [Figure 120 on page 1352](#) for an example of the output.

Listing information about data sets backed up as part of a copy pool backup version

In this example, DFSMSHsm lists all data sets backed up as part of a copy pool backup version, including those that have been deleted. The data sets are listed in alphabetical order by catalog.

```
LIST COPYPOOL(CP1) DATASETS(GENERATION(1))
```

Note: DATASETS only returns information related to a copy pool backup version if the catalog information for a given backup version is captured at the time of backup.

See [Figure 120 on page 1352](#) for an example of the output.

Listing summary information from the MCDS

This example shows you how to list summary information from the MCDS and write the results to an output data set.

```
LIST DSN SUMMARY ODS(LIST)
```

See [Figure 123 on page 1354](#) for an example of the output.

Listing specified volume information

This example shows you how to list only the volume information for the specific volume specified by the *volser* and how to write the results of the request to the terminal.

```
LIST TT0C(A00800) NODATASETINFORMATION TERM
```

See [Figure 152 on page 1384](#) for an example of the output.

Listing specified volume and data-set-related information

This example shows you how to list the volume information for the specified volume *volser* and the data-set-related information (the default of DATASETINFORMATION will be used) and how to write the results of the request to an output data set.

```
LIST TT0C(BATP01) OUTPUTDATASET(LIST)
```


See [Figure 153 on page 1385](#) for an example of the output.

Listing volume information for failed RECYCLE ML2

This example shows you how to list the volume information for the migration level 2 volumes that have previously failed RECYCLE and how to write the results of the request to an output data set.

```
LIST TT0C SELECT(ML2 FAILEDRECYCLE) ODS(LIST)
```

See [Figure 157 on page 1386](#) for an example of the output.

Listing tape volumes with incomplete data set information

This example shows you how to list all the tape volumes that have incomplete data set information saved in the OCDS TT0C record.

```
LIST TT0C SELECT(FAILEDCREATE)
```

See [Figure 158 on page 1386](#) for an example of the output.

Listing volume information for data sets spanning four or more volumes

This example shows how to list the volume information for the ML2 and backup volumes that contain data sets that span four or more volumes and how to write the results of the request to an output data set.

```
LIST TT0C SELECT(BOTH EXCESSIVEVOLUMES) ODS(LIST)
```

See [Figure 159 on page 1386](#) for an example of the output.

Listing information for connected volumes

This example shows all volumes that are connected to this volume and whether or not these volumes are in the library.

```
LIST TT0C SELECT(CONNECTED(A00800)) ODS(LIST)
```

See [Figure 151 on page 1384](#) for an example of the output.

Listing volume information about disaster alternate volumes

This example shows how to list the volume information about disaster alternate volumes.

```
LIST TT0C SELECT (LIBRARY DISASTERALTERNATEVOLUMES) ODS(LIST)
```

See [Figure 156 on page 1385](#) for an example of the output.

Listing volume information about alternate tape volumes

This example shows you how to list the volume information for the migration level 2 and backup volumes that do *not* have an ALTERNATE tape volume and are in a tape library. The result is printed to an output data set.

```
LIST TT0C SELECT (LIBRARY NOALTERNATEVOLUME) ODS(LIST)
```

See [Figure 155 on page 1385](#) for an example of the output.

Listing information about available tape volumes

This example shows you how to list tape volumes that are available for selection (not marked full) and thus identify the empty and partially full backup and migration tape volumes. The results go to the SYSOUT class.

```
LIST TT0C SELECT(NOTFULL) SYSOUT(H)
```

Listing information about partially filled ML2 tape volumes not associated for output by any migration or recycle task

This example shows you how to list all partially filled ML2 tapes that are not currently associated with an ML2 output task in any host. The results of the request are written to an output data set.

```
LIST TT0C SELECT(NOTASSOCIATED) ODS('JCA.ASSOCNOT.D041798')
```

See [Figure 160 on page 1386](#) for an example of the output.

Listing information about contents of the dump volume

This example shows you how to list all data sets that have been dumped to a specified dump volume.

```
LIST DUMPVOLUME(A00359) DUMPCONTENTS(PRIM02) ODS(List)
```

See [Figure 137 on page 1370](#) for an example of the output.

Listing information about a dump volume

This example shows how to list all data sets that have been dumped to a volume with a specified *volser*. The output includes the dump copy data set name.

```
LIST DUMPVOLUME(A00359) ODS(LIST)
```

See [Figure 134 on page 1366](#) for an example of the output.

Listing information about a specific dump class profile

This example shows how to list all data sets that have been dumped to a specific DUMPCLASS.

```
LIST DUMPCLASS(WEEKLY) ODS(LIST)
```

See [Figure 131 on page 1363](#) for an example of the output.

Listing information about aggregate data sets

This example shows how to list the aggregate version record (ABR record) Version 2 to the terminal.

```
LIST AGGREGATE(PAY1) VERSION(0002) TERM
```

See [Figure 110 on page 1344](#) for an example of the output.

Listing information about aggregate data sets and saving the output to the ODS

This example shows how to list the aggregate version record (ABR record) Version 1 to the ODS.

```
LIST AGGREGATE(PAY1) VERSION(0001) ODS(LIST)
```


See [Figure 111 on page 1344](#) for an example of the output.

Listing ML2 volumes taken away by recall or ABACKUP

This example shows how to list the ML2 volumes that were taken away from an output function because RECALL or ABACKUP functions need a data set on the volume.

```
LIST TT0C SELECT(RECALLTAKEAWAY)
```

Listing tape volumes that can use available tape capacity

This example shows how to list all tape volumes that are written by an IBM 3590 that is emulating an IBM 3490 in CAPACITYMODE(EXTENDED).

```
LIST TT0C SELECT(CAPACITYMODE(EXTENDED))
```

Listing tape volumes capable of being resumed after AUDIT MEDIACONTROLS was interrupted

This example shows how to list all tape volumes that were interrupted during AUDIT MEDIACONTROLS and are capable of being resumed.

```
LIST TT0C SELECT(RESUMEAUDIT)
```

Chapter 54. LOG command: Entering data into the DFSMSHsm log

The LOG command enters information into the DFSMSHsm log. This information could be a message or an explanation that the storage administrator or system programmer wants to write into the DFSMSHsm log. The information could also be automatically generated DFSMSHsm log data.

Syntax of the LOG command

► LOG — *data* ◄

Required parameters of the LOG command

This topic describes the required parameter of the LOG command.

data: Specifying the data for the log

Explanation

data is a required positional parameter specifying the information to enter into the DFSMSHsm log. For *data*, substitute any character string consisting of alphanumeric characters, special characters, and \$, #, or @.

Defaults

None.

Example of how to code the LOG command

The following example shows how to code the LOG command to enter data into the DFSMSHsm log.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Sending a message to the log

In this example, a message concerning the DFSMSHsm functions is sent to the DFSMSHsm log.

```
LOG ALL MIGRATION FUNCTIONS HELD AT 3:15 AM
```


Chapter 55. MIGRATE command: Requesting a space management function

The MIGRATE command requests the type of space management you want DFSMSHsm to do: migration, data set deletion, or data set retirement. You can issue the MIGRATE command to cause space management for:

- All eligible data sets from a specific level 0 volume. A level 0 volume is an SMS volume in a storage group, a DFSMSHsm primary volume, or a nonmanaged user volume.
- All eligible data sets from all primary volumes (PRIMARY).
- Storage groups.

You can issue the MIGRATE command to cause migration for:

- A specific data set from a level 0 volume to a migration level 1 volume or a migration level 2 volume (DATASETNAME)
- A specific data set from a DASD migration level 2 volume to a tape migration level 2 volume (DATASETNAME and MIGRATIONLEVEL2)
- A specific data set from a migration level 1 volume to a migration level 2 volume (DATASETNAME and MIGRATIONLEVEL2)
- All eligible data sets from all migration level 1 volumes to migration level 2 volumes (MIGRATIONLEVEL1).

On SMS-managed volumes only, you can also issue the MIGRATE command to cause class transitions for:

- All data sets on a specific volume. The management class is used to determine whether a data set is eligible to be migrated, transitioned, or both.
- All data sets in a specific storage group. The management class is used to determine whether a data set is eligible to be migrated, transitioned, or both.
- A specific data set.

On SMS-managed volumes only, you can also issue the MIGRATE command with the MOVE parameter to:

- Move specific data sets from one volume to another
- Move all data sets off a volume
- Move all data sets out of a storage group.

Note:

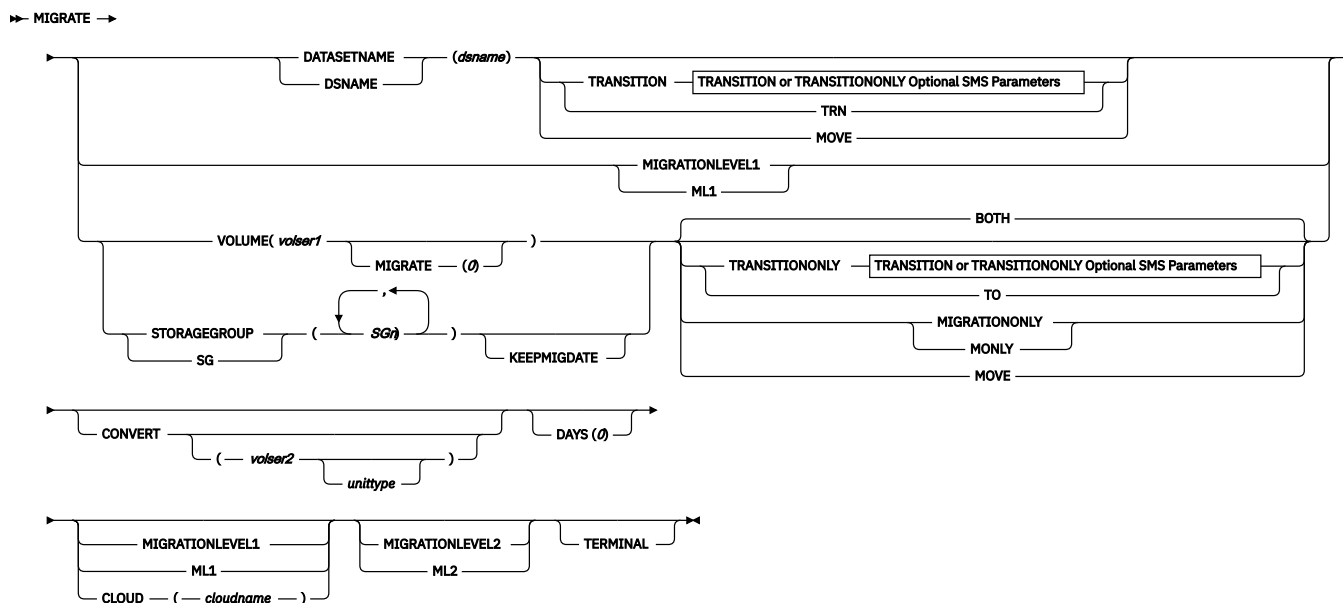
1. If you want to move data sets from one tape migration level 2 volume to another tape migration level 2 volume, you must recycle the tape migration level 2 volume. [Chapter 61, “RECYCLE command: Consolidating valid data to one tape from other tapes,” on page 1015](#) contains information about recycling tape volumes.
2. If you want to move all eligible data sets from a specific migration level 1 volume or a specific migration level 2 DASD volume to other migration level 2 volumes, you must use the FREEVOL command. See [Chapter 48, “FREEVOL command: Freeing DFSMSHsm-owned volumes,” on page 877](#).
3. Fast subsequent migration can take place under both data set and volume MIGRATE commands. For more information on setting up fast subsequent migration, see [“Reconnecting data sets using fast subsequent migration” on page 94](#).
4. For either volumes or storage groups, use the MIGRATIONONLY, TRANSITIONONLY, and BOTH options to indicate that only migrations should be processed, only transitions should be processed, or that both should be processed. The default is to process BOTH.
5. When processing the MIGRATE command, DFSMSHsm checks data sets for migration eligibility before attempting either migration or transition or both.

Requesting a space management function for SMS-managed data sets

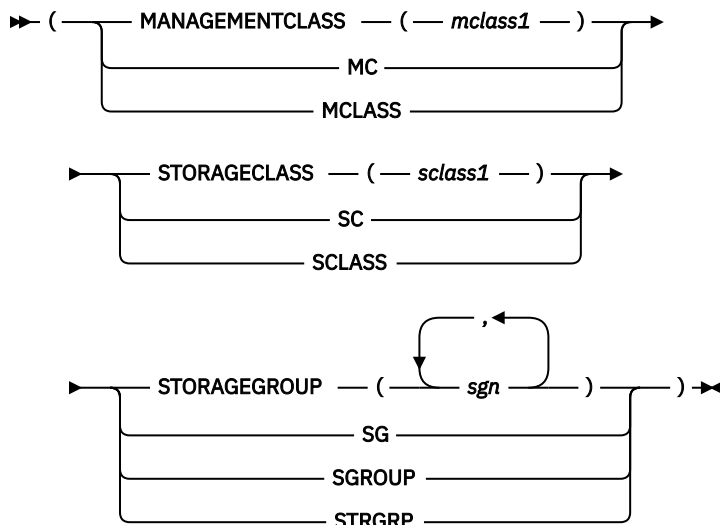
The functions performed for SMS data sets are determined by each data set's management class attributes. COMMAND migration must be allowed by the management class to which the data set belongs.

Note: If a management class cannot be obtained for an SMS-managed data set, the data set will not be migrated, and an error message is issued.

Syntax of the MIGRATE command for SMS-managed data sets



TRANSITION or TRANSITIONONLY Optional SMS Parameters



Note: You must specify either DATASETNAME, MIGRATIONLEVEL1, VOLUME, or STORAGEGROUP with each MIGRATE command.

Summary of parameters

The following table is a summary of the combination of parameters you can specify with the MIGRATE command. Specifying other combinations of parameters results in either an error message or one or more of the parameters being ignored.

Parameter	Related Parameters
CLOUD	VOLUME (MIGRATE) STORAGEGROUP DATASETNAME
CONVERT	VOLUME (MIGRATE)
DATASETNAME	CONVERT MIGRATIONLEVEL1 MIGRATIONLEVEL2 TRANSITION MOVE CLOUD
DAYS	MIGRATIONLEVEL1 VOLUME (MIGRATE) STORAGEGROUP
MIGRATIONLEVEL1	DAYS TERMINAL
MIGRATIONLEVEL2	DATASETNAME
MOVE	VOLUME (MIGRATE) STORAGEGROUP DATASETNAME
STORAGEGROUP	DAYS TRANSITIONONLY MIGRATIONONLY BOTH MOVE KEEPMIGDATE CLOUD
TRANSITION	DATASETNAME
TRANSITIONONLY MIGRATIONONLY BOTH	VOLUME (MIGRATE) STORAGEGROUP
VOLUME	CONVERT TERMINAL TRANSITIONONLY MIGRATIONONLY BOTH MOVE CLOUD

Required parameters of the MIGRATE command for SMS-managed data sets

This topic describes the required parameters of the MIGRATE command.

DATASETNAME: Specifying migration and/or class transition for a specific data set

Explanation: **DATASETNAME(*dsname*)** is the required parameter to migrate and/or perform class transition for a specific data set. For *dsname*, substitute the fully qualified name of the data set you want to migrate. You cannot specify an alias for *dsname*. If you do, DFSMSHsm does not migrate or perform a class transition for the data set.

The MIGRATE command overrides the data set's PRIMARY-DAYS-NON-USAGE and class transition management class attribute. The data set, however, is not migrated if the data set's management class specifically prevents migration.

If you choose class transition, any migration-related parameters are disallowed.

Defaults: None.

Note:

1. The volume where the data set resides must be mounted before you issue the MIGRATE command.
2. A data set that is eligible for migration with the target device of TAPE, that is also eligible for back up, is not migrated, and an error message issued.
3. If a management class cannot be obtained for a data set, the data set is not migrated, and an error message is issued.
4. DFSMSHsm does not handle partitioned data set members individually. If you specify a member name of a partitioned data set, DFSMSHsm processes the entire partitioned data set.
5. If *dsname* is a VSAM data set name, specify the base cluster name. The entire VSAM sphere is migrated. If the sphere has more than one AIX®, more than one path, or more than one path on the AIX, the data set can only be recalled by the base cluster name.

MIGRATIONLEVEL1: Specifying migration of data sets from level 1 to level 2 volumes

Note: Required vs. Optional: The following explanation refers to the MIGRATIONLEVEL1 required parameter. For the explanation of the MIGRATIONLEVEL1 optional parameter, see [“MIGRATIONLEVEL1: Specifying migration of a data set”](#) on page 962.

Explanation: **MIGRATIONLEVEL1** is the required parameter to migrate eligible data sets from all migration level 1 volumes to migration level 2 volumes.

Each migrated SMS data set is processed according to its management class attribute value for LEVEL-1-DAYS-NON-USAGE.

Defaults: None.

Note:

1. If you specify this parameter in a direct-to-tape environment, the command fails.
2. If you specify this parameter when level 1 to level 2 migration of automatic secondary space management is running, the command fails.

VOLUME: Specifying the space management attribute for the eligible data sets on a volume

Explanation: **VOLUME(*volser1*)** is the required parameter to specify migration and/or class transition for eligible data sets on one volume. Information for any migration is printed in the migration activity log.

For *volser1*, substitute the serial number of the level 0 volume from which you want to migrate or delete eligible data sets. Eligibility for processing is determined by each data set's management class attributes.

If the MIGRATE command is specified with only the VOLUME parameter, eligibility for migration and/or class transition is determined by the data set's management class attributes. Data sets will be processed until the low threshold has been met, regardless of whether there are more data sets eligible.

When the MIGRATE command is issued for an SMS-managed volume, that volume is considered DFSMSHsm-managed. DFSMSHsm maintains control records for the volume until a DELVOL command is issued against it.

Defaults: None.

Notes:

- The management class attribute COMMAND-OR-AUTO-MIGRATE=BOTH must be specified for an SMS-managed data set to be eligible for migration by a VOLUME command.
- The COMMAND-OR-AUTO-MIGRATE attribute does not effect Class Transition processing.

STORAGEGROUP: Specifying the space management attribute for the eligible data sets in a storage group

Explanation: **STORAGEGROUP(SGn)** is a required parameter specifying that DFSMSHsm migrate the eligible data sets on the volumes in one or more storage groups. For *SGn*, substitute the name or names of the storage groups that DFSMSHsm will be processing.

- If STORAGEGROUP is specified without *SGn*, the command fails.
- If STORAGEGROUP is specified with more than 30 storage groups, only the first 30 are processed.

When the MIGRATE STORAGEGROUP command is issued, DFSMSHsm will read the SMS storage group definition and determine the list of volumes defined by this storage group. The data sets residing on these volumes will then be processed in a similar way to Primary Space Management (PSM) processing:

- Phase 1:
 1. Issue a message for unmatched data sets.
 2. Delete temporary data sets.
 3. Delete expired data sets.
 4. Determine data sets that are eligible for migration.
 5. Exclude from migration to tape those data sets that are eligible to be backed up.
 6. Determine data sets that are eligible for extent reduction.
 7. Release unused space for eligible data sets
 8. Perform fast subsequent migration for eligible data sets.
 9. Determine data sets that are eligible for class transitions.
 10. Perform class transitions with management class changes only.
- Phase 2:
 1. Transition data sets that are candidates both for a class transition and extent reduction.
 2. Migrate data sets that are candidates both for migration and for extent reduction.
 3. Migrate data sets that are candidates only for extent reduction. Schedule a recall for each data set.
 4. Transition data sets that are candidates for a class transition in FIFO until the low threshold of occupancy is reached (DAYS(0) parameter is not specified).
 5. Migrate data sets in priority order until the low threshold of occupancy is reached (DAYS(0) parameter is not specified).

Notes:

1. If TRANSITIONONLY is specified, only Steps 9 and 10 of Phase 1 and Step 4 of Phase 2 are performed.
2. The max migration tasks SETSYS command (MAXMIGRATIONTASKS) determines the tasking level used for this function.

Defaults: None.

Optional parameters of the MIGRATE command for SMS-managed data sets

This topic describes the optional parameters of the MIGRATE command.

BOTH: Specifying that transition or migration of data sets should be processed

Explanation: **BOTH** is an optional parameter that you use to migrate SMS-managed data sets from a level 0 volume to a migration level 1 or migration level 2 volume, or transition to another level 0 volume. Data sets eligible for migration will be migrated, while data sets eligible for transition only will be transitioned. If a data set is eligible for both migration and transition, it will be migrated.

This parameter can be specified only with MIGRATE VOLUME or MIGRATE STORAGEGROUP command.

Defaults: None.

CONVERT: Specifying movement of data sets from one volume to another

Explanation: **CONVERT** is an optional parameter that you use to migrate all data sets, or a specific data set, from a level 0 volume to a migration level 1 volume and then immediately recall those data sets to another level 0 volume. You can use this parameter when you want to remove a volume from the system. The automatic class selection routines select the volume to which the data sets should be recalled.

The management class attribute PRIMARY-DAYS-NON-USAGE is not used for a MIGRATE CONVERT request. All eligible SMS-managed data sets are migrated to a migration level 1 volume. Before an SMS-managed data set is migrated, DFSMSHsm determines whether or not the data is eligible to be expired.

Defaults: Automatic class-selection routines choose the volume and type of unit to which SMS data sets are recalled.

Note:

1. The management class attribute, LEVEL-1-DAYS-NON-USAGE, are not used when processing a MIGRATE CONVERT request. All eligible data sets are migrated to a migration level 1 volume.
2. If you specify the DAYS(*days*) parameter with the CONVERT parameter, the *days* value must be zero. The default is zero, and the only value allowed is zero.

CLOUD: Specifying migration of a data set directly to cloud

Explanation: **CLOUD**(*cloud_network_connection_name*) is an optional parameter that is used to specify that a data set migrates from an SMS-managed volume to the specified cloud storage. The *cloud_network_connection_name* is the name of the SMS network connection construct or the CDA provider file (when OA66197 enabled). When *cloud_network_connection_name* is specified, the data set is migrated to the requested cloud storage, regardless of the data set's management class attributes, with the exception of migration not being allowed.

The CLOUD parameter cannot be specified with the following parameters: MIGRATIONLEVEL1, MIGRATIONLEVEL2, and CONVERT.

Note:

The following DFSMSHsm SETSYS settings do not apply to CLOUD(*cloud_network_connection_name*):

COMPACT, COMPACTPERCENT – COMPACT(ALL)

Migrating a data set to cloud storage ignores any current COMPACT setting. If the data set is being migrated to a cloud object storage that supports transparent cloud tiering compression, then the data

set can be compressed by TCT compression if it is not already compressed, encrypted, or compressed and encrypted.

CONVERSION(REBLOCKTOANY)

Because data movement is not performed by the host, reblocking of a data set cannot be performed.

CONCURRENT

Migrating a data set to Cloud storage will ignore any current setting relating to CONCURRENT.

Defaults: None.

DAYS: Specifying migration eligibility of data sets

Explanation: **DAYS(0)** is an optional parameter supplied for an SMS volume that allows the movement of all data sets for which automatic space management is supported, or transition of all of the data sets eligible for transition processing. The data set's management class attribute value for PRIMARY-DAYS-NON-USAGE need not have been met. The data set's management class attributes values for Time Since Creation, Time Since Last Use and Periodic are used to determine the data set eligibility for class transition processing. For SMS-managed volumes, the value specified for the DAYS subparameter must be zero. If a value other than zero is specified or if no value is specified, the command fails, and a message indicating the error is issued.

Defaults: None.

Notes:

- The management class attribute COMMAND-OR-AUTO-MIGRATE=BOTH must be specified for an SMS-managed data set to be eligible for migration by a VOLUME command.
- The COMMAND-OR-AUTO-MIGRATE attribute does not effect Class Transition processing.
- The DAYS(0) parameter behaves differently based on the keywords specified. Transition processing behaves differently than migration processing:
 - For migration processing, all data sets that are eligible to migrate will be migrated off of the volume, regardless of whether or not they met their Age criteria. Conversely, transition processing still requires the transition criteria to be met. During transition processing, the Low Threshold does not stop processing of data sets on the volume. All data sets on the volume are processed for migration and transition eligibility, even after the low threshold has been met.
 - If MIGRATIONONLY is specified, processing remains unchanged.
 - If BOTH is specified, since all data sets eligible for migration will be migrated and migration takes precedence over transitions, essentially all data sets eligible for migration are migrated. Only data sets that can not migrate but are eligible to be transitioned are transitioned. These data sets must still meet the transition eligibility criteria.
 - If TRANSITIONONLY is specified, all data sets eligible to be transitioned are processed, regardless of the Low Threshold on the volume.

KEEPMIGDATE: Specifying migration of a data set should keep its original migration date, if one exists.

Explanation: **KEEPMIGDATE** is an optional parameter that is used only when STORAGEGROUP(SGn) is specified. This optional parameter is used to specify that DFSMSHsm should use the original migration date (for example, if the data set was migrated, but since recalled), if one exists, upon migration of the data set.

Defaults: None.

Note: KEEPMIGDATE only applies when STORAGEGROUP(SGn) is specified, and as such, only applies to SMS managed data sets.

MIGRATIONLEVEL1: Specifying migration of a data set

Note: Required vs. Optional: The following explanation refers to the MIGRATIONLEVEL1 optional parameter. For the explanation of the MIGRATIONLEVEL1 required parameter, see [“MIGRATIONLEVEL1: Specifying migration of data sets from level 1 to level 2 volumes”](#) on page 958.

Explanation: **MIGRATIONLEVEL1** is an optional parameter when specifying the DATASETNAME(dsname) parameter. When you specify ML1 with dsname, the data set is migrated to the appropriate hierarchical level, as determined by the data set's management class.

Defaults: None.

MIGRATIONLEVEL2: Specifying migration of a data set directly to a level 2 volume

Explanation: **MIGRATIONLEVEL2** is an optional parameter specifying that a specific data set migrate from an SMS volume directly to a migration level 2 volume. You must specify the data set name with the DATASETNAME parameter to cause DFSMSHsm to migrate a specific data set directly to a migration level 2 volume.

An SMS-managed data set residing on an SMS volume or a migrated SMS-managed data set residing on a migration volume is migrated to a migration level 2 volume, regardless of the data set's management class attributes.

Defaults: None.

Note:

1. You must specify the MIGRATIONLEVEL2 parameter when you request that DFSMSHsm migrate a data set on a migration level 1 volume. DFSMSHsm migrates the data set to the type of migration level 2 volume that is consistent with your environment. For example, if you are in a tape migration environment, DFSMSHsm migrates the data set to a tape migration level 2 volume rather than a DASD migration level 2 volume. You specify the type of migration environment with the TAPEMIGRATION parameter of the SETSYS command.
2. The MIGRATIONLEVEL2 parameter is not necessary in a direct-to-tape environment.
3. The MIGRATIONLEVEL2 parameter can be used to override the management class attribute LEVEL-1-DAYS-NON-USAGE.

MIGRATIONONLY: Specifying that only migration of data sets should be processed

Explanation: **MIGRATIONONLY** is an optional parameter that you use to migrate all data sets, or a specific data set, from a level 0 volume to a migration level 1 or migration level 2. Data sets eligible for transition only will not be processed.

This parameter can be specified only with MIGRATE VOLUME or MIGRATE STORAGEGROUP command.

Defaults: None.

MOVE: Specifying that only moving of data sets should be processed

Explanation: **MOVE** is an optional parameter that you use to move all SMS-managed data sets from a level 0 volume to another level 0 volume. This parameter can be specified only with the MIGRATE VOLUME, MIGRATE STORAGEGROUP or MIGRATE DATASETNAME command.

The MOVE parameter invokes the ACS routines to determine the new volume to which a data set should be moved.

Defaults: None.

Note: When moving Db2 objects, DFSMSHsm must be authorized to the Db2 STOP and START commands.

TERMINAL: Requesting that volume space management messages appear at the terminal

Explanation: **TERMINAL** is an optional parameter specifying that all space management messages associated with the space management of a volume be sent to the system console and to the migration activity log.

Defaults: If you do not specify **TERMINAL** when you specify **MIGRATIONLEVEL1** or **VOLUME**, volume space management messages go only to the migration activity log.

Note: **TERMINAL** does not apply when you specify the **DATASETNAME** parameter. If you specify **TERMINAL** when it does not apply, DFSMSHsm ignores it.

TRANSITION: Specifying that only transitions of data set should be processed

Explanation: **TRANSITION** is an optional parameter that you use to transition eligible SMS-managed data sets from a level 0 volume to another level 0 volume.

The optional SMS subparameters of the **TRANSITION** parameter are:

MANAGEMENTCLASS

Specifies the target management class used for class transition processing. Alias: MC

STORAGECLASS

Specifies the target storage class used for class transition processing. Alias: SC

STORAGEGROUP

Specifies the target storage group list used for class transition processing. Up to 15 storage groups can be specified by this parameter. If **STORAGEGROUP** is specified with more than 15 storage groups, only the first 15 are processed. Alias: SG

Notes:

1. If the **TRANSITION** parameter is specified for a non-SMS managed data set, the command will be failed with the RC20 return code and the REAS38 reason code.
2. If the **TRANSITION** keyword for a data set that *cannot* be transitioned, the command will be failed with the RC20 return code and the REAS36 reason code.
3. If at least one SMS parameter is specified, then the specified value or values will be used and the ACS routines will not be called during class transition processing. The omitted values of management class, storage class or storage group will be taken from the data set's SMS construct.

Defaults: None.

TRANSITIONONLY: Specifying that only transition of data sets should be processed

Explanation:

TRANSITIONONLY

is an optional parameter that you use to transition all eligible SMS-managed data sets from a level 0 volume to another level 0 volume. Data sets eligible only for migration are not processed.

This parameter can be specified only with **MIGRATE VOLUME** or **MIGRATE STORAGEGROUP** command.

The optional SMS subparameters of the **TRANSITIONONLY** parameter are:

MANAGEMENTCLASS

Specifies the target management class used for class transition processing. Alias: MC

STORAGECLASS

Specifies the target storage class used for class transition processing. Alias: SC

STORAGEGROUP

Specifies the target storage group list used for class transition processing. Up to 15 storage groups can be specified by this parameter. If STORAGEGROUP is specified with more than 15 storage groups, only the first 15 are processed. Alias: SG

Defaults: None.

Note: If at least one SMS parameter is specified, then the specified value or values are used and the ACS routines are not called during class transition processing. The omitted values of management class, storage class, or storage group are taken from the data set's SMS construct.

Examples of how to code the MIGRATE command for SMS-managed data sets

The following examples present different ways to code the MIGRATE command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Migration of eligible data sets from a specified volume

In this example, migration is requested for all eligible data sets on the specified volume. Eligibility is determined by each data set's management class.

```
MIGRATE VOLUME(SMS003)
```

Removing all data sets eligible to migrate from an SMS-managed level 0 volume

The following command migrates all eligible data sets from the specified volume. Those data sets meeting their management class expiration attribute values are expired.

```
MIGRATE VOLUME(SMS003) CONVERT
```

Performing class transitions for all eligible data sets in a storage group

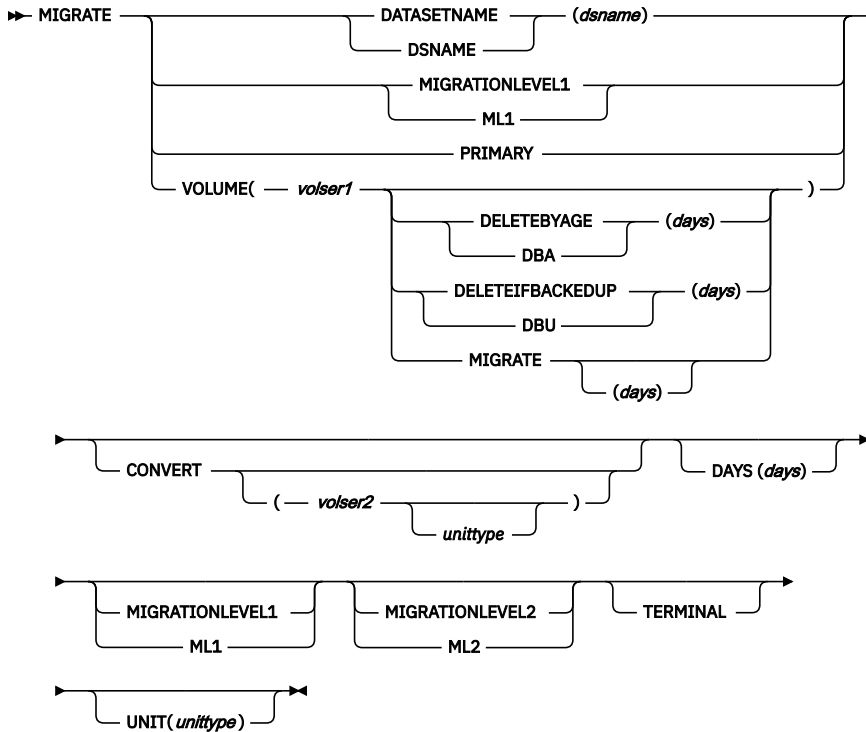
The following command performs class transitions for all eligible data sets in the specified storage group. The TRANSITIONONLY keyword specifies that only class transitions are performed, and no migrations.

```
MIGRATE STORAGEGROUP(STG004) TRANSITIONONLY
```

Requesting a space management function for non-SMS-managed data sets

Use the MIGRATE command to request that DFSMSHsm migrate eligible non-SMS data sets from ML1 volumes, primary volumes, or a specific volume. You can also use this command to migrate a specific data set.

Syntax of the MIGRATE command for non-SMS-managed data sets



Note: You must specify either DATASETNAME, PRIMARY, MIGRATIONLEVEL1, or VOLUME with each MIGRATE command.

Summary of parameters

The following table is a summary of the combination of parameters you can specify with the MIGRATE command. Specifying other combinations of parameters results in either an error message or one or more of the parameters being ignored.

Parameter	Related Parameters
CONVERT	UNIT VOLUME (MIGRATE)
DATASETNAME	CONVERT MIGRATIONLEVEL2
DAYS	MIGRATIONLEVEL1 VOLUME MIGRATE
MIGRATIONLEVEL1	DAYS TERMINAL
MIGRATIONLEVEL2	DATASETNAME
PRIMARY	TERMINAL
VOLUME	CONVERT TERMINAL UNIT

Required parameters of the MIGRATE command for non-SMS-managed data sets

This topic describes the required parameters of the MIGRATE command.

DATASETNAME: Specifying migration of a specific data set

Explanation: **DATASETNAME(*dsname*)** is the required parameter to migrate a specific cataloged data set. When you specify this parameter, DFSMSHsm migrates the cataloged data set without checking whether the data set has met the requirements for migration (such as minimum migration age). For *dsname*, substitute the fully qualified name of the data set you want to migrate. You cannot specify an alias for *dsname*. If you do, DFSMSHsm does not migrate the data set.

Defaults: None.

Note:

1. DFSMSHsm does not handle partitioned data set members individually. If you specify a member name of a partitioned data set, DFSMSHsm processes the entire partitioned data set.
2. The volume where the data set resides must be mounted before you issue the MIGRATE command.

MIGRATIONLEVEL1: Specifying migration of data sets from level 1 to level 2 volumes

Note: Required vs. Optional: The following explanation refers to the MIGRATIONLEVEL1 required parameter. For the explanation of the MIGRATIONLEVEL1 optional parameter see [“MIGRATIONLEVEL1: Specifying migration of a data set directly to appropriate level of hierarchy” on page 970.](#)

Explanation: **MIGRATIONLEVEL1** is the required parameter to migrate eligible data sets from all migration level 1 volumes to migration level 2 volumes.

When you specify MIGRATIONLEVEL1 for non-SMS-managed data sets, two conditions apply:

- If you also specify DAYS(*days*), DFSMSHsm migrates non-SMS-managed data sets based on the *days* value.
- If you do not specify DAYS(*days*), DFSMSHsm migrates non-SMS-managed data sets based on the current DFSMSHsm value for the MIGRATIONLEVEL1DAYS parameter of the SETSYS command.

Defaults: None.

Note:

1. If you specify this parameter in a direct-to-tape environment, the command fails.
2. If you specify this parameter when level 1 to level 2 migration of automatic secondary space management is run, the command fails.

PRIMARY: Specifying space management of eligible data sets on all primary volumes

Explanation: **PRIMARY** is the required parameter to migrate or delete eligible data sets on all primary volumes, depending on the space management technique for each volume. You specify the space management technique with the ADDVOL command. When you specify this parameter, DFSMSHsm processes all primary volumes, not just primary volumes with the primary volume attribute of automatic space management. You specify the primary volume attributes with the ADDVOL command.

Defaults: None.

VOLUME: Specifying the space management attribute for the eligible data sets on a volume

Explanation: **VOLUME(*volser1*)** is the required parameter to specify migration, data set deletion, or data set retirement for eligible data sets on one volume. Eligible data sets on a volume not managed by DFSMSHsm or SMS can migrate or be scratched if you specify the UNIT parameter. Information for any volume migration or data set deletion is printed in the migration activity log. Eligible data sets on a volume can be scratched after they have been inactive for the number of days you specified.

For *volser1*, substitute the serial number of the level 0 volume from which you want to migrate or delete eligible data sets.

The MIGRATE command fails if the command is specified with a single migration level 1 or migration level 2 volume and the MIGRATE optional parameter is either specified on the command or is the default. This function is allowed only by using the FREEVOL command.

Subparameter	Explanation
DELETEBYAGE (<i>days</i>)	The data sets on the volume are scratched if they have been inactive for the number of days you specified. If you specify DELETEBYAGE, the eligible data sets are scratched even if they do not have backup versions; however, if the expiration date of the data set has not been reached, DFSMSHsm does not delete the data set. For <i>days</i> , substitute a decimal number from 0 to 999. For example, if you substitute 10 for <i>days</i> , DFSMSHsm scratches those data sets that have not been referred to for 10 days.
DELETEIFBACKEDUP (<i>days</i>)	<p>The data sets on the volume are scratched if they have been inactive for the number of days you specified. The data set, however, must have a current backup version before DFSMSHsm can scratch it from the volume. A backup version is current if it was created on a date after the data set was last referenced. For <i>days</i>, substitute a decimal number from 1 to 999. For example, if you substitute 25 for <i>days</i>, DFSMSHsm scratches those data sets that have not been referred to for at least 25 days and that have current backup versions.</p> <p>DFSMSHsm can do data set deletion or data set retirement on any primary or migration volume. The migration level 2 volume can be tape or DASD.</p>

Subparameter	Explanation
MIGRATE	<p>All eligible data sets migrate from the specified level 0 volume to a migration level 1 volume or tape migration level 2 volume. The migration level used is determined by the SETSYS TAPEMIGRATION command. For <i>days</i>, substitute a decimal number from 0 to 999. For example, if you substitute a 5, DFSMSHsm migrates those data sets that have not been referred to for 5 days. This value is called the minimum migration age.</p> <p>When you request migration from a primary volume, the following conditions apply:</p> <ul style="list-style-type: none"> • If you specify the MIGRATE(<i>days</i>) subparameter, DFSMSHsm migrates the data sets based on the minimum migration age you specify with <i>days</i>. DFSMSHsm ignores any thresholds of occupancy for the volume. • If you do not specify the MIGRATE(<i>days</i>) subparameter, two conditions apply: <ul style="list-style-type: none"> – If you defined thresholds of occupancy with the ADDVOL command, DFSMSHsm migrates the data sets (starting with the oldest) until the low threshold of occupancy is reached or until DFSMSHsm finds a data set whose inactive age is less than the minimum migration age. You specified this minimum migration age with the ADDVOL or SETSYS command. – If you did not define thresholds of occupancy, DFSMSHsm migrates the data sets based on the minimum migration age specified with the MIGRATE(<i>days</i>) parameter of the ADDVOL command. If you did not specify the minimum migration age with the ADDVOL command, DFSMSHsm uses the current minimum migration age from the DAYS parameter of the SETSYS command.

Defaults: If you do not specify one of the MIGRATE, DELETEIFBACKEDUP, or DELETEBYAGE subparameters, DFSMSHsm uses the space management technique you specified in the ADDVOL command.

Note:

1. If you specify a volume that you have not added to DFSMSHsm with the ADDVOL command, you must also specify the UNIT parameter.
2. If you specify a tape migration level 2 volume and the MIGRATE subparameter, the command fails. To move data sets from a tape migration level 2 volume to another tape migration level 2 volume, use the RECYCLE command.
3. If the MIGRATE subparameter of the VOLUME parameter is specified on the MIGRATE command for a DASD migration level 1 volume, the command will fail, and an error message is issued. The migrate space management function for a DFSMSHsm migration level 1 DASD volume is supported only by using the FREEVOL command.

Optional parameters of the MIGRATE command for non-SMS-managed data sets

This topic describes the optional parameters of the MIGRATE command.

CONVERT: Specifying movement of data sets from one volume to another

Explanation: **CONVERT** is an optional parameter that you use to migrate all data sets, only aged data sets, or a specific data set from a level 0 volume to a migration level 1 volume and then immediately recall

those data sets to another level 0 volume. You can use this parameter when you want to remove a volume from the system. For example, you can use this parameter to move data sets from a 3380 volume to a 3390 volume.

For *volser2*, specify the volume where DFSMSHsm is to recall the data set. For *unittype2*, specify the type of unit where DFSMSHsm is to mount the target volume. If you do not specify *volser2*, DFSMSHsm selects the target volume the same way it normally selects target volumes during recall. If you use the *volser2* subparameter to specify a volume not managed by DFSMSHsm, you must also specify the *unittype2* subparameter.

In addition, when you specify MIGRATE(0), DFSMSHsm can process types of data sets that it does not normally process during migration. These are:

- Cataloged list and utility data sets that cannot be scratched because they have been referred to too recently.
- Data sets that have temporarily been marked as not able to be migrated because a job in a JES3 system planned to use them.

Note: In a JES3 system, two jobs could use the same data set at the same time if you submit one job before the conversion is done and you submit the second job after the conversion. Therefore, ensure that no jobs are waiting to run if those jobs use the data sets on the volume being converted.

Defaults: For cataloged data sets, you do not have to specify *volser2* or *unittype2*. If you do not specify these values, DFSMSHsm chooses the volume and type of unit.

Note:

1. The space management technique must be migration. You can specify the MIGRATE subparameter of the VOLUME parameter, the MIGRATE parameter of the ADDVOL command, or use the default of migration on the ADDVOL command.
2. To prevent the source volume from being selected as the target volume for recall, specify the NOAUTORECALL parameter of the ADDVOL command for the source volume until the migration finishes.
3. If you want DFSMSHsm to convert all data sets on a level 0 volume, you must also specify the MIGRATE(0) subparameter of the MIGRATE command. If, however, you used the SETMIG command to specify the DATASETNAME, LEVEL, or VOLUME parameter with the NOMIGRATION parameter, DFSMSHsm will not migrate the specified entry when you specify the CONVERT parameter even if you also specified the MIGRATE(0) parameter. For example, if you specified SETMIG LEVEL(USER) NOMIGRATION, those data sets with the first qualifier of USER do not migrate from the primary volume.
4. If you specify the CONVERT parameter in a direct-to-tape environment, the command fails.
5. You can use the CONVERT parameter with the UNIT parameter if you want to convert non-SMS-managed volumes not managed by DFSMSHsm to volumes managed by DFSMSHsm.
6. VSAM data sets created on the same day that the MIGRATE VOLUME (MIGRATE(0)) CONVERT command was issued are not migrated because of a data integrity exposure.

DAYS: Specifying migration eligibility of data sets

Explanation: **DAYS(days)** is an optional parameter specifying how many consecutive days that data sets on migration level 1 volumes are to remain inactive before they are eligible for migration. For *days*, substitute a decimal number from 0 to 999.

Defaults: If you do not specify DAYS(days), DFSMSHsm migrates the data sets from the volume based on the current DFSMSHsm value for the MIGRATIONLEVEL1DAYS parameter of the SETSYS command.

Note: The DAYS(days) parameter can be used for non-SMS level 0 volumes and has the same meaning as if it was specified using the VOLUME(volser MIGRATE(days)) method. See the VOLUME parameter for a description of the parameter. If both DAYS(days) and VOLUME(volser MIGRATE(days)) are specified, the VOLUME(volser MIGRATE(days)) value is used.

MIGRATIONLEVEL1: Specifying migration of a data set directly to appropriate level of hierarchy

Note: Required vs. Optional: The following explanation refers to the MIGRATIONLEVEL1 optional parameter. For the explanation of the MIGRATIONLEVEL1 required parameter, see [“MIGRATIONLEVEL1: Specifying migration of data sets from level 1 to level 2 volumes”](#) on page 966.

Explanation: **MIGRATIONLEVEL1** is an optional parameter that can be used with the DATASETNAME(dsname) parameter. When you specify ML1 with dsname, the data set will be migrated to the appropriate hierarchical level, as determined by the SETSYS TAPEMIGRATION parameter.

Defaults: None.

MIGRATIONLEVEL2: Specifying migration of a data set directly to a level 2 volume

Explanation: **MIGRATIONLEVEL2** is an optional parameter specifying that a specific data set migrate from a primary volume directly to a migration level 2 volume. You must specify the data set name with the DATASETNAME parameter to cause DFSMSHsm to migrate a specific data set.

Defaults: None.

Note:

1. You must specify the MIGRATIONLEVEL2 parameter when you request that DFSMSHsm migrate a data set on a migration level 1 volume. DFSMSHsm migrates the data set to the type of migration level 2 volume that is consistent with your environment. For example, if you are in a tape migration environment, DFSMSHsm migrates the data set to a tape migration level 2 volume rather than to a DASD migration level 2 volume. You specify the type of migration environment with the TAPEMIGRATION parameter of the SETSYS command.
2. The MIGRATIONLEVEL2 parameter is not necessary in a direct-to-tape environment.

TERMINAL: Requesting that volume space management messages appear at the terminal

Explanation: **TERMINAL** is an optional parameter specifying that all space management messages associated with the space management of a volume be sent to the system console and to the migration activity log.

Defaults: If you do not specify TERMINAL when you specify MIGRATIONLEVEL1, PRIMARY, or VOLUME, volume space management messages go only to the migration activity log.

Note: TERMINAL does not apply when you specify the DATASETNAME parameter. If you specify TERMINAL when it does not apply, DFSMSHsm ignores it.

UNIT: Specifying the type of device

Explanation: **UNIT(unittype)** is an optional parameter specifying the type of unit where DFSMSHsm can allocate the nonmanaged source volume.

For *unittype*, substitute the type of unit where DFSMSHsm can allocate the volume.

The following are valid types of DASD units:

- 3380
- 3390
- 9345

Defaults: None.

Note:

1. You must specify the UNIT parameter if the volume you specify is not managed by DFSMSHsm.
2. DFSMSHsm ignores the UNIT parameter if you specify the MIGRATIONLEVEL1 or DATASETNAME parameters.

Examples of how to code the MIGRATE command for non-SMS-managed data sets

The following examples show different ways to code the MIGRATE command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Migration of all the data sets from a volume

In this example, migration is requested for all the data sets on the primary volume.

```
MIGRATE VOLUME( PRIM03 MIGRATE(0) )
```

The following command also migrates all data sets from a primary volume:

```
MIGRATE VOLUME( PRIM01 ) DAYS(0)
```

Migration of data sets based on the number of elapsed days since the data set was last referred to

In this example, migration is requested for data sets that reside on the primary volume and have not been referred to for at least five days.

```
MIGRATE VOLUME( PRIM01 MIGRATE(5) )
```

Direct migration of a data set to a level 2 volume

In this example, the specified data set migrates directly to a migration level 2 volume.

```
MIGRATE DATASETNAME( FET7603.JOB33.LOAD ) MIGRATIONLEVEL2
```

Migration of data sets from a volume not managed by DFSMSHsm

In this example, migration is attempted for all the data sets that meet the DAYS criterion of the SETSYS command and reside on a volume not managed by DFSMSHsm.

```
MIGRATE VOLUME( MSS004 ) UNIT(3390)
```

Data set deletion of eligible data sets on a volume

In this example, all eligible data sets on the primary volume that have not been referred to for at least 20 days and whose expiration dates have been reached are scratched.

```
MIGRATE VOLUME( VOL005 DELETEBYAGE(20) )
```

Moving all data sets from a volume not managed by DFSMSHsm to primary volumes

In this example, all DFSMSHsm-supported data sets on a volume not managed by DFSMSHsm move to primary volumes that allow automatic recall.


```
MIGRATE VOLUME(VOL081 MIG(0)) UNIT(3390) CONVERT
```

Moving all data sets from a primary volume

In this example, you want to remove all data sets from a primary volume.

```
MIGRATE VOLUME(ONVOL MIG(0)) CONVERT
```

Moving data sets from a primary volume to another primary volume

In this example, the volume was added to control of DFSMSHsm with the space management technique of MIGRATE. The target volume is a volume managed by DFSMSHsm. All supported data sets that have not been referred to in the last five days are moved from the 3380 volume (USER80) to the target 3390 volume (NEW90).

```
MIGRATE VOLUME(USER80 MIGRATE(5)) CONVERT(NEW90)
```

Moving one data set from one primary volume to another primary volume

In this example, one data set is moved from a volume to another primary volume without setting up an IEHMOVE utility job.

```
MIGRATE DATASETNAME(MYDS) CONVERT
```

You could issue this command to cause extent reduction or to move this data set from a full volume to another volume whose free space is in less demand.

Migration of data sets from all migration level 1 volumes

In this example, migration is requested for the data sets on all migration level 1 volumes that have not been referred to for at least two days.

```
MIGRATE MIGRATIONLEVEL1 DAYS(2)
```

Data set deletion of a tape migration level 2 volume

In this example, data set deletion is requested for the data sets on a tape migration level 2 volume that have not been referred to for 30 days and whose expiration dates have been reached.

```
MIGRATE VOLUME(M2TP04 DELETEBYAGE(30))
```

Data set retirement of a migration level 1 volume

In this example, data set retirement is requested for the data sets on a migration level 1 volume that have not been referred to for 45 days and that have a current backup version.

```
MIGRATE VOLUME(ML1003 DELETEIFBACKEDUP(45))
```


Chapter 56. ONLYIF command: Running commands in ARCCMDxx conditionally

The ONLYIF command allows a single command or group of commands contained within a BEGIN ... END block, immediately following the ONLYIF command, to be run conditionally depending on the host specified. An ONLYIF command cannot follow another ONLYIF command and it cannot be nested within a BEGIN ... END block.

The following examples apply to all supported keywords on the ONLYIF statement. This includes HSMHOST, SYSNAME and HMSYM.

Text on the line immediately following an ONLYIF command is run as the conditional command. That is, if the line is blank or commentary, it is considered the command to be run. For example, the comment after the ONLYIF command is considered for processing, not the command (*command1*). The command applies to all hosts.

```
ONLYIF HSMHOST(hostID1)
/* The following command is issued on all hosts. */
command1
```

To avoid this error, use a continuation character to join the comment to the command. For example:

```
ONLYIF HSMHOST(hostID1)
/* The following command is issued on hostID1 only. */ +
command1
```

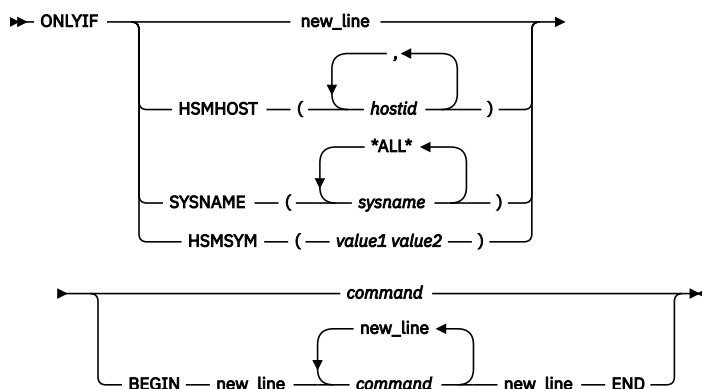
Or, surround multiple lines with BEGIN and END. For example:

```
ONLYIF HSMHOST(hostID1,hostID2)
BEGIN
/* The following commands are issued for hostID1 and hostID2 only. */
command1
command2
END
```

Related reading

For more information about the commands you can specify in the DFSMSHsm SYS1.PARMLIB member ARCCMDxx, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Syntax of the ONLYIF command



Required parameters of the ONLYIF command

This topic describes the required parameter of the ONLYIF command.

Topic

[“HSMHOST: Specifying the host ID of a specific host or hosts” on page 974](#)

[“SYSNAME: Specifying the system name where the next line should be processed” on page 974](#)

[“HMSYM: Specifying the HSM system symbol and its value where the next line should be processed” on page 974](#)

HSMHOST: Specifying the host ID of a specific host or hosts

Explanation

HSMHOST(*hostid*, *hostid*, ...) is a required parameter used to specify the host ID for the DFSMSHsm host on which the command or BEGIN ... END block should be run. The parameter variable is explained:

hostid

Single character host ID of the DFSMSHsm host. The host ID is specified by the HOST keyword in the PROC statement used to start DFSMSHsm. Multiple host IDs are separated by a comma.

Defaults

None.

SYSNAME: Specifying the system name where the next line should be processed

Explanation

SYSNAME is a required parameter used to specify the system name on which the command on the next line, or a BEGIN ... END block, should be run, if the system HSM is initialized on matches this system name. Valid values are 1-8 character names or the reserved string *ALL*. The system name is defined by the SYSNAME parameter in the IEASYMxx or IEASYSxx PARMLIB members.

Defaults

If the keyword is not specified or *ALL* is specified, the next line is processed on all systems.

HMSYM: Specifying the HSM system symbol and its value where the next line should be processed

Explanation

HMSYM is a required parameter used to perform a comparison of a user defined system symbol, *value1*, and its current value to the literal *value2* specification.

value1, *value2*

These values specify the system symbol and the literal value that is checked to match the current value set in the system for this symbol. The command on the next line, or a BEGIN...END block, should be run if the literal value for this keyword matches the system setting for it. The system symbol must be between 1-44 characters long. The values must adhere to standard system symbolic syntax rules for defining static system symbols. The system symbol and its current value is defined by the SYSDEF SYMDEF command of the IEASYMxx parmlib member.

Defaults

None.

Optional parameters of the ONLYIF command

This topic describes the optional parameters of the ONLYIF command.

BEGIN: Begin a group of commands

Explanation

BEGIN is an optional parameter (or operator) that is specified on a new line immediately following the ONLYIF command. All commands following the BEGIN parameter to the next END parameter are conditionally issued as a group.

Defaults

None.

END: End a group of commands

Explanation

END is an optional parameter (or operator) which marks the end of a group of commands started by the BEGIN parameter. The END parameter must be specified on a new line immediately following the last conditional command.

Defaults

None.

Examples of how to code the ONLYIF command

The following examples present how to code the ONLYIF command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

The following examples apply to all supported keywords on the ONLYIF statement. This includes HSMHOST, SYSNAME and HSMSYM.

Specifying a different HSMPLEXNAME in an HSM configuration for a production system and a disaster recovery exercise system that both share the same SYSNAME and the same DFSMSHsm parmlib member

In this example, an HSM system symbol HSMSYS is set to PROD on the production system and DR for the disaster recovery system. These system symbols are defined by statements in the IEASYMxx parmlib. When DFSMSHsm is started on the production system, the HSMSYS symbol resolves to PROD which matches its second literal value of PROD on the first ONLYIF HSMSYM command. In this case, the ONLYIF evaluates to be true and the subsequent statement is executed. Additional statements between the BEGIN/END keywords can be added to conditionally be run. The second ONLYIF HSMSYM evaluates not to be true since it is looking for a value of DR and the subsequent statement, and only that statement, is skipped.


```

ONLYIF HSMSYM(&HSMSYS PROD)
BEGIN
  SETSYS HSMPLEXNAME(PROD)
END
ONLYIF HSMSYM(&HSMSYS DR)
  SETSYS HSMPLEXNAME(DR)

```

Specifying different start and stop times for automatic primary space management for 2 hosts

In this example, DFSMSHsm host HSM1 is configured with a planned start time for automatic primary space management at 4:00 a.m. and the quiesce time is 7:30 a.m. DFSMSHsm host HSM2 is configured with a planned start time for automatic primary space management at 9:00 a.m. and the quiesce time is 11:00 a.m.

```

ONLYIF HSMHOST(HSM1)
  SETSYS PRIMARYSPMGMTSTART(400 730)
ONLYIF HSMHOST(HSM2)
  SETSYS PRIMARYSPMGMTSTART(900 1100)

```

Specifying different maximum number of tasks DFSMSHsm can concurrently process for several hosts

In this example, DFSMSHsm host HSM1 can have a maximum of 7 automatic volume space management tasks that can run concurrently and a maximum of 5 volume backup tasks that can run concurrently. All other hosts can have a maximum of 3 automatic volume space management tasks that can run concurrently and a maximum of 9 volume backup tasks that can run concurrently.

```

SETSYS MAXMIGRATIONTASKS(3) MAXBACKUPTASKS(9)
ONLYIF HSMHOST(HSM1)
  SETSYS MAXMIGRATIONTASKS(7) MAXBACKUPTASKS(5)

```

Specifying continuation characters for commands with several parameters

In this example, all DFSMSHsm hosts (host HSM1) will have the following attributes:

- a maximum of 1 automatic volume space management task.
- when small data sets migrate, they are not packed into predefined VSAM key-sequenced small-data-set-packing data sets on a migration level 1 volume.
- interval migration is not performed.
- a maximum of 2 recall tasks can run concurrently.
- a maximum of 1 tape recall task that can request tape mounts.

Host HSM1 will have these attributes:

- a maximum of 3 automatic volume space management tasks can run concurrently.
- when small data sets migrate, they are eligible to be packed into predefined VSAM key-sequenced small-data-set-packing data sets on a migration level 1 volume.
- interval migration is performed.
- a maximum of 8 recall tasks can run concurrently.
- a maximum of 4 tape recall tasks that can concurrently request tape mounts.

```

SETSYS MAXMIGRATIONTASKS(1) NOSMALLDATASETPACKING NOINTERVALMIG +
  MAXRECALLTASKS(2) TAPEMAXRECALL(1)
ONLYIF HSMHOST(HSM1)
  SETSYS MAXMIGRATIONTASKS(3) SMALLDATASETPACKING INTERVALMIG +
    MAXRECALLTASKS(8) TAPEMAXRECALL(4)

```


Specifying various commands within ARCCMDxx using ONLYIF

In this example, DFSMSHsm host HSM1 runs automatic backup on weekdays (Monday through Friday) and the cycle is started or reset on January 3, 1994 (Monday). Host HSM2 runs automatic backup on Saturday only and the cycle is started or reset on January 3, 1994 (Monday). Host HSM3 prevents all DFSMSHsm functions from running.

```
ONLYIF HSMHOST(HSM1)
  DEFINE BACKUP( YYYYYNN -      /* run backup on weekdays only */
    CYCLESTARTDATE(1994/01/03) /* cycle defined on a Monday */
ONLYIF HSMHOST(HSM2)
  DEFINE BACKUP( NNNNNYN -      /* run backup only on Saturdays */
    CYCLESTARTDATE(1994/01/03) /* cycle defined on a Monday */
ONLYIF HSMHOST(HSM3)
  HOLD ALL                      /* functions held on test system */
```

Specifying that automatic dump only run on three hosts

In this example, the SETSYS and DEFINE commands to set up automatic dump are ran on DFSMSHsm hosts HSM1, HSM2, and HSM3 only.

Note: The indentation of lines between the BEGIN and END parameters is for ease of reading only, and is not required.

```
ONLYIF HSMHOST(HSM1,HSM2,HSM3)
BEGIN
  SETSYS AUTODUMPSTART(0300 0400 0500)
  SETSYS BACKUP(TAPE)
  SETSYS MAXDUMPTASKS(3)
  DEFINE BACKUP(Y)
  DEFINE DUMPCYCLE(Y)
  DEFINE DCLASS(DCLASS1 DATASETRESTORE FREQ0)
END
```

Chapter 57. PATCH command: Modifying storage within DFSMSHsm

The PATCH command is to be used only for maintenance purposes and is explained in the [*z/OS DFSMSHsm Diagnosis*](#).

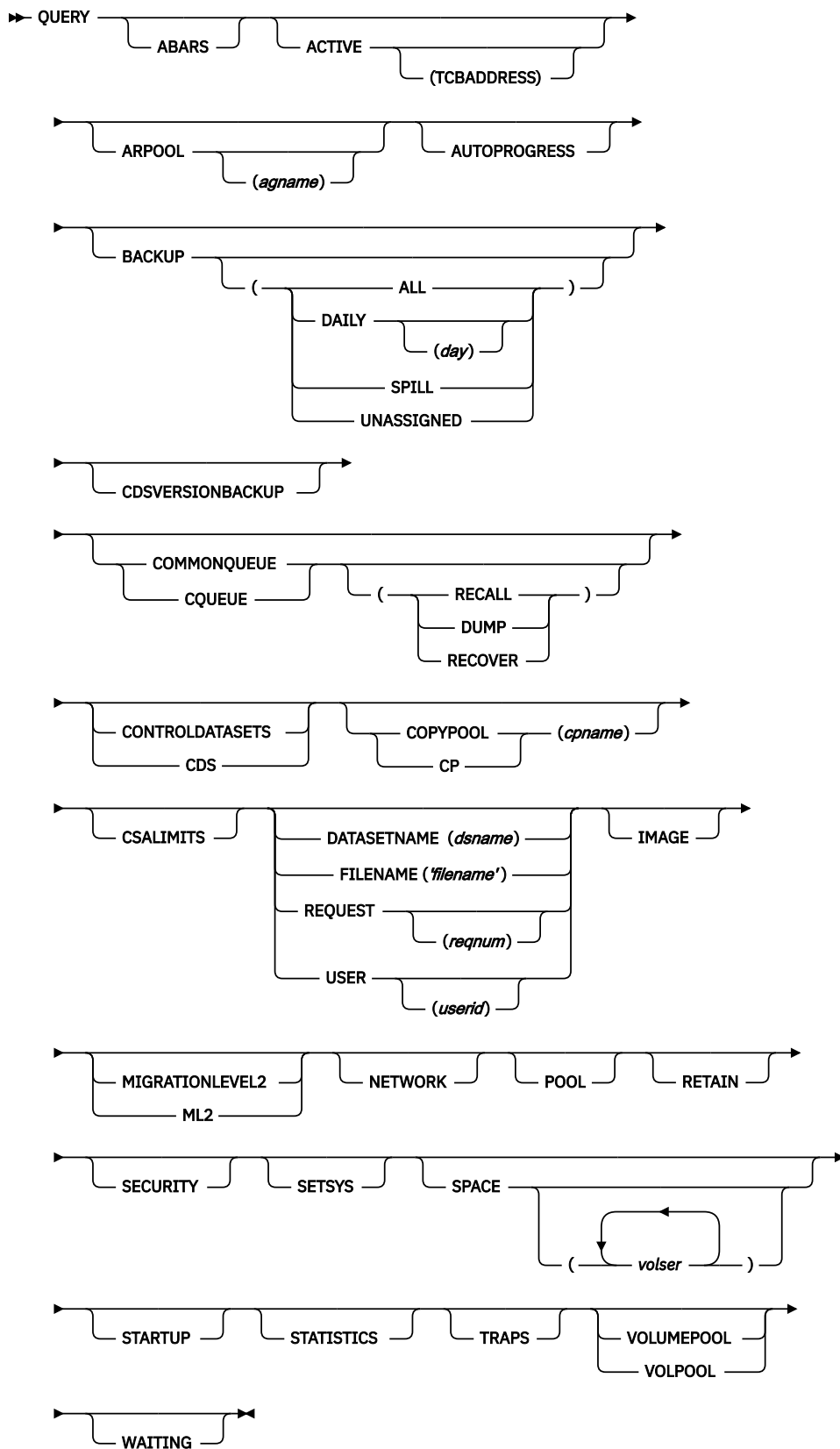
Chapter 58. QUERY command: Displaying the status of parameters, statistics, and requests

The QUERY command can do the following tasks:

- Determine the DFSMSHsm control parameters (SETSYS).
- Determine the DFSMSHsm control parameters that apply to aggregate backup and recovery (ABARS).
- Request a list of DFSMSHsm statistics (STATISTICS).
- Determine the status of pending DFSMSHsm requests (WAITING).
- Request volume space-use information (SPACE).
- Request control data set space-use information (CONTROLDATASETS).
- Determine whether multiple backup data set support is enabled (CDSVERSIONBACKUP).
- Request the common service area storage limits (CSALIMITS).
- Request the status of each DFSMSHsm volume and data set subtask, each long-running command, and what each active one is doing (ACTIVE).
- Request a list of recall pools (POOL).
- Request a list of volumes defined to be used during aggregate recovery processing (ARPOOL).
- Request a list of the backup parameters and backup volumes (BACKUP).
- Request a list of the source volume serial numbers for any volumes in a specified copy pool that have an outstanding FlashCopy relationship (COPYPOOL).
- Request a list of selected DFSMSHsm requests (DATASETNAME|REQUEST|USER).
- Request a list of the DFSMSHsm address spaces that are active on the z/OS image (IMAGE).
- Request a list of key ranges and associated DASD ML2 volumes, and tape ML2 volumes (MIGRATIONLEVEL2).
- Request a list of the first qualifiers of data sets excluded from space management (RETAIN).
- Determine whether access to DFSMSHsm commands is controlled using the RACF FACILITY class or the AUTH command (SECURITY).
- Determine the parameters specified when DFSMSHsm was started (STARTUP).
- Request a list of trap requests specified with the TRAP command (TRAPS).
- Request a list of current volume pools (VOLUMEPOOL).
- Request a list that indicates the number of volumes that have not been processed and the total volumes eligible to be processed. This list can be used to indicate how much work remains to be done for an automatic function currently in progress (AUTOPROGRESS).
- Request percent full information for the common queue (COMMONQUEUE).
- Request percent full, outstanding recall requests, and recall requests that are currently being processed in the common recall queue (COMMONQUEUE(RECALL)).
- Request outstanding and active dump or recover requests that are being processed in a common dump queue or common recover queue.
- Request information concerning cloud endpoints DFSMSHsm knows about (NETWORK).

For more information about the messages associated with the QUERY command, see [Chapter 77, “Using the QUERY command,”](#) on page 1389.

Syntax of the QUERY command



Note:

1. You must specify at least one of the optional parameters to have anything displayed.

2. Any QUERY command issued within the startup procedure may not always produce correct data because DFSMSHsm is not yet completely initialized.
3. The SPACE parameter is not supported for SMS-managed volumes.
4. The QUERY command does not support sending the results of the query to an output data set; however, it does support extended consoles. Using TSO/E extended consoles, you can submit programs that can issue a console command, such as a QUERY, and get the results back for interpretation. DFSMSHsm provides a sample REXX program called QUERYSET in SYS1.SAMPLIB(ARCTOOLS). For more details, refer to [z/OS TSO/E REXX Reference](#).

Optional parameters of the QUERY command

This topic describes the optional parameters of the QUERY command.

ABARS: Displaying aggregate backup and recovery control parameters

Explanation: **ABARS** is an optional parameter requesting a list of the DFSMSHsm control parameters that apply to aggregate backup and recovery. These control parameters are:

- ABARSACTLOGMSGLVL
- ABARSACTLOGTYPE
- ABARSBUFFERS
- ABARSDELETEACTIVITY
- ABARSOPTIMIZE
- ABARSTAPES
- ABARSPROCNAME
- ABARSUNITNAME
- ABARSVOLCOUNT
- ARECOVERML2UNIT
- ARECOVERPERCENTUTILIZED
- ARECOVERTGTGDS
- ARECOVERUNITNAME
- MAXABARSADDRESSSPACE

Defaults: None.

ACTIVE: Displaying all active requests

Explanation: **ACTIVE** is an optional parameter requesting the status of each DFSMSHsm volume, data set subtask, and long-running command, and what each active one is doing.

TCBADDRESS is an optional subparameter of **ACTIVE** requesting only messages directly related to data movement activities be listed, along with extra information allowing cancellation of the active task.

To aid in virtual storage constraint relief (VSCR), the active parameter will display the amount of virtual storage (above and below the 16M line) that is unallocated and available to DFSMSHsm. The two largest free areas are shown to indicate the level of fragmentation.

Defaults: None.

Note: In a common recall, dump, or recover queue environment, only recall requests from the host that is issuing the QUERY ACTIVE command are returned.

ARPOOL: Displaying aggregate recovery pools

Explanation: **ARPOOL** is an optional parameter requesting a list of the aggregate recovery pools as specified with the DEFINE ARPOOL command.

If *agname* is specified, the pool for that *agname* and the associated volume serial numbers are listed. If *agname* is not specified, all pools and the volume serial numbers for each pool are listed. If the DEFINE ARPOOL was not previously issued or was issued with the ALL, LOVOLS(*), or ML1VOLS(*) parameters, then a message is issued indicating this. A list of the currently ADDVOLed ML1 and primary volumes is also displayed.

Defaults: None.

AUTOPROGRESS: Displaying an indication of volumes to be processed

Explanation: **AUTOPROGRESS** is an optional parameter you can specify to request information about the number of volumes to be processed. You can get an indication of the number of volumes that have not been processed by an automatic function and the total number of volumes that are eligible for processing by an automatic function. For each automatic function that is processing DFSMSHsm-managed volumes on the host where the QUERY AUTOPROGRESS command is issued, the following is reported:

- The number of eligible SMS-managed volumes restricted to this DFSMSHsm host that have not been processed. *Restricted to this DFSMSHsm host* means that the storage group with which a volume is associated specifies processing for this function only by the host in which the QUERY AUTOPROGRESS command was issued.
- The total number of eligible SMS-managed volumes restricted to this DFSMSHsm host. *Restricted to this host* means that the storage group with which a volume is associated specifies processing for this function only by the DFSMSHsm host in which the QUERY AUTOPROGRESS command was issued.
- The number of eligible SMS-managed volumes that are not restricted to processing by any DFSMSHsm host and have not been processed by this function.
- The total number of eligible SMS-managed volumes that are not restricted to processing by any DFSMSHsm host.
- The number of eligible non-SMS-managed volumes that have not been processed by this function.
- The total number of non-SMS-managed volumes that are eligible for processing by this function in this DFSMSHsm host.
- The number of copy pools and copy pool volumes that remain to be processed by auto dump.

When QUERY AUTOPROGRESS is used, DFSMSHsm displays the information for each automatic function currently processing DFSMSHsm-managed volumes. If no automatic functions are processing, DFSMSHsm indicates this with message ARC0246I.

Note:

1. The count of volumes that remain to be processed may not reflect another processing unit's activity. A volume that is eligible for processing by multiple-DFSMSHsm-hosts may have been processed in a DFSMSHsm host other than the one where the QUERY AUTOPROGRESS command was issued. If the DFSMSHsm host running the QUERY AUTOPROGRESS command has not detected that another DFSMSHsm host has processed a volume, that volume is included in the count of volumes that have not been processed.
2. A volume is considered *eligible* if it is defined (via the Storage Group for SMS-managed volumes or the ADDVOL command for non-SMS-managed volumes) as being eligible to be processed by this DFSMSHsm host for the active function. For Interval Migration, eligible volumes are only the ones that are above their associated threshold level.
3. A volume is considered *not processed* when both the following conditions are true:
 - The volume is not currently being processed by the active function.
 - The minimum time has elapsed since the last time the associated automatic function processed the volume.

Defaults: None.

BACKUP: Displaying backup parameters and backup volumes

Explanation: **BACKUP** is an optional parameter for requesting a list of the backup and dump parameters and backup volumes.

Subparameter	Explanation
ALL	A list of all nonfull tape backup volumes and all DASD backup volumes is generated.
DAILY	A list of all nonfull daily tape backup volumes and all daily DASD backup volumes is generated. For <i>day</i> , substitute a decimal number from 1 to 31 to list the daily backup volumes associated with the specified day in the backup cycle.
SPILL	A list of all nonfull tape spill backup volumes and all DASD spill backup volumes is generated.
UNASSIGNED	<p>A list of all nonfull unassigned tape backup volumes and all unassigned DASD backup volumes is generated. There are two types of unassigned volumes:</p> <ul style="list-style-type: none"> Unassigned backup volumes are those that you specified with the ADDVOL command without using the DAILY or SPILL subparameter (ADDVOL <i>volser</i> BACKUP). Unassigned daily backup volumes are those that you specified as daily backup volumes with the ADDVOL command, but that you did not associate with any day in the backup cycle (ADDVOL <i>volser</i> BACKUP(DAILY)).

Defaults: If you specify DAILY without *day*, DFSMSHsm lists all daily nonfull tape backup volumes and all daily DASD backup volumes.

Note: Full backup tapes may appear in the list if you have not issued a FIXCDS REFRESH command.

CDSVERSIONBACKUP: Displaying the control data set multiple backup support parameters

Explanation: **CDSVERSIONBACKUP** is an optional parameter requesting a list of the parameters related to the multiple backup copies of the control data sets and journal data set. When you specify this parameter, DFSMSHsm lists the values you specified with the CDSVERSIONBACKUP parameter of the SETSYS command.

Defaults: None.

COMMONQUEUE: Displaying common queue information

Explanation: **COMMONQUEUE** is an optional parameter that you can specify without any subparameters to display percent full information for the common RECALL queue. The common DUMP and RECOVER queue information will not be displayed if no subparameter is specified.

COMMONQUEUE(RECALL) is a subparameter that you can specify to display percent full, outstanding recall requests, the HOST ID from which the recall request was originated, and recall requests that are currently being processed in the common recall queue. When you specify this parameter, DFSMSHsm provides a snapshot of all recall activity across the CRQplex.

COMMONQUEUE(DUMP) is a subparameter that you can specify to display the status of dump requests that are being processed and queued in the common dump queue of which the host issuing the query command is a member. When connected to a DFSMSHsm dump group, this query provides a snapshot of all dump activity across all dump group members.

COMMONQUEUE(RECOVER) is a subparameter that you can specify to display the status of recover requests that are being processed and queued in the recover group queue of which the host issuing the query command is a member. When connected to a DFSMSHsm recover group, this query provides a snapshot of all recover activity across all dump group members.

Defaults: None.

CONTROLDATASETS: Displaying control data set space-use information

Explanation: **CONTROLDATASETS** is an optional parameter requesting a list of the space-use information for the MCDS, BCDS, OCDS, and journal data set. The list consists of the following information:

- The serialization technique being used.
- The total space in each data set.
- The percentage of the total occupied space in each data set to the extent that DFSMSHsm can determine it.
- The threshold percentage value currently assigned for each data set. When this threshold is exceeded, DFSMSHsm sends warning messages to the operator.
- The key boundaries of multicluseter data sets.

Defaults: None.

Note: Information for the JOURNAL data set is based on the last journal update on the host that the QUERY command is entered on. The actual value can be affected by updates made on other hosts.

COPYPOOL: Displaying copy pool information

Explanation: **COPYPOOL**(*cpname*) is an optional parameter that you specify to list which source volumes in the specified copy pool have an outstanding FlashCopy relationship.

You can use this command to determine if another backup version can be created or if a recover can be performed.

Defaults: None.

CSALIMITS: Displaying the common service area storage limits

Explanation: **CSALIMITS** is an optional parameter requesting a list of the common service area (CSA) storage limits. You specify the CSA storage limits with the SETSYS command.

Defaults: None.

DATASETNAME, FILENAME, REQUEST, and USER: Displaying selected requests

Explanation: **DATASETNAME**(*dsname*), **FILENAME**(*filename*), **REQUEST**, and **USER** are mutually exclusive, optional parameters that request a list of selected DFSMSHsm requests.

DATASETNAME(*dsname*) specifies a list of the requests that are associated with the specified data set name. For *dsname*, substitute the fully qualified name of the data set for which you want the list of requests.

FILENAME(*filename*) specifies a list of requests that are associated with the specified UNIX file name enclosed in single quotation marks. For *filename*, substitute the absolute path and UNIX file name for the file for which you want the list of requests. *filename* must be enclosed in single quotation marks.

REQUEST(*reqnum*) specifies a list of the specified request number for all requests. For *reqnum*, substitute the DFSMSHsm request number.

USER(*userid*) specifies a list of the requests that are associated with a specified user. For *userid*, substitute a 1-8 alphanumeric character string for the identification of the particular user.

Defaults: If you specify USER without *userid*, DFSMSHsm lists requests for all users. If you specify REQUEST without *reqnum*, DFSMSHsm lists information for all requests.

Note:

1. DFSMSHsm does not handle partitioned data set members individually. If you specify a partitioned data set with a member name, DFSMSHsm displays the entire partitioned data set.
2. In a common recall, dump, or recover queue environment, recall requests from only the host that is issuing the QUERY DATASETNAME|REQUEST|USER command are returned.

IMAGE: Displaying information for the DFSMSHsm address spaces active within a single z/OS image

Explanation: **IMAGE** is an optional parameter requesting information on the DFSMSHsm address spaces that are active within a single z/OS image. Information displayed includes the host identifier, procedure name used to start the address space, the job identifier, the address space identifier, and the host mode (MAIN or AUX). The information is displayed in message ARC0250I.

Defaults: None.

MIGRATIONLEVEL2: Displaying key ranges and migration level 2 volumes

Explanation: **MIGRATIONLEVEL2** is an optional parameter requesting a list of key ranges and the DASD migration level 2 volume associated with each key range as specified with the DEFINE command. This parameter also displays the tape migration level 2 volumes currently selected for use by the migration functions that output to tape if migration tapes are used in the requesting DFSMSHsm host.

Defaults: None.

NETWORK: Displaying cloud endpoint statistics known to DFSMSHsm

Explanation: **NETWORK** is an optional parameter requesting DFSMSHsm to output informational statistical data about known cloud endpoints.

Defaults: None.

Note: DFSMSHsm does not display endpoints information for S3 cloud providers.

POOL: Displaying recall pools

Explanation: **POOL** is an optional parameter requesting a list of the recall pools as specified with the DEFINE command.

Defaults: None.

Note: An SMS-managed volume may be displayed on the output listing if the volume was converted from non-SMS-managed to SMS-managed since the last time the pool was defined.

RETAIN: Displaying data set levels prevented from space management

Explanation: **RETAIN** is an optional parameter requesting a list of the first qualifiers of data sets prevented from space management with the SETMIG command. Data set names and volumes prevented from migrating with the DATASETNAME or VOLUME parameters of the SETMIG command are not listed.

Defaults: None.

SECURITY: Displaying the type of command authorization in effect

Explanation: **SECURITY** is an optional parameter that you specify to request information about the type of command authorization in effect. When you specify this command, DFSMSHsm issues a message that indicates whether RACF profiles or the DFSMSHsm AUTH command protects DFSMSHsm commands.

Defaults: None.

SETSYS: Displaying control parameters

Explanation: **SETSYS** is an optional parameter requesting a list of the DFSMSHsm control parameters. You can change most of these parameters with the SETSYS command; some can only be changed at the startup of DFSMSHsm.

Defaults: None.

SPACE: Displaying volume space-use information

Explanation: **SPACE** is an optional parameter requesting a list of information about space used on the specified non-SMS-managed primary volumes and migration level 1 volumes. For *volser* ..., substitute the serial numbers of the volumes whose space-use information you want listed.

Defaults: If you specify SPACE without *volser*, DFSMSHsm lists space-use information for all primary non-SMS-managed volumes and migration level 1 volumes.

STARTUP: Displaying the startup information

Explanation: **STARTUP** is an optional parameter requesting a list of the parameters that were specified when DFSMSHsm was started. These parameters were specified on the PROC statement in the DFSMSHsm startup procedure.

Defaults: None.

STATISTICS: Displaying daily statistics

Explanation: **STATISTICS** is an optional parameter requesting a list of selected DFSMSHsm daily statistics that have accumulated for the current day.

Defaults: None.

TRAPS: Displaying trap requests

Explanation: **TRAPS** is an optional parameter requesting a list of trap requests that were specified with the TRAP command.

Defaults: None.

VOLUMEPOOL: Displaying current volume pools

Explanation: **VOLUMEPOOL** is an optional parameter requesting a list of the current volume pools that you have specified with the DEFINE command. The current volume pool names and the volumes that make up the volume pools are listed.

Defaults: None.

WAITING: Displaying the number of requests waiting to be processed

Explanation: **WAITING** is an optional parameter requesting a list of the number of requests on each functional queue waiting for DFSMSHsm to process them.

Defaults: None.

Note: In a common recall, dump, or recover queue environment, only recall requests from the host that is issuing the QUERY WAITING command are returned.

Examples of how to code the QUERY command

For a list of the messages generated by the QUERY command, see [Chapter 77, “Using the QUERY command,” on page 1389](#).

Chapter 59. RECALL command: Recalling a specific migrated data set

The RECALL command recalls a migrated data set. If you want, you can specify the volume and type of unit where DFSMSHsm is to recall the data set.

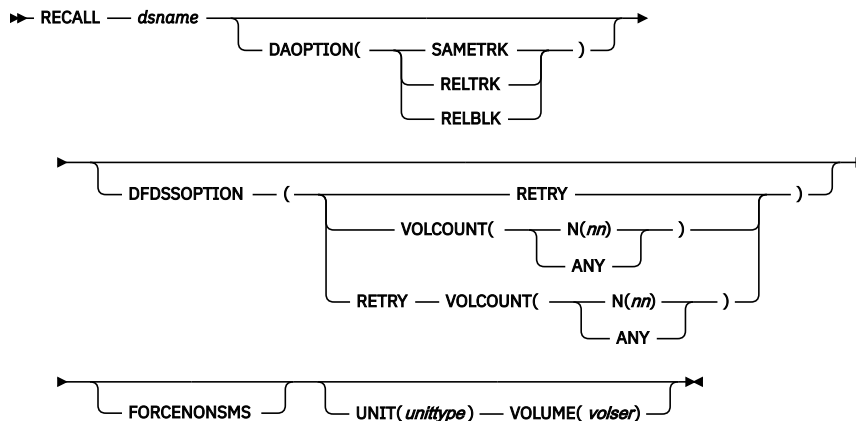
If you specify the volume and type of unit when recalling an SMS-managed data set, the volume and unit type will be passed to the automatic class selection (ACS) Routines. ACS ultimately determines to which volume the data set is recalled.

The FORCENONSMS parameter allows the ability to force recall of an SMS-managed data set to a non-SMS-managed volume. It is not possible to convert a data set that is *not* extended format to a data set that is extended format during a recall. However, you can convert an extended format data set to a data set that is *not* extended format during a recall, if you specify the FORCENONSMS optional parameter.

If you do not specify the target volume for a recalled non-SMS-managed data set, and the data set migrated from a primary volume whose primary volume attribute was automatic backup or automatic migration, DFSMSHsm automatically tries to recall it to a primary volume having the same set of attributes as the volume it migrated from. If no volume is found and the UNLIKE subparameter was specified with the SETSYS RECALL command, then DFSMSHsm can recall to a volume that has unlike attributes.

If a host is connected to a common recall queue (CRQ), recall requests are sent to the CRQ. For information about the common recall queue functions, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

Syntax of the RECALL command



Note: You can specify RETRY, VOLCOUNT, or both with the DFDSSOPTION, but you must specify at least one parameter.

Required parameters of the RECALL command

This topic describes the required parameter of the RECALL command.

dsname: Specifying the name of the data set to be recalled

Explanation: *dsname* is a required positional parameter specifying the data set that you want to recall. For *dsname*, substitute the fully qualified name of the data set.

Defaults: None.

Note:

1. Because *dsname* is a required positional parameter, you must specify it immediately after the command name.
2. You cannot specify an alias for *dsname*. If you do, DFSMSHsm does not recall the data set.
3. DFSMSHsm does not handle partitioned data set members individually. If you specify a member name of a partitioned data set, DFSMSHsm processes the entire partitioned data set. If *dsname* is a VSAM data set name, specify the base cluster name, any component name, or any path name. If the sphere has more than one AIX, more than one path, or more than one path to the AIX, only the base cluster name can be specified for RECALL. The entire VSAM sphere is recalled.

Optional parameters of the RECALL command

This topic describes the optional parameters of the RECALL command.

DAOPTION: Selecting target volume track length

Explanation: **DAOPTION(SAMETRK | RELTRK | RELBLK)** are mutually exclusive, optional parameters specifying the type of data set accessing required (relative track or relative block) upon recall of a direct access data set. This, in turn, implies the allowable target volume device types.

Subparameter	Explanation
SAMETRK	If DFSMSHsm is not directed to a volume by the VOLUME parameter, it selects a target volume having the same track length as the last L0 volume from which the data set was migrated. If DFSMSHsm is directed to a specific volume by the VOLUME parameter, this volume must have the same track length as the last L0 volume from which the data set was migrated. Data will be moved as a track-to-track image, accessible for both relative track and relative block processing.
RELTRK	If DFSMSHsm is not directed to a volume by the VOLUME parameter, it attempts to select a target volume having the same track length as the last L0 volume from which the data set was migrated. If no volume of equal track length is available, the target volume selected may have a larger track length than the last L0 volume. If DFSMSHsm is directed to a specific volume by the VOLUME parameter, this volume must have the same or greater track length than the last L0 volume from which the data set was migrated. Data is moved as a track-to-track image, accessible by relative track addressing.
RELBLK	If DFSMSHsm is not directed to a volume by the VOLUME parameter, it attempts to select a target volume having the same track length as the last L0 volume from which the data set was migrated. If no volume of equal track length is available, the target volume selected may have a larger or smaller track length than the last L0 volume. If DFSMSHsm is directed to a specific volume by the VOLUME parameter, this volume can have any track length. Data will be moved to fill out the track, accessible by relative block addressing.

Defaults: If DAOPTION is not specified, the target volume selection is unchanged and data is moved as a track image, allowing for relative track addressing.

Note: Only basic direct access method (BDAM) data sets are supported by this option. Using DAOPTION to recall a data set as SMS-managed is not supported. If a data set would be SMS-managed after the recall, the FORCENONSMS parameter of RECALL must be used to force it to be non-SMS-managed.

DFDSSOPTION: Pass optional parameters to DFSMSdss

Explanation: **DFDSSOPTION(RETRY | VOLCOUNT)** is an optional parameter that causes DFSMSHsm to pass optional parameters to DFSMSdss when recalling certain types of data sets or to specify the number of volumes a data set can extend to.

RETRY is applicable to the recalling of the following types of data sets and is ignored by DFSMSHsm if specified for other types of data sets:

- Partitioned data set: When the initial recall of the data set fails and a DFSMSdss message requests the use of the NOPACKING option. RETRY specifies for DFSMSHsm to pass this NOPACKING option to DFSMSdss.
- Keyed VSAM data set: When the initial recall of the data set fails and a DFSMSdss message requests the use of the FREESPACE(0 0) option, to retry the restore to a like device, or both. If the data set is not SMS-managed, also specify VOLUME(volser) UNIT(unittype) parameter on the recall command. RETRY specifies for DFSMSHsm to pass this FREESPACE option to DFSMSdss.

Note: If the RETRY parameter is specified, then the ACS routine that indicates a RECALL environment will always be invoked, even if one or more class transitions were missed while the data set was migrated or if the data set was transitioned before migration.

VOLCOUNT(ANY | N(nn)) allows an SMS data set to extend to additional volumes within the same storage group, when space is available. The VOLCOUNT parameter is limited to SMS data set types supported as multivolume data sets. Refer to *z/OS DFSMSdss Storage Administration* for more information regarding restrictions and limitations of the VOLCOUNT parameter.

VOLCOUNT(ANY) results in the allocation of the target data set on as many volumes as required, to a maximum of 59.

VOLCOUNT(N(nn)) allows users to specify the number of volumes to be used for SMS data set allocation. nn can be any value from 0 to 59. When you specify the number of volumes the data set is allocated to, the allocation of the data set will be limited to that number of volumes or will be extended to that number of volumes.

The VOLCOUNT(N(nn)) parameter is ignored if specified with the DFSMSHsm VOLUME parameter. If a volume count VOLCOUNT(N(nn)) is specified and the VOLUME parameter is also specified, the resulting volume count for the data set will be one (1). If inadequate space exists on the specified volume, the allocation fails.

Defaults: None.

Note: If you specify the RECALL command with the VOLCOUNT parameter, it overrides any option used on the SETSYS VOLCOUNT command.

FORCENONSMS: Recalling an SMS-managed data set to a non-SMS-managed volume

Explanation: **FORCENONSMS** is an optional parameter that forces a data set to be recalled to a non-SMS-managed volume. Use this parameter if SMS is not active or if SMS is active and you need to return a data set to non-SMS-managed status.

Defaults: None.

Note: If SMS is not active, a recall request for an SMS-managed data set will fail unless the FORCENONSMS parameter is specified on the RECALL command.

UNIT: Specifying the type of device

Explanation: **UNIT(unittype)** is an optional parameter specifying the type of unit to which DFSMSHsm can allocate the target volume. A target volume is the volume that receives the recalled data set. For *unittype*, substitute the type of unit to which DFSMSHsm can allocate the target volume.

The following are valid types of DASD units:

- 3380
- 3390
- 9345

Defaults: None.

Note:

1. If you specify UNIT, you must also specify the VOLUME parameter.
2. If the block size of the data set is greater than the track capacity of the target volume, track overflow must be supported in both software and hardware. This applies to devices whose track size is less than 32K.
3. When you specify the UNIT parameter, DFSMSHsm passes that unit name to the automatic class selection (ACS) routines to determine whether the data set should be recalled to an SMS-managed volume. See [Chapter 2, “Space management of SMS-managed storage,” on page 23](#) for more information about automatic class selection.

VOLUME: Specifying the volume to receive the recalled data set

Explanation: **VOLUME(*volser*)** is an optional parameter specifying the volume that is to receive the recalled data set. This volume is called the target volume. The volume does not have to be managed by DFSMSHsm. For *volser*, substitute the serial number of the target volume.

Defaults: If you do not specify the VOLUME parameter, DFSMSHsm recalls the non-SMS-managed data set to the eligible primary volume with the most space available, unless the recalled data set is associated with a recall pool.

Note:

1. If you specify VOLUME, you must also specify the UNIT parameter.
2. In a JES3 environment, do not specify the *volser* of a volume because JES3 preselects a set of volumes. If you specify a volume that is not in that preselected list, DFSMSHsm fails the recall.
3. When you specify the VOLUME parameter, the target volume must have enough space available for the data set. Otherwise, DFSMSHsm fails the recall.
4. When you specify the VOLUME parameter, DFSMSHsm passes that volume serial number to the automatic class selection (ACS) routines to determine whether the volume is managed by SMS before recalling the data set. See [Chapter 2, “Space management of SMS-managed storage,” on page 23](#) for more information about automatic class selection.
5. If both VOLUME and the VOLCOUNT(N(*nn*)) parameters are specified, the resulting volume count for the data set will be one (1). If space does not exist on the specified volume, the allocation fails.

Examples of how to code the RECALL command

The following examples present different ways to code the RECALL command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Recalling an SMS-managed data set to a non-SMS-managed volume

In this example, a data set that was SMS-managed at the time it was migrated is recalled as a non-SMS-managed data set.

```
RECALL JMG4916.PLACE.TEXT FORCENONSMS VOLUME(VOL255) UNIT(3390)
```


Recalling an SMS-managed data set to a volume selected by ACS services

In this example an SMS-managed data set is recalled to a volume selected by automatic class selection routines.

```
RECALL JMG2384.PLACE2.TEXT
```

Recalling a non-SMS-managed data set to a specific volume

In this example, the data set is recalled to a specific volume.

```
RECALL XXX5634.PARTS.ASM VOLUME(VOL123) UNIT(3390)
```

Recalling a non-SMS-managed data set to a volume selected by DFSMSHsm

In this example, the data set is recalled to a primary volume selected by DFSMSHsm.

```
RECALL LHE1256.REVISION.TEXT
```

Recalling a DA data set for relative block accessing

In this example, a data set is directed to a specific volume and the data is moved so that it can be accessed by relative block addressing.

```
RECALL ND5718.PARTS DAOPTION(RELBLK) +  
VOLUME(VOL195) UNIT(3390)
```

Recalling a data set that failed because of inadequate space

In this example, a data set was not recalled by DFSMSdss because an inadequate space condition occurred. The RECALL command can be issued again with the VOLCOUNT parameter allowing the data set to be recalled to as many volumes as needed (maximum of 59) for successful allocation. DFSMSHsm passes the VOLCOUNT parameter to DFSMSdss.

```
RECALL C253800.ID.PARTS DFDSSOPTION(VOLCOUNT(ANY))
```


Chapter 60. RECOVER command: Recovering a backup version or dump copy of a data set, or UNIX file, or a volume

The RECOVER command causes DFSMSHsm to invoke DFSMSdss to do a physical restore of the requested data set or volume when you specify the FROMDUMP parameter with the command. If you also specify the APPLYINCREMENTAL parameter with the RECOVER command (following the restore), DFSMSHsm recovers data sets incrementally backed up by DFSMSHsm if backup occurred after the dump. You can tell DFSMSHsm to use a particular dump copy as the source of the restore by specifying one of the following parameters with the RECOVER command:

- DATE
- DUMPCCLASS
- DUMPGENERATION
- DUMPVOLUME

When the RECOVER command is issued without the FROMDUMP parameter, DFSMSHsm recovers a level 0 volume, or an incremental backup version of a data set or UNIX file, or a dump copy of a data set, depending on which copy is more current. If there are no records of the data set or UNIX file in the BCDS, and there is no catalog entry for the data set on the volume from which the dump copy was made, the recovery fails. If a dump copy of the data set exists, a RECOVER command issued with the FROMDUMP parameter could successfully recover the data set.

When FROMDUMP DUMPVOLUME is not specified, the restore of a multiple striped data set will be from an incremental backup version. If no incremental backup version exists, the restore will fail. To restore an extended format data set from a dump volume, the FROMDUMP DUMPVOLUME parameter must be used for each extended format data set.

You can use the recovered version of the data set or UNIX file to replace an existing data set or UNIX file accordingly. You can also rename the recovered version to have two versions of the same data set. If you want to recover backup versions of a migrated data set, use one of the following procedures:

Recall the migrated data set and recover a backup version of that data set. Delete the migrated data set and recover a backup version of the deleted data set. Specify the NEWNAME parameter and recover a backup version of that data set. The NEWNAME parameter cannot be the name of a migrated data set. If none of the previous three procedures works, use the FIXCDS command to delete the MCDS data set (MCD) record for the migrated data set and uncatalog the data set. Then, recover the backup version of the data set.

You normally run volume recovery when an entire DASD pack has been lost or damaged or when a significant amount of the data is no longer accessible.

If a data set being recovered spans to a second or more volumes, and the data set resides on disaster alternate volumes that are in a tape library, DFSMSHsm will verify and correct the tape library volume record for the first volume and the original volumes in the volume list only. The message ARC0184I RC=4 is issued for all other volumes that are disaster alternate volumes. The message ARC0184I RC=4 is issued for an MCT record that cannot be found for the disaster alternate volume.

Disaster alternate volumes do not have MCT records. The MCT is for an original volume only. The message ARC0184I RC=4 is not an indication of an error condition for a disaster alternate volume. The disaster alternate volume will be allocated and the recovery process will continue.

This following example shows a volume list for a data set spanning four volumes.


```
TAPEA - DISASTER ALTERNATE VOLUME - TAPE LIBRARY RECORD VERIFIED.
TAPEB - DISASTER ALTERNATE VOLUME - MESSAGE ARC0184I RC=4.
TAPEC - ORIGINAL TAPE VOLUME - TAPE LIBRARY RECORD VERIFIED.
TAPED - DISASTER ALTERNATE VOLUME - MESSAGE ARC184I RC=4.
```

When replacing an existing SMS-managed data set, DFSMSHsm normally recovers or restores the data set to the volume on which the data set is currently cataloged; however, for SMS-managed VSAM data sets, VTOC/Data Set Services selects the target volume. In addition, the SMS class names that currently exist in the catalog for the data set are maintained, regardless of what the class names were at the time of backup or dump.

When recreating a deleted data set, DFSMSHsm determines a new class name by invoking the ACS routines. The result of invoking ACS determines if the data set is to be SMS-managed, VTOC/Data Set Services chooses the target volume, and the newly determined class name is associated with the data set.

The exception to the previous rules for replacing and re-creating SMS-managed data sets is when the FORCENONSMS parameter is issued with the RECOVER command. The FORCENONSMS parameter can force recovery of an SMS-managed data set to a non-SMS-managed volume. It is not possible to convert a data set that is not extended format to a data set that is extended format during recovery. However, you can convert an extended format data set to a data set that is not extended format during a recovery, if you specify the FORCENONSMS optional parameter.

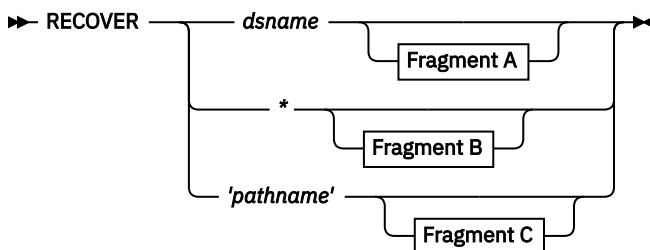
If the dump data was encrypted with a password and DFSMSHsm will be used to recover the dump, then DFSMSHsm will automatically send the encryption password to DFSMSdss to decrypt the data during recovery. If the dump data will be recovered with DFSMSdss directly, then use the LIST DUMPVOLUME command to determine which password should be specified on the DFSMSdss RESTORE command. If the dump data was encrypted with an RSA key label, the data will automatically be decrypted upon recovery if the recovery is performed within the same encryption environment as the dump was created. If the encryption environment has changed or the recovery is being performed in a different environment, the appropriate RSA key label must be specified on the DFSMSHsm RECOVER or DFSMSdss RESTORE command.

The REPLACE subparameter is required when replacing an existing data set.

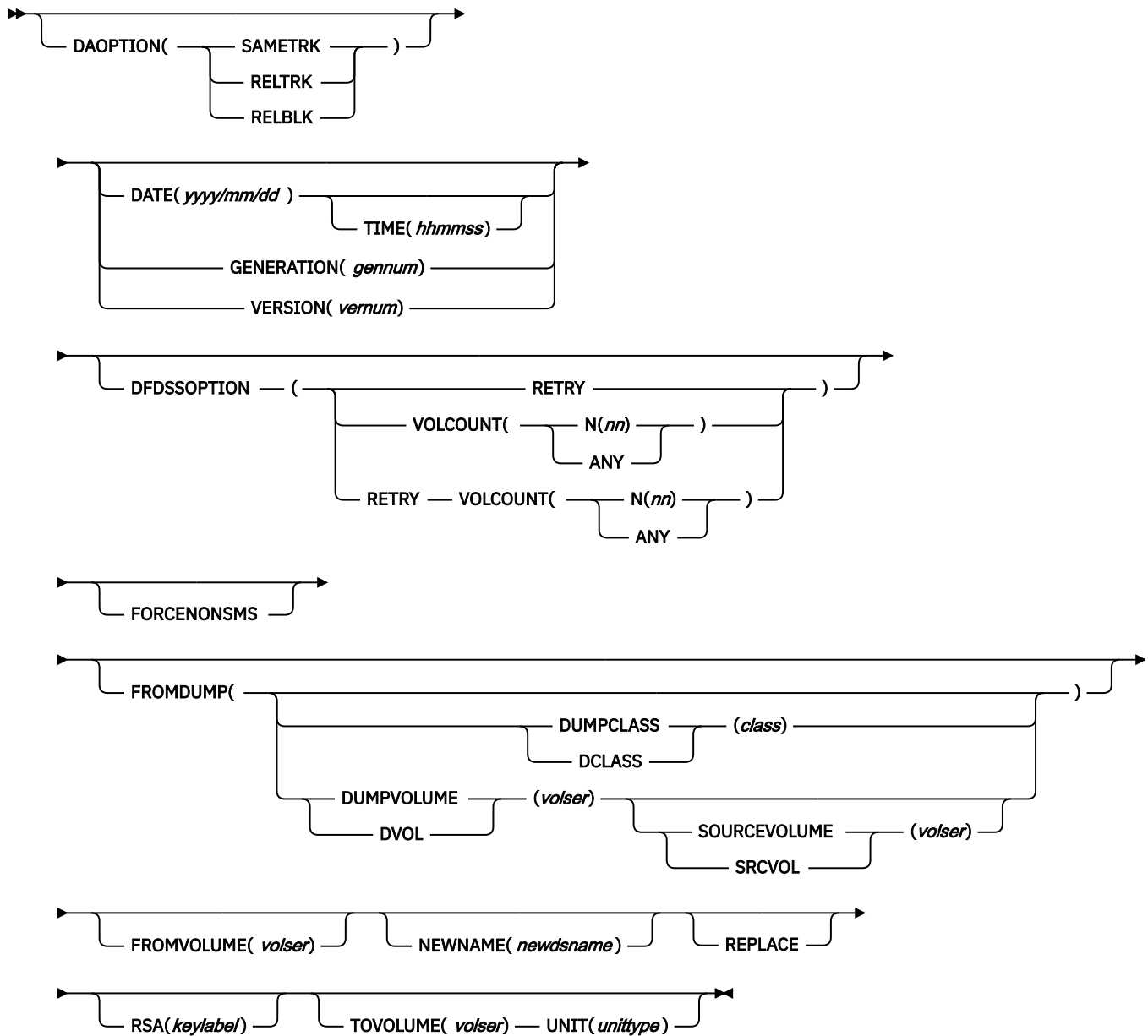
Related reading

- For more information about recovering SMS-managed data sets, see [“Restoring a data set from a DFSMSHsm dump copy”](#) on page 251.
- For more information about recovering non-SMS-managed data sets, see [“Data set restore from a DFSMSHsm dump copy”](#) on page 314.

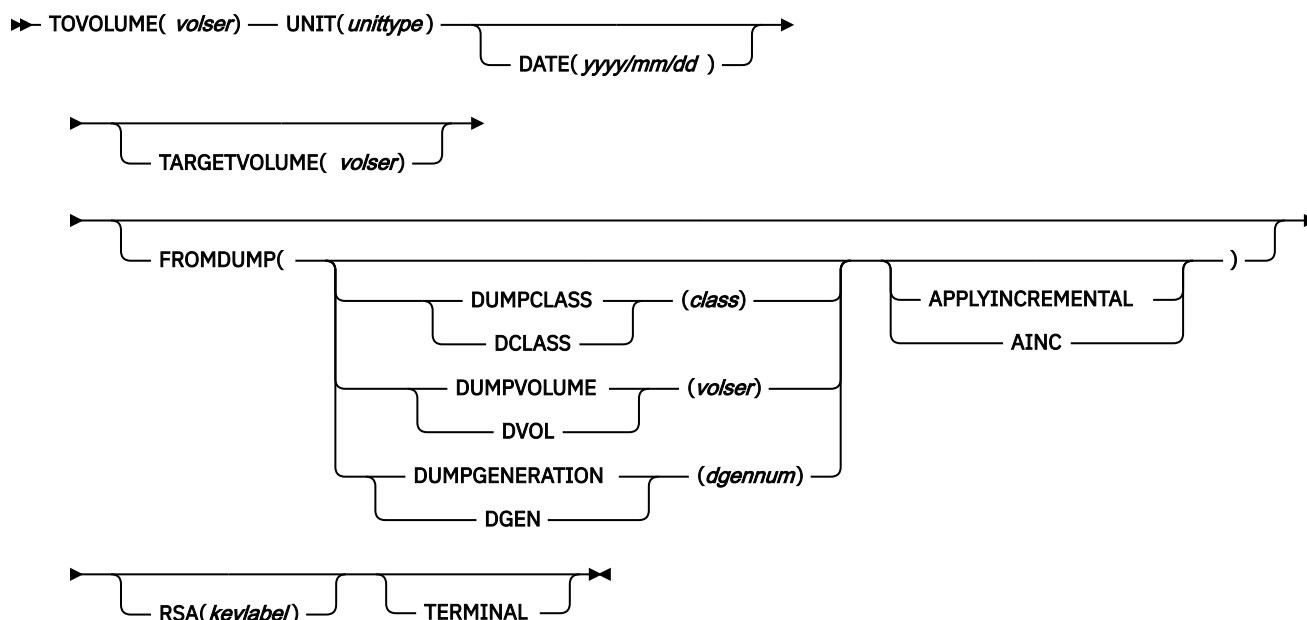
Syntax of the RECOVER command



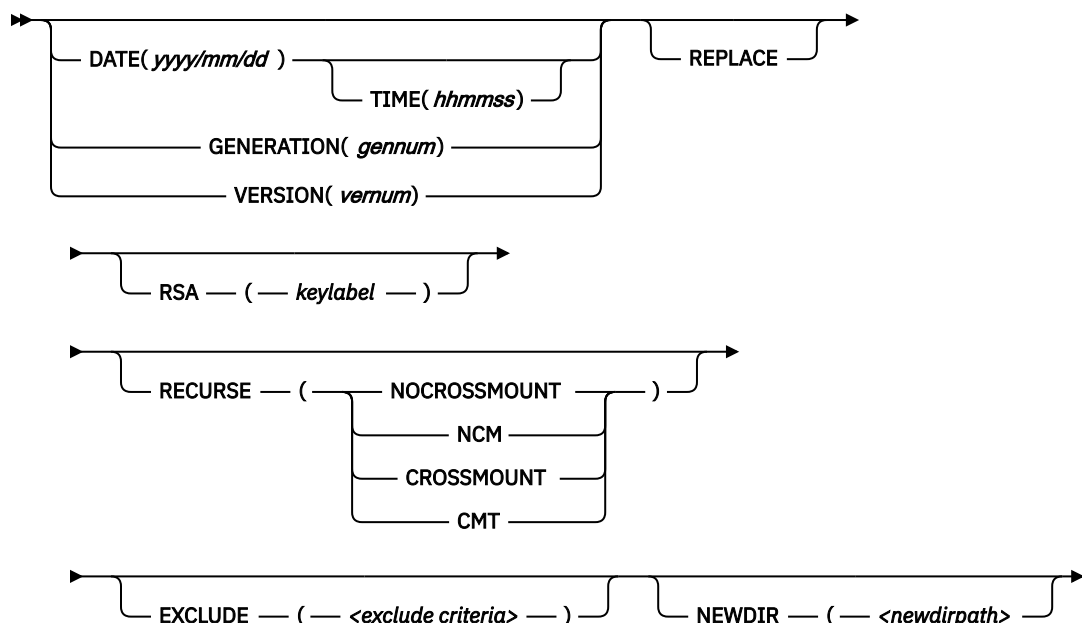
A: RECOVER optional parameters



B: RECOVER optional parameters



C: RECOVER optional parameters



Required parameters of the RECOVER command

This topic describes the required parameters of the RECOVER command.

dsname: Specifying the name of the data set to be recovered

Explanation: *dsname* is a required positional parameter specifying the fully qualified name of the data set you want to recover. For *dsname*, substitute the fully qualified name of the data set you want to recover. You can specify an asterisk (*) as the *dsname* to request that the volume you identify with the TOVOLUME parameter is recovered. Unless FROMDUMP is specified, the volume must be a level 0 volume. If FROMDUMP is specified, the volume can be a DFSMSHsm-owned DASD volume.

During volume recovery, if DFSMSHsm created backup versions of both cataloged and uncataloged data sets that have the same data set name and DFSMSHsm created them from the volume being

recovered, DFSMSHsm recovers the backup version that has the latest creation date and the same data set organization.

Defaults: None.

Note:

1. Because *dsname* is a required positional parameter, you must specify it immediately after the command name. You cannot specify an alias for *dsname*. If you do, DFSMSHsm does not recover the data set.
2. DFSMSHsm does not handle partitioned data set members individually. If you specify a partitioned data set with a member name, DFSMSHsm recovers the entire partitioned data set.
3. A DUMP VTOC copy data set must exist in order to restore a data set from a DUMP copy, except when the DUMPVOLUME subparameter is specified. By using the DUMPVOLUME subparameter, you are deliberately bypassing this requirement; therefore, no data set verification is performed on the dump tape. If the data set you are trying to restore is not on the dump tape, DFSMSHsm still ends with a successful message even though the data set was not restored.
4. DFSMSHsm cannot recover an integrated catalog facility catalog data set from a dump copy. An integrated catalog facility catalog that was dumped by DFSMSHsm can be recovered from a dump copy without DFSMSHsm.

***pathname*: Specifying the name of the UNIX file to be recovered**

Explanation: *pathname* specifies that DFSMSHsm will recover one or more files from their respective backup versions. *pathname*, must be enclosed in single quotation marks and is considered to be made up of two parts: an absolute directory path starting and ending in a forward slash, followed by a *pattern*(/*directory/path/pattern*):

- The *pattern* can contain any valid UNIX file name characters, and may contain wildcard characters ? and *. If the UNIX file name contains the question mark (?), asterisk (*), or backslash (\) characters, the backslash character (\) must be used to escape that character. (for example, \?, *, or \\).
- If *pattern* doesn't contain any wildcard characters, this is treated as an absolute path to a specific UNIX file. This indicates that DFSMSHsm will recover that UNIX file from its backup inventory, if it exists.
- If *pattern* doesn't contain any characters at all, or is only a single asterisk, this is treated as an absolute path to a directory - (/u/dir1/). This indicates that DFSMSHsm will recover all the files and empty directories contained therein. If the RECURSE keyword is specified, DFSMSHsm will examine subdirectories for files to recover.
- Wildcard in *pattern* An asterisk (*) can be used to match any number of characters within a file name. The asterisk may be used at any position in the pattern. A second asterisk may also be used at the end of the pattern to indicate any number of characters following specific text. A question mark (?) can be used to represent any single character in the file name. Multiple ? characters are allowed in the pattern. If the RECURSE keyword is specified, DFSMSHsm will examine subdirectories for files matching the pattern.

Optional parameters of the RECOVER command

This topic describes the optional parameters of the RECOVER command.

DAOPTION: Selecting target volume track length

Explanation: **DAOPTION(SAMETRK | RELTRK | RELBLK)** are mutually exclusive, optional parameters specifying the type of data set accessing required (relative track or relative block) upon recovery of a direct access data set. This, in turn, implies the allowable target volume device types.

Subparameter	Explanation
SAMETRK	Data is moved as a track-to-track image and the target volume track size must be the same as the L0 volume from which the data set was backed up. If DFSMSHsm is directed to a specific volume by the TOVOLUME parameter, this volume must have the same track length as the L0 volume from which the data set was backed up. Data can be accessed for both relative track and relative block processing.
RELTRK	Data is moved as a track-to-track image and the target volume track size must be the same or greater than the L0 volume from which the data set was backed up. If DFSMSHsm is directed to a specific volume by the TOVOLUME parameter, this volume must have the same or greater track length as the L0 volume from which the data set was backed up. Data can be accessed for relative track processing.
RELBLK	Data is moved to fill out the track and the target volume track size can be the same, greater, or smaller than the L0 volume from which the data set was backed up. Data can be accessed for relative block processing.

Defaults: If DAOPTION is not specified, the data is moved as a track image, allowing for relative track accessing.

Note: Only basic direct access method (BDAM) data sets are supported by this option. Using DAOPTION to recover a data set as SMS-managed is not supported. If a data set would be SMS-managed after the recover, the FORCENONSMS parameter of RECOVER must be used to force it to be non-SMS-managed.

DATE, GENERATION, and VERSION: Specifying the particular backup version or dump copy to recover

Explanation: **DATE**(*yyyy/mm/dd*), **GENERATION**(*gennum*), and **VERSION**(*vernum*) are mutually exclusive, optional parameters specifying the particular backup version or dump copy of one or more data set or UNIX file you want to recover.

DATE(*yyyy/mm/dd*) can be used to specify the particular dump copy to be used for a restore. DATE applies when recovering a backup version or dump copy of a specific data set or UNIX file or an entire volume. For *yyyy/mm/dd*, substitute the date for the backup version or dump copy of the data set or UNIX file or volume that you want to recover in the following form. A leading zero is not required for a one-digit month or day. If DFSMSHsm is recovering a data set or UNIX file, it recovers the latest backup version that was created on or before the date you specified. If DFSMSHsm is recovering a volume, DFSMSHsm recovers the latest backup version of each data set that was backed up on or after the date you specified. If DFSMSHsm is recovering a volume from a dump copy, the most recent dump copy made on or before the date is selected. If DFSMSHsm is recovering a data set from a dump copy and the data set is cataloged, the most recent dump copy made on or before the date is selected. If the data set is not cataloged, there must be an incremental backup version created on or prior to the date for the data set to be recovered from the dump copy.

TIME(*hhmmss*)

is an optional sub-parameter of DATE that specifies the exact time in hours, minutes, and seconds (*hhmmss*) when the backup version is created. The valid range for hours is 00–23, and for minutes and seconds is 00–59. If you specify TIME, you must also specify DATE, otherwise the RECOVER command fails. You must specify a fully qualified data set name. You cannot specify TIME on a volume recovery request. If a version with specified date and time cannot be found, the command fails.

GENERATION(*gennum*) specifies that you want to recover a particular incremental backup version of a specific data set or UNIX file. For *gennum*, substitute the relative generation number of the backup version of the data set or UNIX file to be recovered. A 0 specifies the latest version, 1 specifies the next to the latest version, and so on, to the maximum number of versions existing for the data set or UNIX file. The *gennum* can be up to 3 digits long.

VERSION(*vernum*) specifies that you want to recover a particular version of a backed up data set or UNIX file. For *vernum*, substitute a decimal number from 1 to 999 for the particular backup version you want to recover. If DFSMSHsm is unable to find the specified backup version, the recover fails and message ARC1128I is issued.

Defaults: If you do not specify DATE, GENERATION, or VERSION, DFSMSHsm recovers the latest backup version of the data set or UNIX file.

Note:

1. The GENERATION and VERSION parameters do not apply to volume recovery. If you specify GENERATION or VERSION when it does not apply, DFSMSHsm ignores it.
2. The GENERATION or VERSION parameters should not be specified if FROMDUMP is specified.
3. DATE should not be specified if DUMPVOLUME or DUMPGENERATION is specified.
4. If either the GENERATION or the VERSION parameter is specified, DFSMSHsm selects only from incremental backups and will not recover from physical dump copies, even if the physical dump is more recent.
5. To recover a retained backup version specify the date, and if necessary, the time that the retained backup copy was created. You cannot recover a retained copy with the GENERATION and VERSION keywords.

DFDSSOPTION: Pass optional parameters to DFSMSdss

Explanation: **DFDSSOPTION(RETRY | VOLCOUNT)** is an optional parameter that causes DFSMSHsm to pass optional parameters to DFSMSdss when recovering certain types of data sets or to specify the number of volumes a data set can extend to.

RETRY is applicable to the recovery of the following types of data sets and is ignored by DFSMSHsm if specified for other types of data sets:

- Partitioned data set: When the initial recovery of the data set fails and a DFSMSdss message requests the use of the NOPACKING option. RETRY specifies for DFSMSHsm to pass this NOPACKING option to DFSMSdss.
- Keyed VSAM data set: When the initial recovery of the data set fails and a DFSMSdss message requests to use the FREESPACE(0 0) option, to retry the restore to a like device, or both. If the data set is not SMS-managed, also specify the TOVOLUME(volser) UNIT(unitttype) parameters on the recover command. RETRY specifies for DFSMSHsm to pass this FREESPACE option to DFSMSdss.

VOLCOUNT(ANY | N(*nn*)) allows an SMS data set to extend to additional volumes within the same storage group, when space is available. The VOLCOUNT parameter is limited to SMS data set types supported as multivolume data sets. Refer to [z/OS DFSMSdss Storage Administration](#) for more information regarding restrictions and limitations of the VOLCOUNT parameter.

VOLCOUNT(ANY) results in the allocation of the target data set on as many volumes as required, to a maximum of 59.

VOLCOUNT(N(*nn*)) allows users to specify the number of volumes to be used for SMS data set allocation. *nn* can be any value from 0 to 59. When you specify the number of volumes the data set is allocated to, the allocation of the data set will be limited to that number of volumes or will be extended to that number of volumes.

The VOLCOUNT(N(*nn*)) parameter is ignored if specified with the DFSMSHsm TOVOLUME parameter. If a volume count VOLCOUNT(N(*nn*)) is specified and the TOVOLUME parameter is also specified, the resulting volume count for the data set will be one (1). If inadequate space exists on the specified volume, the allocation fails.

Defaults: None.

Note: If you specify the RECOVER command with the VOLCOUNT parameter, it overrides any option used on the SETSYS VOLCOUNT command.

EXCLUDE and EXCLUDEF: Specifying files or directories to be excluded from processing

EXCLUDE | EXCLUDEF are mutually exclusive, optional parameters that are used to specify files and directories that are to be excluded during Unix File backup processing.

EXCLUDE(<exclude_criteria>) - specifies a comma-separated list of exclude criteria directly on the command. This criteria is to be applied to the UNIX files examined during the UNIX File backup processing.

The exclude criteria list specified on the command with EXCLUDE keyword has the following restrictions and rules unique to the keyword:

- The *exclude_criteria* list must be specified in single quotes
- The *exclude_criteria* list cannot exceed 1023 characters

EXCLUDEF(' <filepath>') - specifies the UNIX path to a file that contains the exclude criteria to be applied to the UNIX files examined during the UNIX File backup processing. The file path must be specified in single-quotes and must be an absolute path. Any imbedded single quotes in the filename must be specified with a second single quote, e.g. the filename exclude'criteria.txt would be specified as: 'exclude'criteria.txt'. (optional)

The exclude criteria file specified using the EXCLUDEF keyword has the following restrictions and rules unique to the keyword:

- Line starting with a blank character is a comment.
- Each line can only contain one criterion.
- A single line cannot be longer than 1023 characters.
- Total length of all exclude criteria with added commas to separate them cannot exceed 32767 characters. For 'n' number of criteria, 'n-1' number of COMMA should also be accounted for to calculate the total size.

Exclude criteria, whether specified on the command in a comma separated list using the EXCLUDE keyword or specified within a file using the EXCLUDEF keyword, must conform to the following rules:

- Absolute directory path, beginning and ending with a forward-slash character.
- Partial directory path, ending with a forward-slash character. If the directory name begins with a forward slash character, then it is considered an absolute path. If the directory name does not begin with a forward slash character, it is considered a relative path that begins at the directory requested on the command. A directory name that is a relative path is considered relative to the directory requested on the command only, even when the command requests recursion.
- Filename pattern (does not contain forward slashes). Can contain * or ? wildcard. An asterisk (*) can match with 0 or more characters in a filename. A question mark (?) can match with a single character. Wildcard characters may be escaped with the backslash character if desired.
- A filename pattern can begin with an absolute or relative path, however wildcard characters can only be used in the filename pattern. A directory path containing an asterisk (*) or question mark (?) is considered an invalid pattern.
- If a filename contains special characters that are not to be treated as wildcard or escape characters i.e. an asterisk (*), a question mark (?), or backslash (\), these characters must be escaped using a backslash (\). If these characters are not escaped, they will be treated as wildcard or escape characters.
- A directory wildcard may be specified in the directory portion of a criterion by specifying /.../ (<forward slash><period><period><period><forward slash>). A directory wildcard can match with 0 or more directories.
- If a criterion begins with a directory wildcard (/.../) then it is treated as relative path.
- A comma cannot be used with the exclude criterion.
- A single criterion cannot exceed 1023 characters.
- A single forward slash or a single asterisk are treated as invalid patterns.

- A directory path cannot contain consecutive forward slashes.

The example below shows sample criteria in a file specified by the EXCLUDEDEF keyword. The same criteria could be specified on the command using the EXCLUDE keyword where each criterion is separated by a comma.

```
Useful comment (maybe with commas) <= blank in 1st position - ignored
                                     <= empty line - ignored
/a/b/c/                             <= absolute directory
/a/b/c/x1.txt                       <= absolute directory with filename
/a/.../d/.../g/syslog1.txt         <= absolute directory with wildcards
/a/.../d/.../g/*log?.txt           <= absolute directory and filename with wildcards
e/f/g                               <= relative directory
e/f/g/mvs.log                      <= relative directory with filename
d/e/f/g/*log?.txt                  <= relative directory with filename wildcard
d/.../g/sys*. *                    <= relative directory and filename with wildcards
/.../d/.../g/sys*. *               <= relative difrectory and filename with wildcards
*.log                              <= filename with wildcards
*.log2                             <= blank in 1st position makes a comment, ignored
\*abc                              <= escaping wildcards - files with exact name *abc
\\abc                              <= escaping wildcards - files with exact name \abc
```

Defaults: No files are excluded from processing.

FORCENONSMS: Recovering an SMS-managed data set to a non-SMS-managed volume

Explanation: **FORCENONSMS** is an optional parameter that forces a data set to be recovered to a non-SMS-managed volume. Use this parameter if SMS is not active or if SMS is active and you need to return a data set to non-SMS-managed status.

Defaults: None.

Note:

1. If SMS is not active, a recover request for an SMS-managed data set fails unless the FORCENONSMS parameter is specified on the RECOVER command.
2. A non-SMS target volume must be known to DFSMSHsm to complete the request. This can be obtained through:
 - The TOVOLUME and UNIT parameters if the data was backed up as SMS.
 - The volume from which the data set was backed up if backed up as non-SMS.
 - The volume from which the data set migrated if migrated as non-SMS.

FROMDUMP: Specifying which dump copy is used for restoring data sets

Explanation: **FROMDUMP** is an optional parameter specifying which dump copy should be used for the restore.

FROMDUMP specifies the recovery should use a dump copy from dump volumes.

DUMPClass(*class*) | DUMPVOLUME(*volser*) | DUMPGENERATION(*dgennum*) are mutually exclusive, optional subparameters of FROMDUMP specifying a particular class, volume, or generation from which the restore should be made. Neither DUMPGENERATION nor DUMPVOLUME should be specified if the DATE parameter is specified.

Subparameter	Explanation
DUMPClass(<i>class</i>)	The dump copy to be used to restore a data set or volume should come from a particular class. This subparameter is needed only if the source volume was dumped to more than one dump class. For <i>class</i> , substitute from 1 to 8 alphanumeric characters for the class of the volume from which the data set or volume is to be restored.

Subparameter	Explanation
DUMPVOLUME(<i>volser</i>)	A particular dump volume is used for restoring a data set or a volume. For <i>volser</i> , substitute the volume serial number of one of the dump volumes that contains part of the wanted dump copy of the source volume to be restored. DUMPVOLUME is not valid when GENERATION, VERSION, DATE, DUMPCLASS, or DUMPGENERATION are specified.
SOURCEVOLUME(<i>volser</i>)	The identified dump copy is used when the data set to be recovered from a dump is not cataloged or is migrated, and has no backup copy, and more than one valid dump copy is stacked on the volume specified with DUMPVOLUME.

If a multivolume, cataloged data set is to be recovered, the RECOVER command must be issued with the FROMDUMP parameter and the DUMPVOLUME subparameter to ensure that DFSMSHsm directs each part of the data set to the correct volume. In this case, the TOVOLUME parameter is optional but, if used, must be the name of the volume as identified in the catalog entry.

There must be similar device types between the original DASD and the target DASD, with the exception that you may recover data dumped from a 3380 DASD to a 3390 DASD that is in 3380 emulation mode. The following is a scenario example:

- The volume serial number PRIM01 is a 3380 DASD dumped to tape cartridge DUMP05.
- All 3380 DASD devices have been replaced by 3390 DASD.
- The 3390 DASD runs in a 3380 emulation mode and initializes with the volume serial number PRIM01.
- The command to recover PRIM01 is as follows:

```
RECOVER * TOVOLUME (PRIM01) UNIT (3380) FROMDUMP (DUMPVOL (DUMP05))
```

DUMPGENERATION(*dgennum*) specifies that a particular dump generation should be used as a source for restoring a volume. The *dgennum*, can be up to three digits long, but its value must be between zero and 99. Zero is the most-recent generation. Use the LIST command to determine what generation numbers exist. DUMPGENERATION should not be specified if you are restoring a single data set.

APPLYINCREMENTAL specifies that an incremental volume recovery process is to follow immediately after the volume restore and is to be a part of the single request to recover a specific volume. APPLYINCREMENTAL has meaning only when a volume is being restored. The volume does not have to have been processed by the DFSMSHsm volume backup function. If the volume was not backed up with the DFSMSHsm function, DFSMSHsm still tries to find more recent backup versions created from individual data set backup requests after the dump copy was created. It is not applicable if a data set request is entered.

Note:

1. The APPLYINCREMENTAL parameter is not honored if DFSMSHsm determines that the target volume is owned by a ICF catalog that resides on the target volume. This is to avoid conflicts in the way DFSMSHsm supports ICF catalogs and the data sets cataloged in them.
2. If Data Facility Product Version 2 Release 3.0 or higher is not installed, you cannot use the APPLYINCREMENTAL parameter during backup and recovery when the target volume has an Integrated Catalog Facility catalog.
3. The APPLYINCREMENTAL parameter is not allowed when the target volume is a DFSMSHsm-owned volume.
4. The SOURCEVOLUME subparameter is only used to identify which dump copy on a dump volume should be used to restore a data set. It does not have all the implications of the FROMVOLUME parameter.
5. If you use DUMPVOLUME to specify a dump tape, and that dump volume has more than one valid dump copy stacked on it, each dump copy is for a different source volume. For DFSMSHsm to restore the data set, it must be told which source volume (which dump copy) is needed. It uses information in

the following order: FROMVOLUME (if specified); SOURCEVOLUME (if specified); the volser where the data set is cataloged; if not cataloged, the volume from which the data set was last backed up. If none of these is specified or available, the RECOVER command fails.

6. The VTOC and the volume serial on the target volume may have changed as a result of this operation. Before the volume can be accessed on any remote system, the UCB must be refreshed. The refresh occurs automatically if the volume is online and the device manager REFUCB function is enabled. You enable the REFUCB function through PARMLIB member DEVSUPxx or the MODIFY DEVMAN command. For more information, refer to the description of the REFUCB keyword in [z/OS MVS Initialization and Tuning Reference](#) or [z/OS MVS System Commands](#).

Defaults: None.

FROMVOLUME: Specifying the volume from which the backup version of a data set has been created

Explanation: **FROMVOLUME(volser)** is an optional parameter specifying that the data set was uncataloged and that it resided on the volume specified by *volser* when DFSMSHsm created the backup version or dump copy. For *volser*, substitute the serial number of the volume where the data set resided when DFSMSHsm created the backup version or dump copy. If FROMVOLUME is used to direct DFSMSHsm to restore a data set from a dump copy, the dump copy made from FROMVOLUME specified is used regardless of the catalog status of the data set when the dump copy was first made. The data set is restored to the FROMVOLUME specified and left uncataloged.

Defaults: None.

Note:

1. The FROMVOLUME parameter applies only to data set recovery; it does not apply to volume recovery. If you specify FROMVOLUME when it does not apply, DFSMSHsm ignores it.
2. You must specify the FROMVOLUME parameter if the data set you want to recover was uncataloged when it was backed up.
3. DFSMSHsm recovers only cataloged data sets unless you specify FROMVOLUME.
4. DFSMSHsm does not catalog the recovered backup version or dump copy when you specify the FROMVOLUME parameter.
5. The FROMVOLUME parameter does not apply to VSAM data sets.
6. If the data set does not exist on the target volume for the recovery, automatic class selection routines are invoked to determine if the data set should be recovered as SMS, if FROMVOLUME is not specified.
7. If FROMVOLUME is specified, DFSMSHsm recovers the data set as a non-SMS.

NEWDIR : Recover files to a different directory

During a file recovery request, this optional parameter specifies a target directory that the recovered files should be placed into. The specified *newdirpath* must be an absolute path ending in a forward slash, for an existing directory. The requesting user must have search and write permission for the directory.

Restriction: This parameter only applies when a file path is specified on the request. The total length of the recovered file names plus the length of the new directory cannot exceed 1023 characters.

Defaults: Recovered files are placed in the original location where they existed at the time of backup.

NEWNAME: Specifying a new data set name for the recovered data set

Explanation: **NEWNAME(newdsname)** is an optional parameter specifying a new, fully qualified data set name for the recovered backup version or dump copy. For *newdsname*, substitute a new, fully qualified data set name for the recovered data set.

If a cataloged data set exists with the same name as the new name you are specifying, you must specify the REPLACE parameter to replace the existing cataloged data set. If an uncataloged data set with the

same name as the new name you are specifying exists on the volume, you must specify the REPLACE parameter to replace the existing uncataloged data set.

Defaults: None.

Note:

1. The NEWNAME parameter applies only to data set recovery. If you specify NEWNAME when it does not apply, DFSMSHsm ignores it.
2. You must specify NEWNAME when a data set exists with the same data set name as the data set that you are recovering and you want to keep both versions of the data set.
3. If a VSAM data set is being recovered and you specify NEWNAME, the new data set name can be cataloged in the same or different catalog as the original data set under the following restrictions:
 - a. If the new data set will be cataloged in a different catalog than the original, only the base cluster will be recovered, even when there were alternate indexes at the time of backup. You may use the access method services (AMS) BLDINDEX facility to rebuild the alternate indexes (AIX).
 - b. If the old data set does not exist, the new data set can be recovered provided the original data set's data and index components follow these standard naming conventions:
 - The old data component was the base cluster name followed by the .DATA qualifier.
 - The old index component was the base cluster name followed by the .INDEX qualifier.

For VSAM data sets not cataloged in the Integrated Catalog Facility catalog, if you specify TOVOLUME with NEWNAME, the volume must be owned by the catalog where the data set being recovered will be cataloged.

4. The following only applies if DFHSM 2.6.0 (or earlier release) was used to move the data during backup. When you specify NEWNAME for VSAM data sets, DFSMSHsm will fail the recovery if the original data set exists and has two or more paths defined to the base cluster. In some cases the path defined to the base cluster will not be recovered; use the (AMS) DEFINE PATH command.

Additionally, the data and index object names of the data set being recovered must be the same as the data and index object names that were exported. The object names of the data set being recovered will be passed to AMS IMPORT for renaming the NEWNAME data set. The access method services facility (AMS) then verifies that the data and index object names of the data set being recovered are the same as the ones that were exported.

5. DFSMSHsm does not handle partitioned data set members individually. If you specify a partitioned data set with a member name for either the data set name or the new data set name, DFSMSHsm fails the command.
6. You can recover a backup version of a migrated data set if you specify the NEWNAME parameter. For VSAM data sets, the following is required for recovery to a new name:
 - There is zero or one path defined to the base cluster, and
 - There is zero or one alternate index, and
 - There is zero or one path defined to the alternate index.

The newname parameter cannot be the name of the migrated data set.

RECURSE: Examine subdirectories of the specified UNIX file path

Explanation: **RECURSE(CROSSMOUNTS|CMT|NOCROSSMOUNTS|NCM)** Requests DFSMSHsm to examine subdirectories of the specified path, looking for files that match pattern to recover from the DFSMSHsm backup inventory.

NOCROSSMOUNTS is the default, and directs DFSMSHsm to not recover files that came from a different file system than the initial specified path.

CROSSMOUNTS tells DFSMSHsm to examine directories that are in a different filesystem from the file system containing the starting location when choosing files to recover.

Defaults: None. Only the requested directory is examined for files to recover.

REPLACE: Specifying that the recovered backup version or dump copy is to replace any existing data set of the same name

Explanation: **REPLACE** is an optional parameter specifying the recovered backup version or a dump copy of the uncataloged data set replaces any existing uncataloged data set with the same data set name on the target volume. If the data set is cataloged, DFSMSHsm scratches it and recatalogs the data set with the same data set name. If you specify the NEWNAME parameter and the new name is the same as the name of an existing data set, you must specify the REPLACE parameter. If you do not, DFSMSHsm fails the RECOVER command..

Defaults: When DFSMSHsm recovers a volume, it always replaces data sets with the same name on the target volume.

Note:

1. If you do not specify the REPLACE or NEWNAME parameter for an existing data set, DFSMSHsm does not recover the data set. For cataloged data sets, the old copy remains cataloged. For uncataloged data sets, the data set by the same name remains on the target volume.
2. The data set to be replaced must have the same organization as the data set from which the backup version was created. If the backup version is being recovered and both the backup version and the target data set are cataloged, DFSMSHsm fails the RECOVER request when the data set organizations do not match.

RSA: Specifying that the RSA parameter is to override the RSA key label

Explanation: **RSA** Specifies the RSA parameter to override the RSA key label that DFSMSdss would otherwise use to restore the encrypted dump. The key label can consist of up to 64 characters, of which the first character must be alphabetic or national.

TARGETVOLUME: Specifying the volume to receive the restored data sets

Explanation: **TARGETVOLUME**(*volser*) is an optional parameter specifying that volume restore should use the volume specified by *volser* and change this *volser* to the *volser* of the TOVOLUME parameter during the restore operation.

Defaults: None

Note:

1. TARGETVOLUME is only valid when the TOVOLUME and FROMDUMP parameters are specified. If you specify TARGETVOLUME when it does not apply, an error message will be issued.
2. If you do not specify TARGETVOLUME, a volume with the *volser* specified in the TOVOLUME parameter must exist and be online.
3. If an SMS-managed volume is being restored and APPLYINCREMENTAL is specified, either the TARGETVOLUME or another SMS-managed volume must be available to be selected as directed by your ACS routines in order for the APPLYINCREMENTAL portion to recover later backup versions. Note that even if the TARGETVOLUME is SMS and selectable, APPLYINCREMENTAL processing may select a different SMS volume if available.
4. If APPLYINCREMENTAL and TARGETVOLUME are specified, the TOVOLUME must be varied OFFLINE before issuing the RECOVER command. If the TOVOLUME is ONLINE, the command will fail and new message ARC0256I will be issued. This new message is documented below.
5. After the RECOVER command completes, additional steps may be required to complete the volume change and to establish access to the restored volume. Check all DFSMSHsm, DFSMSdss, and system messages issued during the Volume Restore. Since the TARGETVOLUME must be online, if the TOVOLUME is also ONLINE, check for DFSMSdss message ADR344I indicating that a duplicate volume will be varied offline. ADR344I includes the unit to be varied offline. Following this, check for messages like IEF281I 0E19 NOW OFFLINE indicating the volume is offline. If this is the unit for TARGETVOLUME, after the command completes, you will need to vary the TOVOLUME unit OFFLINE

and then vary the TARGETVOLUME unit ONLINE. You should display the units before issuing the command, and note the units and associated volsers for both the TOVOLUME and TARGETVOLUME.

TERMINAL: Requesting that volume recovery messages appear at the terminal

Explanation: **TERMINAL** is an optional parameter requesting that volume recovery messages be sent to the system console and to the backup activity log when DFSMSHsm recovers a volume.

Defaults: If you do not specify **TERMINAL** when you want to recover a volume, DFSMSHsm sends the output only to the backup activity log.

Note: **TERMINAL** applies only to volume recovery. In addition, you must issue the **TERMINAL** parameter of the **RECOVER** command from the system console. If you specify **TERMINAL** when it does not apply, DFSMSHsm ignores it.

TOVOLUME: Specifying the volume to receive the recovered data set

Explanation: If you are recovering a data set, **TOVOLUME(volser)** is an optional parameter specifying which volume should receive the recovered data set. For *volser*, substitute the serial number of the volume that is to receive one or more recovered data sets.

If you are recovering a volume, **TOVOLUME(volser)** is a required parameter specifying which volume you want to recover. For *volser*, substitute the serial number of the volume that is to be recovered. The *volser* that is to be recovered must be the same as the source *volser* of the volume that was backed up or dumped.

If you specify **TOVOLUME** when recovering an SMS-managed data set, the *volser* will be passed to the automatic class selection (ACS) routines; ACS, however, ultimately determines to which volume the data set is recovered. The **TOVOLUME** parameter is ignored if an existing SMS data set is recovered as SMS.

Defaults: If you are recovering from a backup version of a cataloged data set and do not specify **TOVOLUME**, DFSMSHsm recovers the backup version to the volume where the catalog entry shows the data set currently exists. If no catalog entry exists, DFSMSHsm recovers the backup version to the volume where the data set resided when it was backed up unless the backup version was created from a data set on a migration volume. If the data set was backed up while residing on a migration volume and you do not specify the **TOVOLUME** parameter, DFSMSHsm recovers the backup version to the volume from which the data set last migrated.

If you are recovering from a backup version of an uncataloged data set and do not specify **TOVOLUME**, DFSMSHsm recovers the backup version to the volume where the data set resided when it was backed up, unless it is recovered as SMS.

Note:

1. If you specify **TOVOLUME**, you must also specify the **UNIT** parameter.
2. If DFSMSHsm is recovering a VSAM data set not cataloged in an integrated catalog facility catalog, **TOVOLUME** can only specify a volume owned by the catalog where the data set being recovered is cataloged or will be cataloged.
3. The **TOVOLUME** parameter is passed to the ACS routines when processing a data set that will be recovered to an SMS-managed volume.
4. If a data set that was SMS-managed when backed up has been deleted, and if ACS determines that data set is now non-SMS, **TOVOLUME** is required to recover that data set to a non-SMS-managed volume. If **TOVOLUME** is not specified, the recovery will fail with message ARC1170I and reason code 13.
5. If both **TOVOLUME** and the **VOLCOUNT(N(nn))** parameters are specified, the resulting volume count for the data set will be one (1). If space does not exist on the specified volume, the allocation fails.

Attention

If you specify TOVOLUME for a data set that was backed up with DFSMSdss and you direct it to a volume that has an uncataloged data set with the same name, DFSMSdss tries to restore the cataloged data set on top of the uncataloged data set; as a result, your attempt fails. DFSMSHsm scratches the uncataloged data set because DFSMSdss fails to restore the cataloged data set.

UNIT: Specifying the type of device

Explanation: **UNIT(*unittype*)** is an optional parameter specifying the type of unit where DFSMSHsm can mount the target volume. You use this parameter with the TOVOLUME parameter.

DFSMSHsm incremental volume recovery and volume restore with APPLYINCREMENTAL do not support a device change on the volume that is recovered. If the device type is changed after the volume is backed up and the device type in the catalog for all data sets residing on the volume is not changed, all VSAM data sets will not be recovered. All non-VSAM data sets may appear to be recovered successfully, but the first allocation after the recovery will fail due to the inconsistent unit name and device type that was created by the recovery.

If you specify the type of unit when recovering an SMS-managed data set, that unit type will be passed to the automatic class selection (ACS) routines; ACS, however, ultimately determines to which unit type the data set is recovered. For *unittype*, substitute the type of unit where DFSMSHsm is to mount the target volume.

The following are valid types of DASD units:

- 3380
- 3390
- 9345

Defaults: None.

Note:

1. If you specify UNIT, you must also specify the TOVOLUME parameter.
2. If the block size of the data set is greater than the track capacity of the target volume, track overflow must be supported in both the software and the hardware. This applies to devices whose track size is less than 32 KB.

Examples of how to code the RECOVER command

The following examples present different ways to code the RECOVER command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Recovering a cataloged data set to a specific volume

In this example, a backup version or dump copy of a cataloged data set is recovered to the TOVOLUME specified.

```
RECOVER PAC3105.PROJECT.LOAD TOVOLUME(VOL005) UNIT(3390)
```

Recovering an uncataloged data set to a specific volume

In this example, a data set that was uncataloged when it was backed up or dumped is recovered to the TOVOLUME specified.

```
RECOVER JAB1234.ANALYSIS.TEXT TOVOLUME(VOL001) UNIT(3390) +
FROMVOLUME(JAB456)
```


Recovering a next-to-latest backup generation of a cataloged data set

In this example, a next-to-latest backup generation of a cataloged data set is recovered.

```
RECOVER MIL9876.SETUP.ASM GENERATION(1)
```

Recovering the oldest backup version of a cataloged data set

In this example, the oldest backup version of a cataloged data set is recovered.

```
RECOVER C253800.PUBS.ASM VERSION(1)
```

Recovering a data set to replace an existing data set with the same name

In this example, a data set is recovered to the volume the data set is currently cataloged on, and it replaces a data set having the same name. DFSMSHsm scratches the original data set.

```
RECOVER PAC5432.REPORT.CNTL REPLACE
```

Recovering and renaming a data set

In this example, the recovered data set is given a different name. The original data set remains unchanged.

```
RECOVER MIL3210.INPUT.LIST NEWNAME(MIL3210.FINAL.LIST)
```

Recovering an entire volume

In this example, an entire volume is recovered. It is assumed that the volume is current as of the date specified. For example, the volume has been restored with a DFSMSdss dump tape made on that date. Data sets are recovered to the volume only if a backup version was created on or after the specified date.

```
RECOVER * TOVOLUME(SAR005) UNIT(3390) DATE(1991/01/01)
```

Recovering the most recently dumped version of a data set of a specific dump class and renaming the data set

In this example, the most recently dumped version of a specific data set that was dumped to a specific dump class (EXTRA) is recovered and given a different name.

```
RECOVER B110066.POF.N.F230EP06.DSET3 +  
NEWNAME(B110066.POFNEW.NRECOV.F230EP06.DSET3) +  
FROMDUMP(DUMPCCLASS(EXTRA))
```

Recovering the second-most recently dumped version of a volume

In this example, the second-most recently dumped version is used to do a full-volume restore of volume P1. Immediately following the restore, the data sets that were incrementally backed up since the dump occurred are recovered.

```
RECOVER * TOVOLUME(PTU001) +  
FROMDUMP(DUMPGENERATION(1) APPLYINCREMENTAL) +  
UNIT(3390)
```


Recovering a data set and forcing it to a non-SMS-managed volume

In this example, a data set that was SMS-managed when it was backed up or dumped is being forced to a non-SMS-managed volume during recovery.

```
RECOVER SMS.REPT.CNTL FORCENONSMS +
TOVOLUME(VOL006) UNIT(3390)
```

Recovering a data set for relative block accessing

In this example, a data set is to be recovered so that it can be accessed for relative block addressing. Even though the data set would have been SMS-managed after the recovery, it is directed to a non-SMS volume.

```
RECOVER TK1789.PARTS.DA3 DAOPTION(RELBLK) +
TOVOLUME(NONSMS) UNIT(3390) FORCENONSMS
```

Recovering a migrated data set to level 0

In this example, a non-VSAM, migrated data set is recovered. The NEWNAME parameter identifies a data set that is not migrated.

```
RECOVER TEST.DSET.N01 NEWNAME(TEST.DSET.N02)
```

Recovering a lost volume to a DASD volume with a different volser

In this example, a lost volume is to be recovered to a DASD volume with a different volser. The TARGETVOLUME parameter saves you the step of "clipping" a volume to the lost volser.

```
RECOVER * TOVOLUME(VOL008) UNIT(3390) TARGETVOLUME(SAR10) +
FROMDUMP(DUMPVOLUME(DUMP05) APPLYINCREMENTAL)
```

Recovering a data set that failed because of inadequate space

In this example, a data set was not recovered by DFSMSdss because an inadequate space condition occurred. The RECOVER command can be issued again with the VOLCOUNT parameter allowing the data set to be recovered to as many volumes as needed (maximum of 59) for successful allocation. DFSMSHsm passes the VOLCOUNT parameter to DFSMSdss.

```
RECOVER C253800.ID.PARTS DFDSOPTION(VOLCOUNT(ANY))
```

Chapter 61. RECYCLE command: Consolidating valid data to one tape from other tapes

The RECYCLE command consolidates data from any migration or backup tape volume (except an unassigned backup volume) that contains a small percentage of valid data. This lets you reuse your nearly empty tape volumes. Tape volumes that may have data checks or invalid file block identifiers can also be recycled.

When you issue the generic RECYCLE command, DFSMSHsm moves valid backup versions or valid migration copies from several tapes to one tape volume. The specific RECYCLE command moves valid backup versions or valid migration copies from one tape, or a connected set of tapes, to one tape volume. For example, if you specify RECYCLE SPILL, DFSMSHsm moves the valid data from all eligible tape spill backup volumes to one tape spill backup volume until that spill backup volume becomes full. DFSMSHsm then selects another available tape spill backup volume to receive the valid data sets. If you had specified SETSYS TAPEDELETION(SCRATCHTAPE), the recycled tapes become scratch tapes. If you had specified SETSYS TAPEDELETION(HSMTAPE), the recycled tapes become unassigned tapes.

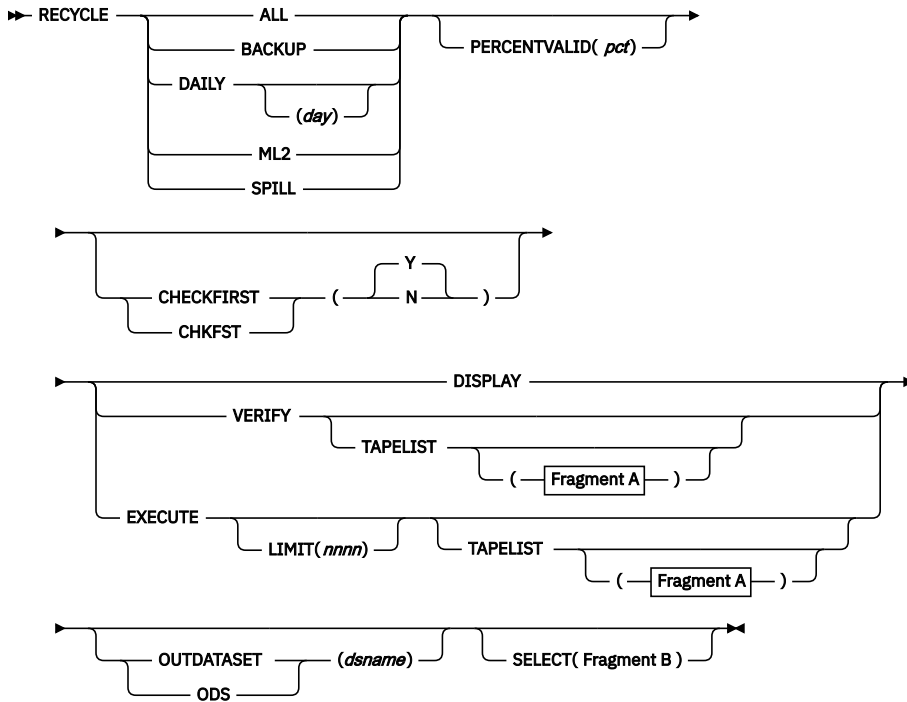
When you recycle a tape backup volume (daily or spill), DFSMSHsm copies the valid data on the tape volume to an available tape spill backup volume. When you recycle a tape migration level 2 volume, DFSMSHsm copies the valid data on the tape volume to an available tape migration level 2 volume.

Although an MVS image can process only one RECYCLE request at a time, RECYCLE initiates up to 15 tape processing tasks when you issue a generic RECYCLE command. Each task that moves data requires two tape drives, one for input and one for output. The SETSYS MAXRECYCLETASKS parameter allows you to specify the number of tape processing tasks that can run at a time.

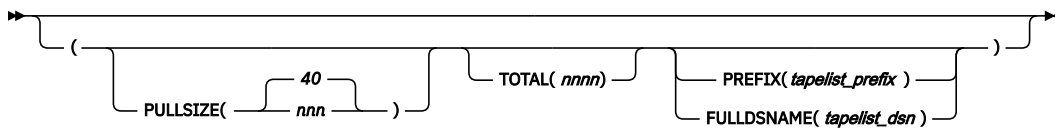
RECYCLE for the same volume type (ML2, DAILY, SPILL) cannot be issued concurrently from two separate hosts. For example, a request to RECYCLE ML2 tapes from a second MVS image fails if the first image is already recycling ML2 tapes.

RECYCLE will only process connected sets that span 254 or fewer volumes.

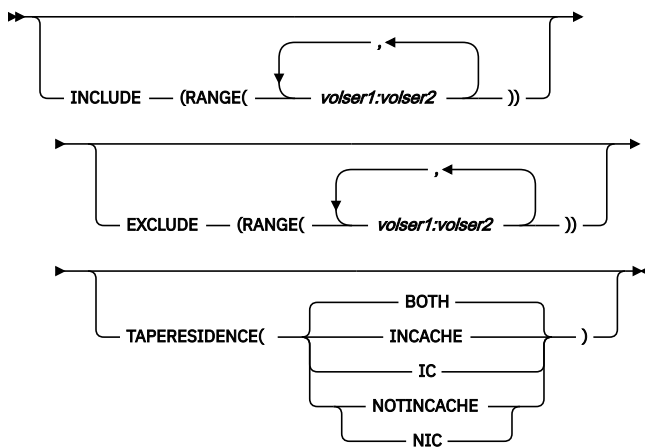
Syntax of the generic RECYCLE command



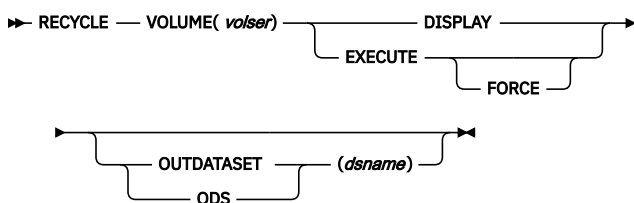
A: TAPELIST optional parameters



B: SELECT optional parameters



Syntax of the specific RECYCLE command



Required parameters of the RECYCLE command

This topic describes the required parameters of the RECYCLE command.

ALL, BACKUP, DAILY, ML2, SPILL and VOLUME: Specifying the tape volumes to be recycled

Explanation: **ALL**, **BACKUP**, **DAILY**, **ML2**, **SPILL**, and **VOLUME(volser)** are mutually exclusive, required parameters specifying the types of tape volumes to be recycled.

ALL specifies that DFSMSHsm should recycle all eligible daily, ML2, and spill tape volumes.

BACKUP specifies that both daily and spill backup tapes are to be used as input in the RECYCLE processing.

DAILY specifies that DFSMSHsm should recycle all eligible tape daily backup volumes.

Subparameter	Explanation
day	DAILY(day) specifies that DFSMSHsm should recycle all eligible tape volumes assigned to that day in the backup cycle. For <i>day</i> , substitute a decimal number to represent the day in the backup cycle assigned to the tape daily backup volume. You use the DEFINE BACKUP command to define the backup cycle.

ML2 specifies that DFSMSHsm should recycle all eligible tape migration level 2 volumes. If the SETSYS ML2PARTIALSNOTASSOCIATEDGOAL(*nnn*) parameter has been specified with a value other than NOLIMIT and the number of ML2 partial tapes not currently associated as output for any migration or recycle task and meeting the recycle criteria is greater than *nnn* then DFSMSHsm recycles the excess over *nnn*, that is, those tapes with the least percent of valid data. A partial is a tape that is not full and not empty.

SPILL specifies that DFSMSHsm should recycle all eligible tape spill backup volumes.

VOLUME(volser) specifies the volume to be recycled. For *volser*, substitute the serial number of the volume to be recycled. When you specify a volume serial number, DFSMSHsm does not check whether the specific tape volume is eligible for recycling. This allows you to recycle tapes that are not full or that were otherwise not selected by generic recycle.

Defaults: None.

Note:

1. If you specify a generic request, a connected set is eligible for recycling only if the following conditions are met for all volumes in the set:
 - The tape volume is a migration level 2 tape or a backup tape, but not an unassigned backup tape.
 - The tape volume is either marked full, or a partial ML2 tape that is not connected to any other tape and not currently associated with any migration or recycle output task.
 - The tape volume is marked full.
 - The percentage of valid data is equal to or less than the applicable value. This percentage is specified with the PERCENTVALID parameter, the SETSYS ML2RECYCLEPERCENT parameter, or the SETSYS RECYCLEPERCENT parameter. When these parameters are not specified, the DFSMSHsm default is 20%.
 - The volume has not previously failed recycle processing, or if it has failed, now contains no valid data.
 - The TTOC for the volume is complete.
 - The volume does not have a disaster alternate volume.
2. Dump volumes are not recycled.

- For performance reasons, DFSMSHsm does not update the OCDS TTOC every time a data set is migrated or backed up to a tape volume. If DFSMSHsm abends, an I/O error occurs, or a system fails, the OCDS TTOC may not be updated. If you specify the RECYCLE command to recycle a tape volume that has missing TTOC entries, message ARC0378I is issued, indicating that the tape media and the OCDS TTOC have a mismatch. Use the AUDIT MEDIACONTROLS FIX command to resolve the problem. For more information, see [“MEDIACONTROLS VOLUMES, MEDIACONTROLS\(SDSP\) VOLUMES and MEDIACONTROLS\(CLOUD\): Requesting an audit of control information”](#) on page 748.

EXECUTE, VERIFY, and DISPLAY: Specifying whether to list or to recycle tape volumes

Explanation: **EXECUTE**, **VERIFY**, and **DISPLAY** are mutually exclusive, required parameters that specify whether to recycle, print tape lists, or list volumes that meet eligibility criteria (percent valid) for recycling.

EXECUTE specifies that DFSMSHsm should recycle a specific tape volume, a category of backup volumes, tape migration level 2 volumes, or a combination of all volumes. Volumes that meet the eligibility criteria are recycled. To recycle volumes that do not meet the eligibility criteria, you must prepare the volumes for recycle by following the specified actions.

VERIFY specifies that DFSMSHsm produces one set of tape volume serial numbers that are listed twice: in two different formats and in one output data set. Customers can use the first list as a **pull list** and the second list as a **mount list**.

- **Pull list:** One or more tape groups of the requested category (ML2, SPILL, and so forth) to be pulled from tape storage. Each group is ordered by volume serial. The groups are in the order in which RECYCLE will most likely request that tapes be mounted when the RECYCLE EXECUTE command is issued. Volumes in a connected set are not split between groups. Empty tapes are included and identified as such, but are not mounted during RECYCLE processing. Pull lists are used by operators to fetch and transport the required tapes during RECYCLE processing.
- **Mount list:** A list of all tapes to be recycled for a given category, ordered by the reuse capacity. This is the order in which recycle most likely requests the tapes be mounted. However, this projected mount order may differ from the actual mount order due to other factors affecting recycle processing including, but not limited to, running more than 1 recycle task, unrecoverable errors, a mixture of device types within the mount list, or connected sets containing more than 5 volumes. The volumes in a connected set are identified and listed together. Empty tapes are included and identified as such, but are not mounted during RECYCLE processing. Mount lists aid tape operators with the mounting order of tapes that are being recycled.

Both of these lists are dynamically allocated to either a SYSOUT data set or to the data set having the prefix that you specified with the TAPELIST PREFIX(*tapelist_prefix*) parameter or to the fully qualified data set name that you specified with the TAPELIST FULLDSNAME(*tapelist_dsn*) parameter. The data set is deallocated when the command completes.

DISPLAY gives you a formatted list in volume serial sequence of the tape volumes eligible for recycling that meet the percent valid criteria. The list is dynamically allocated to either a SYSOUT data set or the data set specified by the OUTDATASET parameter. If you request DISPLAY for a specific tape volume, you will get a list of the connected set containing that volume. DFSMSHsm sends a message containing the information to your terminal or optionally directs this information to a data set specified by the OUTDATASET parameter.

When you specify the VERIFY or DISPLAY parameters, the total output includes one or more of the following volumes:

- Eligible tape daily backup volumes assigned to a particular day (or to all days) in the backup cycle
- Eligible tape spill backup volumes
- Eligible tape migration level 2 volumes

When you specify the VERIFY or DISPLAY parameters, DFSMSHsm does not ask for any tapes to be mounted.

Defaults: None.

Note:

1. If you are a customer without an ATL but with a relatively *small* number of tapes that you must pull manually from shelf storage, use the DISPLAY parameter.
However, if you are a customer without an ATL but with a *large* number of tapes that you must pull manually, the VERIFY parameter (or EXECUTE TAPELIST) is a better choice.
2. With the VERIFY parameter, the number of volumes in a pull group (and the total number of volumes to be pulled) can be specified with the optional TAPELIST parameter.
3. If you specify the VERIFY parameter with a single volume, DFSMSHsm treats the command as if you had specified DISPLAY.

Optional parameters of the RECYCLE command

This topic describes the optional parameters of the RECYCLE command.

CHECKFIRST: Allowing connected set to be recycled sooner

Explanation: **CHECKFIRST** is an optional parameter that allows connected sets to be recycled sooner. In determining whether to recycle a connected set, DFSMSHsm requires the first tape within the connected set to meet the PERCENTVALID specification before continuing on with the calculation for the entire connected set's. You can bypass this first volume check by using the CHECKFIRST(N) parameter of the RECYCLE command. If CHECKFIRST(N) is specified, the entire connected set's percentage of valid data is calculated regardless of the percent of valid data on the first volume.

Notes:

1. If you enter the RECYCLE command with the VOLSER parameter, the system ignores the CHECKFIRST parameter.
2. Be aware that CHECKFIRST(N) causes recycle processing to always determine the percent valid for the entire connected set, which can slow recycle performance if you have many connected sets.

Defaults: If CHECKFIRST is not specified on the RECYCLE command, then recycle performs as if CHECKFIRST(Y) had been specified.

FORCE: Forcing DFSMSHsm to permit a tape recycle

Explanation: **FORCE** is a parameter specifying whether to permit the specified 3480 single-file tape or equivalent to be recycled even though the tape contains data checks or invalid file block identifiers. The FORCE optional parameter is valid only when used with the specific RECYCLE command for a single volume request. The only connected volume set that will be processed, is a connected set containing the specified volume.

When the FORCE parameter is in effect, RECYCLE works the same as when FORCE is not specified, except when a data check or an invalid file block identifier is encountered. DFSMSHsm invalidates the data set being processed before continuing with the next data set on the tape. Migrated data sets are deleted, including the catalog entry. In this case, it is the storage administrator's task to recover the data set from another source. For backup copies, the version with the error is invalidated. Because the records necessary to access the data will be lost when using the FORCE parameter, it should be specified only when no alternate volume previously generated by the TAPECOPY command is available.

When processing tape volumes with known errors, use the following steps:

1. If an alternate tape volume exists, use the tape replace function (see Chapter 69, [“TAPERPL command: Replacing cartridge-type tape volumes with their alternate volumes,”](#) on page 1235) instead of recycling the volume.
2. If you do not have an alternate volume, the following steps are recommended to reduce the exposure of deleting accessible data when an error occurs—such as a hardware malfunction:
 - a. Use the RECYCLE command without the FORCE parameter to move all data without errors to another tape volume.

- b. Now use the RECYCLE command with the FORCE parameter to effectively remove all references to any data that is still inaccessible. After this process, the tape volume is deleted.

For a full discussion of recovery situations, refer to [Chapter 8, “Aggregate backup and recovery support \(ABARS\),”](#) on page 319.

Defaults: None.

Note: If you specify RECYCLE VOLUME FORCE and the TTOC for the volume is incomplete, RECYCLE will not process this volume. You will need to use the AUDIT MEDIACONTROLS function to prepare the volume for RECYCLE. For more information on the recovery procedures that have to be performed to be able to recycle these data sets, refer to [“How to handle inconsistencies between tape media contents and OCDS TTOC records”](#) on page 550.

LIMIT: Quiescing recycle processing when the desired number of tapes have been returned to scratch

Explanation: **LIMIT(*nnnn*)** is an optional parameter specifying when recycle quiesces. RECYCLE quiesces after the net specified number of tapes have been returned to scratch status. Each task completes the connected set currently being processed and no additional ones are started. You can specify the number of scratch tapes needed (*nnnn*) from 1 to 9999. The LIMIT parameter is subject to the number of tapes allowed by the PERCENTVALID parameter; the LIMIT parameter provides a way to stop processing before all the candidates meeting the PERCENTVALID value are processed.

Net is determined by tapes read in for input minus tapes written to for output. For RECYCLE ALL, the LIMIT is applied to ML2 tapes and BACKUP tapes separately. That is, RECYCLE ALL LIMIT(100) results in 100 net ML2 tapes and 100 net BACKUP tapes, unless you do not have enough tapes that meet the eligibility criteria, for example, when the values indicated by the SETSYS RECYCLEPERCENT or the SETSYS ML2RECYCLEPERCENT parameter have been met.

Defaults: If LIMIT(*nnnn*) is not specified on the RECYCLE command, DFSMSHsm processes all tapes eligible for RECYCLE processing.

Note: The LIMIT parameter specifies an upper bound on the net number of volumes to be returned to scratch status in a given category. If the TAPELIST(TOTAL) parameter is also specified, the TOTAL parameter specifies an upper bound on the number of volumes to be recycled in a given category. In each case, DFSMSHsm treats the upper bounds as approximate, since a connected set is recycled completely or not at all. If the value specified with the TOTAL parameter is too small, the value specified with the LIMIT parameter may not be reached.

OUTDATASET: Directing recycle messages

Explanation: **OUTDATASET(*dsn*)** is an optional parameter specifying where to send the messages generated during RECYCLE processing. For (*dsn*), substitute the fully qualified name of the data set to receive the RECYCLE messages. The recycle command log records all data movement activity that results from processing the RECYCLE command. Any error messages that the RECYCLE command processor issues are also written in this log.

If the data set does not exist, DFSMSHsm dynamically allocates and catalogs an output data set with the following characteristics:

- Data set name specified (*dsname*)
- Record format of fixed-blocked with ANSI control characters (FBA)
- Logical record length of 121
- Data set is system reblockable
- Primary allocation of 20 tracks (see note)
- Secondary allocation of 50 tracks (see note)
- Unit of SYSALLDA. (see note)

If the data set already exists, DFSMSHsm will use the data set. The data set must have the following characteristics.

- The data set must be cataloged and on DASD.
- The data set record format must be FBA, and the logical record length must be 121.
- The data set is system reblockable
- The user can choose the primary space allocation.
- If the data set does not contain data, DFSMSHsm starts writing output data at the beginning of the data set.
- If the data set contains data, DFSMSHsm writes the output data after the existing data.

Defaults: If OUTDATASET(*dsn*) is not specified on the RECYCLE command, DFSMSHsm sends all messages generated during RECYCLE processing to the SYSOUT specified by the SETSYS parameter.

Note: If you select the OUTDATASET option, you can use the PATCH command to change the unit name, primary allocation, and secondary allocation. If you select the SYSOUT option, you can use the PATCH command to change whether, and how, DFSMSHsm limits the lines of SYSOUT output. Refer to the [z/OS DFSMSHsm Implementation and Customization Guide](#) for a list of the available PATCH commands.

PERCENTVALID: Specifying the percent valid criteria for recycle eligibility

Explanation: PERCENTVALID(*pct*) is an optional parameter specifying the maximum percentage of valid data that a volume can have and still be eligible for recycling. DFSMSHsm ignores this parameter if you specified the volume to be recycled. For *pct*, substitute a decimal number from 0 to 100. For example, if you specify *pct* as 30, a connected set (including a single volume) whose first volume in the set contains 30% or less valid data AND whose entire set contains 30% or less valid data is eligible for recycling.

Note: If 0 (zero) is specified, no tape drives are required. Only tapes that contain no valid data are returned to scratch.

By default, DFSMSHsm requires the first volume in the connected set to meet the percent valid criterion before determining the connected set's percentage of valid data. To allow connected sets to be recycled earlier and potentially free up more volumes to scratch or tape pools, you can use the CHECKFIRST(N) parameter to specify that the first volume of a connected set need not meet the PERCENTVALID requirement as long as the entire connected set meets the PERCENTVALID requirement. For example:

```
RECYCLE PERCENTVALID(pct) CHECKFIRST(N)
```

Defaults: If PERCENTVALID(*pct*) is not specified on the RECYCLE command, DFSMSHsm selects an appropriate value based on the volume type parameter specified. Also, recycle processing always performs the first volume check when evaluating the connected set's recycle eligibility.

RECYCLE ALL is treated functionally as a RECYCLE ML2 followed by a RECYCLE BACKUP.

When processing ML2 volumes, DFSMSHsm uses the ML2RECYCLEPERCENT value that has been specified on the SETSYS command. If ML2RECYCLEPERCENT has not been specified, DFSMSHsm uses the SETSYS RECYCLEPERCENT value. If RECYCLEPERCENT has not been specified, the DFSMSHsm default is 20%.

For all backup volume type parameter specifications (Backup, Daily, or Spill), DFSMSHsm uses the RECYCLEPERCENT value.

You can override this default by specifying the RECYCLE PERCENTVALID parameter.

Note:

1. During RECYCLE execution, a DELVOL operation is performed for any volume having 0% valid data.
2. For a discussion of how DFSMSHsm calculates percentages based on valid blocks and connected sets, refer to [Chapter 13, "Managing DFSMSHsm media,"](#) on page 527.

SELECT: Specifying tape eligibility for generic recycle

Explanation: **SELECT** is an optional parameter that is used to specify a selection filter for the eligibility test performed by the RECYCLE function. The SELECT parameter provides increased granularity to filter out tapes that are considered eligible by the percent valid criteria but do not meet the range that is specified with the SELECT parameter.

When the SELECT parameter is specified, you must specify **one** or **both** of the following SELECT sub-parameters:

Subparameter	Explanation
INCLUDE	INCLUDE(RANGE (volser1:volser2 ...)) specifies that this range of tape volumes is included in the RECYCLE eligibility tests. INCLUDE(RANGE(volser1:volser2 ...)) is a required sub-parameter of the SELECT parameter.
EXCLUDE	EXCLUDE(RANGE (volser1:volser2,...)) specifies that this range of tape volumes is not included in the RECYCLE eligibility tests. EXCLUDE(RANGE(volser1:volser2,...)) is a required sub-parameter of the SELECT parameter.
RANGE	RANGE (volser1:volser2,...) specifies a single volser range, a list of volser ranges, a single volume, or a list of volumes. When you specify a volser range, <i>volser1</i> is the beginning of the range, and <i>volser2</i> is the end of the range. When you specify a single volume, use that volume serial number for both <i>volser1</i> and <i>volser2</i> . The only limit on the number of volumes or ranges used for RANGE is the RECYCLE command's limit of 1023 characters for a single command. All tape volumes within a specified range are included. If you are including a connected set of tapes, all tape volumes in the connected set must be within the specified range of volumes to be considered as eligible. If any tape volume within a connected set is specified in the excluded range, then that whole connected set is excluded. Message ARC0849I is issued when an ending volume serial is specified before a starting volume serial.
TAPERESIDENCE	TAPERESIDENCE(BOTH INCACHE NOTINCACHE) only applies to IBM virtual tape server volumes. The TAPERESIDENCE parameters are ignored when recycling non-VTS tape volumes. IBM VTS tape volumes that are eligible for recycle and reside either in cache, not in cache, or both in cache and not in cache as specified via the select parameter will be included in recycle processing.

Note: The SELECT parameter applies only to generic RECYCLE processing. The SELECT parameter is not valid if specified with the RECYCLE VOLUME command.

Defaults:

- If SELECT is not specified on the RECYCLE command, DFSMSHsm uses the percentage specified with the PERCENTVALID parameter to determine RECYCLE eligibility.
- When recycling tape residing in an IBM VTS and the SELECT(TAPERESIDENCE(BOTH | INCACHE | NOTINCACHE)) parameters are not specified on the RECYCLE command, DFSMSHsm will use the SETSYS RECYCLETAPERESIDENCE setting as the default.

TAPELIST: Listing volumes for pulling and mounting tapes

Explanation: **TAPELIST** is an optional parameter that can be used with the VERIFY or EXECUTE parameters (it is ignored if used with the DISPLAY parameter). When TAPELIST is used with the VERIFY parameter, it allows you to review the output while other RECYCLE commands can continue processing. When TAPELIST is used with the EXECUTE parameter, DFSMSHsm produces the same lists as those

produced by the VERIFY parameter, plus, you are given the option to continue the RECYCLE processing after the lists are produced.

The lists are dynamically allocated to either a SYSOUT data set or to the data set generated or specified by the TAPELIST PREFIX parameter or the TAPELIST FULLDSNAME parameter.

Subparameter	Explanation
PULLSIZE	PULLSIZE(<i>nnn</i>) allows you to select the number of tape volumes that is feasible to physically transport at one time from shelf storage to your tape drives (for example, on a cart) or the number of tape volumes that can be managed by an operator after they are brought to the vicinity of the tape drives. Specify a value between 1 and 999.
TOTAL	TOTAL(<i>nnnn</i>) allows you to select the total number of volumes (including empty volumes) to be listed in all pull groups for a given category of DFSMSHsm tapes. If the EXECUTE parameter is specified, this restricts the total number of volumes that can be recycled for that category. Specify a value between 1 and 9999.
PREFIX	<p>PREFIX(<i>tapelist_prefix</i>) is mutually exclusive with the FULLDSNAME parameter. This optional parameter is used to specify the name of the TAPELIST output.</p> <p>You can select the data set prefix to be used for that data set name. DFSMSHsm generates a fully qualified data set name by appending the following qualifiers to <i>tapelist_prefix</i>:</p> <ul style="list-style-type: none"> • .BACKUP, .DAILY, .ML2, or .SPILL, depending on the category of tape volumes specified on the command • date stamp .Dyyyyddd, where <i>yyyy</i> represents the year and <i>ddd</i> the day of the year • time stamp .Thhmmss, where <i>hh</i>, <i>mm</i>, and <i>ss</i> represent the hour, minute, and second <p>Because the maximum length of a data set name is 44 characters, the allowable length of <i>tapelist_prefix</i> is 20 characters, so that the generated data set name will be valid regardless of the volume category specified.</p> <p>When the PREFIX parameter is specified, DFSMSHsm dynamically allocates and catalogs an output data set with the following characteristics:</p> <ul style="list-style-type: none"> • Data set name generated from <i>tapelist_prefix</i> • Record format of FBA • Logical record length of 121 • Data set is system reblockable • Primary allocation of 20 tracks • Secondary allocation of 50 tracks • Unit of SYSALLDA

Subparameter	Explanation
FULLDSNAME	<p>FULLDSNAME(<i>tapelist_dsn</i>) is mutually exclusive with the PREFIX parameter. This optional parameter is used to specify the name of the TAPELIST output.</p> <p>You can select the fully qualified name to be used for the TAPELIST output. If <i>tapelist_dsn</i> data set does not exist, DFSMSHsm dynamically allocates and catalogs an output data set with the following characteristics:</p> <ul style="list-style-type: none"> • Data set name of <i>tapelist_dsn</i> • Record format of FBA • Logical record length of 121 • Data set is system reblockable • Primary allocation of 20 tracks • Secondary allocation of 50 tracks • Unit of SYSALLDA <p>However, if <i>tapelist_dsn</i> does exist, this output data set must be available so that DFSMSHsm can use it for output. This output data set must have the following characteristics:</p> <ul style="list-style-type: none"> • Cataloged and reside on DASD • Record format of FBA • Logical record length of 121 • Data set is system reblockable • Primary allocation of the user's choice • If the data set does not contain data, DFSMSHsm starts writing output data at the beginning of the data set. • If the data set contains data, DFSMSHsm writes the output data after the existing data.

When DFSMSHsm completes writing to the data set, DFSMSHsm deallocates the data set allowing you to browse or print it. If you have specified the EXECUTE parameter, then before DFSMSHsm allocates any tape drives, it issues a WTOR message to the tape operator:

```
ARC0825D RECYCLE TAPE LIST CREATED, DSN=dsname. DO YOU WISH
TO CONTINUE? REPLY 'N' TO STOP RECYCLE OR 'Y' WHEN READY TO
MOUNT TAPES.
```

Replying Y causes recycle processing to select and allocate the first input tapes for as many recycle tasks as are defined. Replying 'N' ends recycle processing for that category of volumes.

Defaults

- If the PULLSIZE parameter is not specified, DFSMSHsm uses the value of 40.
- When the TOTAL parameter is not specified, DFSMSHsm lists pull groups for all the eligible volumes in the category being recycled and if the EXECUTE parameter is specified, recycles all of them.
- When the PREFIX parameter and the FULLDSNAME parameter are not specified, DFSMSHsm writes the TAPELIST output to the SYSOUT class specified by the SETSYS parameter.

Note:

1. If you specify the RECYCLE ALL command and do not specify the FULLDSNAME parameter, DFSMSHsm creates two data sets: one for ML2 volumes and one for backup (daily and spill) volumes. If you also specify the EXECUTE parameter, DFSMSHsm writes the ARC0825D message twice, following the deallocation of each of the two data sets. Your response need not be the same to each

message. If the operator's response is to not continue with ML2 volumes, DFSMSHsm later writes the ARC0825D message with the second data set name, allowing you to recycle backup volumes.

2. Because the volumes in a connected set are not split across pull groups, some pull groups may have fewer volumes than the PULLSIZE value.
3. Empty tapes are not mixed with nonempty tapes within a pull group.
4. If a connected set contains more volumes than the PULLSIZE value, that connected set is listed in a pull group by itself.
5. The TOTAL parameter specifies an upper bound on the number of volumes to be in the list and therefore recycled in a given category. The LIMIT parameter specifies an upper bound on the net number of volumes that can be returned to scratch status in a given category. In each case, the upper bounds are approximate, since a connected set is recycled completely or not at all. If the TOTAL value is too small, the LIMIT value may not be reached.
6. If you specify the RECYCLE ALL command, the total number of volumes pulled can be as much as twice the TOTAL value, one set of migration level 2 volumes and one set of backup volumes.
7. If you specify the RECYCLE ALL command with the TAPELIST(FULLDSNAME) parameter, DFSMSHsm uses this specified data set for the TAPELIST of migration level 2 volumes followed by the lists of backup volumes. You should realize that the backup pass may fail if you have this data set in use after the L2 TAPELIST output is written and do not release it before the backup pass starts. Therefore, you should not use ALL and FULLDSNAME together unless you can be sure that the data set will be freed by the time RECYCLE processing requires it.
8. If you specify the TAPELIST parameter with the DISPLAY parameter or when specifying a single volume, DFSMSHsm ignores the TAPELIST parameter.
9. If all of your tape volumes are in automated tape libraries, the output from the TAPELIST or VERIFY parameters will benefit you as a reporting tool.
10. You can use the PATCH command to change the unit name, primary allocation, and secondary allocation. Refer to the *z/OS DFSMSHsm Implementation and Customization Guide* for a list of the available PATCH commands.

Examples of how to code the RECYCLE command

The following examples present different ways to code the RECYCLE command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Recycling all volumes, do not check the first volume in a connected set for percent valid criteria

In this example, this command recycles all eligible backup and migration volumes. It also requests that DFSMSHsm not check that the first volume in a connected set meet the percent valid criteria as long as the average percent valid for all volumes in the connected set meets the percent valid criteria.

```
RECYCLE ALL EXECUTE CHECKFIRST(N) PERCENTVALID(25)
RECYCLE PERCENTVALID(25) CHECKFIRST(N)
```

Displaying a list of all eligible tape daily backup volumes

In this example, a formatted list is printed of all tape daily backup volumes that contain 5% or less valid data and are eligible for recycling. Volumes that have 0% valid data are not listed because they do not need to be pulled for a subsequent recycle.

```
RECYCLE DISPLAY DAILY PERCENTVALID(5)
```


Recycling tape spill volumes

In this example, DFSMSHsm recycles all tape spill backup volumes that contain 25% or less valid data and are eligible for recycling.

```
RECYCLE EXECUTE SPILL PERCENTVALID(25)
```

Recycling tape migration level 2 volumes

In this example, DFSMSHsm recycles all tape migration level 2 volumes that are eligible for recycling. DFSMSHsm uses the value specified with the SETSYS ML2RECYCLEPERCENT command to determine the percent valid criteria. If the SETSYS ML2RECYCLEPERCENT is not specified, DFSMSHsm uses the value specified with the SETSYS RECYCLEPERCENT command.

```
RECYCLE EXECUTE ML2
```

Recycling a specific tape volume

In this example, a specific tape volume is recycled. DFSMSHsm does not check the percentage of valid data on the volume or whether the volume is full before it recycles the volume.

```
RECYCLE EXECUTE VOLUME(BATP01)
```

Recycling a tape volume with known errors

In this example, a specific tape volume with known errors is recycled.

```
RECYCLE VOLUME(TAP230) FORCE EXECUTE
```

Displaying a list of both daily and spill backup tapes

In this example, a list of both daily and spill backup tapes that are already empty but have not yet been returned to scratch is displayed.

```
RECYCLE BACKUP DISPLAY PERCENTVALID(0)
```

Recycling a limited number of backup tapes

In this example, DFSMSHsm recycles both daily and spill backup tapes and it quiesces the RECYCLE process when 150 tapes have been returned to scratch. The messages generated during the RECYCLE processing are written to the requested output data set.

```
RECYCLE BACKUP EXECUTE LIMIT(150) OUTDATASET(C253800.RECYCLE.OUTPUT)
```

Producing a list of tape volume serials before recycling those volumes

In this example, DFSMSHsm produces a list of tape volume serials in a pull list format and a mount list format. An operator can transport 100 tapes on each cart and a total of 500 tapes is requested. The list of tape volume serials is written to the output data set prefixed by the requested data set name. After the tapes have been collected, the tape operator can reply "Y" to the ARC0825D message. Then RECYCLE begins to process those tape volumes.

```
RECYCLE BACKUP EXECUTE TAPELIST(PULLSIZE(100) TOTAL(500) PREFIX(REMOTE))
```


Another example of producing a list of tape volume serials before recycling those volumes

In this example, DFSMSHsm produces a list of tape volume serials in both pull list format and mount list format. The PULLSIZE default of 40 is used. All connected sets whose percent valid is not greater than 75% are listed. The list is written to the output data set with a prefix of 'A'.

Note: This example is used to illustrate outputs. We would not expect you to use this command with a PERCENTVALID percentage of 75%.

```
RECYCLE ALL EXECUTE TAPELIST(PREFIX(A)) PERCENTVALID(75)
```

The following is a sample listing of the output data set. For this example, the pull list and the mount list for only ML2 volumes are shown.

```
OUTPUT LIST DATASET: A.ML2.D1996109.T093226
LEGEND:
'*' INDICATES EMPTY TAPE, '-' INDICATES CONTINUATION FROM OR TO
DFSMSHSM RECYCLE PULL LIST FROM 0.0% TO 0.0%
*555060 *555075 *555100 *555222 *555990 *555995

DFSMSHSM RECYCLE PULL LIST FROM 0.8% TO 49.9%
555001 555003 555004 555005 555006 555007 555008 555009
555011 555012 555013 555014 555015 555016 555017 555018
555019 555022 555026 555027 555028 555029 555031 555032
555034 555035 555036 555037 555038 555040 555041 555042
555043 555044 555045 555046 555047 555048 555049 555050

DFSMSHSM RECYCLE PULL LIST FROM 51.4% TO 57.3%
555021 555023 555024 555039 555115 555116 555117 555118
555119 555120
END OF DFSMSHSM RECYCLE PULL LIST
```

Note:

1. The volumes listed in the 0.0% to 0.0% section are empty and will not be mounted
2. The volumes listed in the 0.8% to 49.9% section represent the pull default size of 40 volumes. The volumes are listed in collating sequence to simplify locating them in tape storage.

```
DFSMSHSM RECYCLE MOUNT LIST FROM 0.0% TO 57.3%
*555060 *555075 *555100 *555222 *555990 *555995 555011 555007
555004 555032 555012 555035 555048 555015 555001 555044
555042 555045 555008 555034- -555029 555009 555028 555038
555040 555018 555006 555049 555005 555041 555046 555037
555026 555036 555047 555016 555022 555017 555013 555050
555031 555019 555003 555027 555014 555043 555024 555039
555119- -555120- -555118- -555117- -555116- -555115 555023 555021
END OF DFSMSHSM RECYCLE MOUNT LIST
```

Note:

1. The volume 555015 needs a different device type than the other volumes. It will not be requested after volume 555048, instead volume 555015 will be requested at a later time, when a new drive allocation is performed.
2. When attempting to recycle connected set 555034–555029, an open error occurred on 555034 causing the connected set to fail recycle processing. A mount for 555029 will not be attempted.
3. The connected set 555119–555120–555118–555117–555116–555115 consisted of more than 5 volumes. This connected set will not be mounted on an already mounted drive, independent of the SETSYS RECYCLEINPUTDEALLOCFREQUENCY setting, but will be skipped until a new drive allocation is performed.

Converting current media tapes to a newer tape media

In this example, the SELECT parameter is used to specify a selection filter for RECYCLE processing. For this example, assume the current media to be 3490 tapes and the newer media to be 3590 tapes. Also

assume that the 3490 tapes all have volsers in the range of A00000 through A39999 and 3590 tapes all have volsers in the range of B00000 through B39999.

You can use the INCLUDE parameter to specify a range of tapes to RECYCLE that are 3490 tape media by entering the following command:

```
RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(A00000:A39999))) PERCENTVALID(100)
```

Or, you can use the EXCLUDE parameter to specify a range of tapes to exclude from RECYCLE processing that are 3590 tape media by entering the following command:

```
RECYCLE ALL EXECUTE SELECT(EXCLUDE(RANGE(B00000:B39999))) PERCENTVALID(100)
```

Selecting connected tapes for RECYCLE using the SELECT parameter

In this example, three ways to use the SELECT parameter when selecting connected sets for RECYCLE processing are shown. For this example, assume that there are only three connected sets:

- Set A consists of TAPE01 and TAPE09
- Set B consists of TAPE02 and TAPE04
- Set C consists of TAPE03, TAPE05, and TAPE10

In this example, you want to recycle only sets A and C. To recycle connected sets A and C using the SELECT INCLUDE parameters, enter the following command:

```
RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(TAPE01:TAPE03,TAPE05:TAPE10)))
```

Although TAPE02 from set B is included in the range, TAPE04 is not included in the range and this makes set B ineligible for RECYCLE processing.

To recycle connected sets A and C using the SELECT EXCLUDE parameters, enter the following command:

```
RECYCLE ALL EXECUTE SELECT(EXCLUDE(RANGE(TAPE02:TAPE02)))
```

Because TAPE02 is specified as the range to exclude and because TAPE02 is part of the connected set B, set B is excluded from RECYCLE processing.

To recycle connected sets A and C using both the SELECT INCLUDE and EXCLUDE parameters, enter the following command:

```
RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(TAPE01:TAPE10)) +  
EXCLUDE(RANGE(TAPE04:TAPE04)))
```

Connected set B is not eligible for RECYCLE processing because TAPE04 is specifically excluded.

Selecting single tapes for RECYCLE using the SELECT parameter

In this example, the SELECT parameter is used to select a list of single volumes for RECYCLE processing. To recycle volumes A, J, and Q, enter the following command:

```
RECYCLE ALL EXECUTE SELECT(INCLUDE(RANGE(A:A,J:J,Q:Q)))
```

Chapter 62. RELEASE command: Releasing all or some DFSMSHsm functions for processing

The RELEASE command releases for processing all or part of the DFSMSHsm functions that have been held. You cannot release the processing of a subfunction while the processing of its main function is being held. For example, you cannot release the subfunction BACKUP(AUTO) if the main function BACKUP is held. When you release a main function, the held main function processing and its associated subfunctions processing are released. Commands that failed after you issued the HOLD command are not restarted.

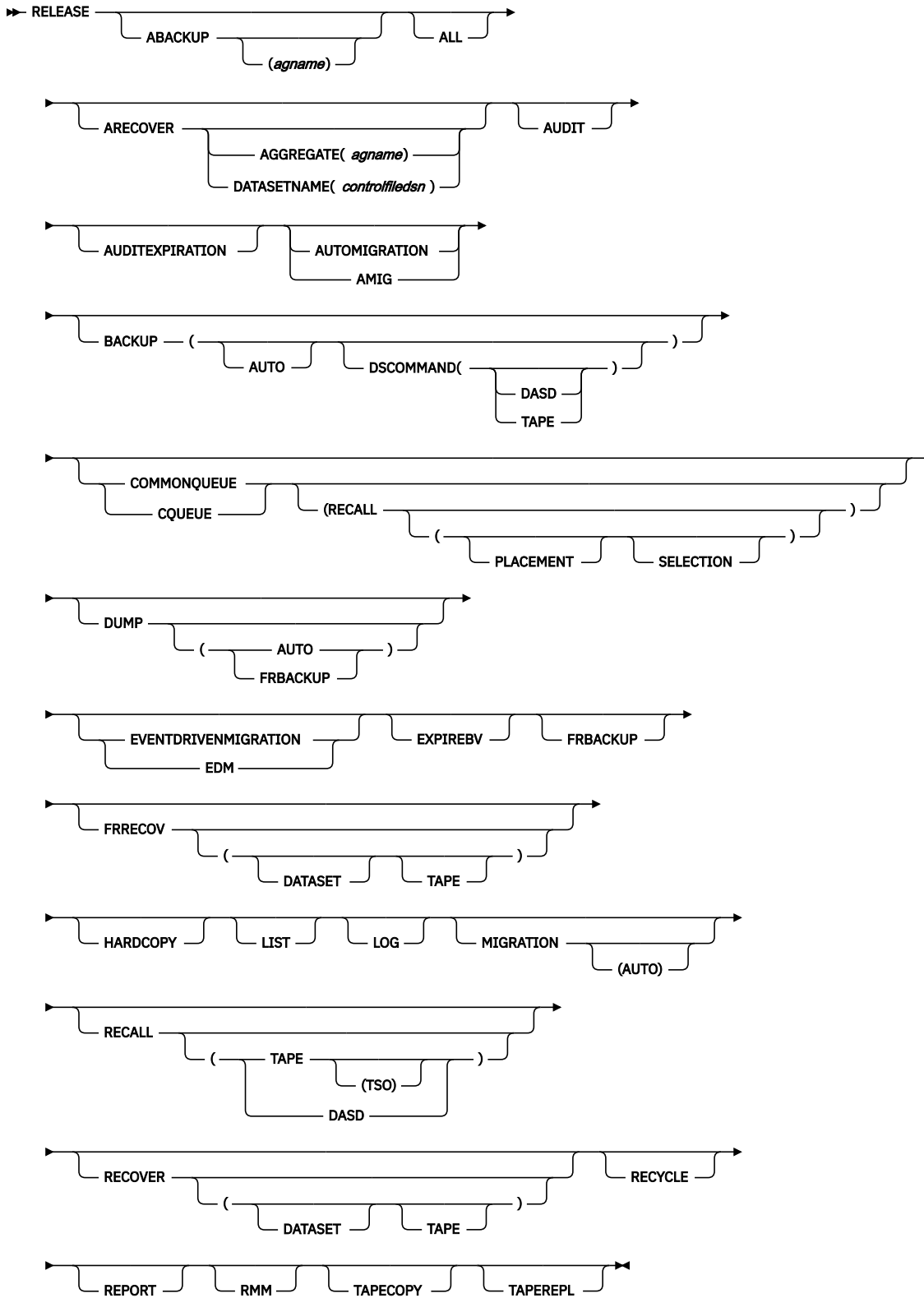
If a journal becomes full, processing is held for all migration, backup, and dump functions. To resume journaling and other DFSMSHsm functions, enter the BACKVOL CDS command to null the journal. If multiple systems share the journal, enter the BACKVOL CDS command on one of the systems, and then enter the RELEASE ALL command on all of the systems that share the journal.

If DFSMSHsm functions are held because of a full journal, the RELEASE command will not appear to be effective until the control data sets are backed up. For more information, see [Chapter 20, “Data recovery scenarios,”](#) on page 629.

The RELEASE command allows migration target levels (ML1 or ML2 tape) that have become unavailable as targets because of some error condition to become available for retry. For more information, see [Chapter 13, “Managing DFSMSHsm media,”](#) on page 527.

In addition, you can also decide to print the activity logs or you can enable the EDGTVEXT interface again after it was disabled because of an ABEND in DFSMSrmm.

Syntax of the RELEASE command



Note:

1. Although the RELEASE command has no required parameters, you must specify at least one parameter to release a DFSMSHsm function for processing.

2. In a multiple address space for DFSMSHsm environment, ABARS commands, like RELEASE ABACKUP or RELEASE ARECOVER, can only be issued by the host identified as the main host. DFSMSHsm ignores the ABARS commands if directed to a host identified as HOSTMODE=AUX.

Optional parameters of the RELEASE command

This topic describes the optional parameters of the RELEASE command.

ABACKUP: Releasing aggregate backup

Explanation

ABACKUP(*agname*) is an optional parameter specifying that DFSMSHsm release for processing any ABACKUP commands currently held in queues and allow new ABACKUP commands to be received. The variable is explained:

agname

Releases any ABACKUP commands held for a particular aggregate group.

Defaults

None.

ALL: Releasing all DFSMSHsm functions

Explanation

ALL is an optional parameter specifying that DFSMSHsm release all functions controlled by this command, except writing in the DFSMSHsm log and printing the activity logs.

Defaults

None.

Usage notes:

1. If you want DFSMSHsm to begin recording in the DFSMSHsm log, you must specify the LOG parameter. If you want DFSMSHsm to print the activity logs, you must specify the HARDCOPY parameter.
2. RELEASE ALL does not release holds on the common recall queue (CRQ). To release a hold on the CRQ, specify RELEASE COMMONQUEUE. To release a hold on only specific CRQ functions, specify the appropriate optional sub-parameters.
3. RELEASE ALL will release AUTOEXPIRATION.

ARECOVER: Releasing aggregate recovery

Explanation

ARECOVER is an optional parameter specifying that DFSMSHsm release for processing any ARECOVER commands currently held in queues and allow new ARECOVER commands to be received. The parameters of the ARECOVER parameter are explained:

AGGREGATE (*agname*)

Releases any ARECOVER commands held for a particular aggregate group.

DATASETNAME (*controlfiledsn*)

Releases any ARECOVER commands held for a particular control file.

Defaults

None.

AUDIT: Releasing the audit function

Explanation

AUDIT is an optional parameter specifying that the audit function is released. DFSMSHsm starts processing any queued AUDIT commands. Any AUDIT command that was interrupted by a HOLD command does not restart.

Defaults

None.

AUTOEXPIRATION: Releasing automatic expiration

Explanation

AUTOEXPIRATION is an optional parameter specifying that DFSMSHsm release a hold on automatic expiration of data sets. Automatic expiration will take place during the next run of a DFSMSHsm space management function.

Defaults

None.

Note: Short hand for the command is 'AEXP'.

AUTOMIGRATION: Releasing only automatic volume and automatic secondary space management

Explanation

AUTOMIGRATION is an optional parameter specifying that DFSMSHsm release automatic volume and automatic secondary space management.

If tape migration has ended while DASD migration continues, or if DASD migration has ended while tape migration continues, this parameter allows both DASD and tape to resume their acceptability as migration targets.

Defaults

None.

Usage notes:

1. You cannot release the AUTOMIGRATION if MIGRATION is held.
2. You can not release the sub-function AUTOEXPIRATION through the RELEASE AUTOMIGRATION command. You must use RELEASE AUTOEXPIRATION (RELEASE AEXP).

BACKUP: Releasing backup

Explanation

BACKUP is an optional parameter specifying whether DFSMSHsm should release all backup processing or release only:

- All data set backup requests
- All data set backup requests to either DASD or tape
- Data set backups to DASD
- Data set backups to tape
- All automatic backups.

BACKUP by itself specifies that DFSMSHsm releases all backup processing. The parameters of the BACKUP parameter are explained:

AUTO

Releases automatic backup processing only.

DSCOMMAND

Releases data set backups for both tape and ML1 DASD requests.

DASD

Releases data set backups directed to ML1 DASD

TAPE

Releases data set backups directed to tape.

Defaults

None.

Usage notes:

1. You cannot release a subfunction if the initial HOLD was done at a higher level. For example, you cannot release the subfunction BACKUP(AUTO) if the main function BACKUP is held.

For another example: If you issue HOLD BACKUP(DSCOMMAND(TAPE)), DFSMSHsm holds only the data set backup by command to tape subfunction. Then, if you issue RELEASE BACKUP(DSCOMMAND), DFSMSHsm releases data set backup by command to tape because TAPE is a subfunction of DSCOMMAND. However, if you issue HOLD BACKUP, DFSMSHsm holds all backup processing. Then, if you issue RELEASE BACKUP(DSCOMMAND(TAPE)), DFSMSHsm *still* holds data set backup by command to tape because all backup to tape was held at the BACKUP level.
2. Releasing a particular function releases the hold on that function's subfunctions, as well.
3. If the time is still within the backup start window, automatic backup starts processing any functions that have not already been processed. Any volumes that were not already fully processed are candidates for processing if less than 24 hours has passed since automatic backup was last started from the beginning and if the time is still within the backup window.
4. If automatic backup is not scheduled to run on a particular day, you can use the BACKUP parameter of the RELEASE command to move backup versions (and backup migrated data sets). Automatic backup is scheduled not to run if you specified SETSYS AUTOBACKUPSTART(0) or specified *N* for this day in the backup cycle.

COMMONQUEUE: Releasing holds on the common queue

Explanation

COMMONQUEUE is an optional parameter that specifies that DFSMSHsm release all holds at the COMMONQUEUE, RECALL, PLACEMENT, and SELECTION levels. All unselected recall requests in the local queue move to the common recall queue (CRQ). The parameters of the COMMONQUEUE parameter are explained:

RECALL

Specifies that if there is no hold at the COMMONQUEUE level, then release a hold at the RECALL, PLACEMENT, and SELECTION levels. All unselected recall requests in the local queue move to the common recall queue (CRQ).

PLACEMENT

Specifies that if there is no hold at the COMMONQUEUE level or RECALL level, then release a hold at the PLACEMENT level. All unselected recall requests in the local queue move to the common recall queue (CRQ).

SELECTION

Specifies that if there is no hold at the COMMONQUEUE level or RECALL level, then release a hold at the SELECTION level.

Defaults

None.

Usage notes:

1. For an overview of how the CRQ operates after a HOLD or RELEASE command is issued, see [Chapter 4, “Other space management considerations,” on page 127.](#)

DUMP: Releasing dump**Explanation**

DUMP is an optional parameter specifying whether DFSMSHsm should release both automatic dump and command dump or only automatic dump.

The parameters of the DUMP parameter are explained:

AUTO

Releases automatic dump processing only.

FRBACKUP

Releases the dump processing of FRBACKUP target volumes.

This command will have the same effect if the system issuing the command is connected to a dump group in the HSMplex. Dump tasks from this system will now be able to be used by the group master scheduler system for dump processing.

The RELEASE DUMP command allows dump processing.

If any variant of RELEASE for dump is issued on a host running a CDQ, that host is eligible to be selected for CDQ volume dump processing if the following conditions are met:

- No other HOLD for DUMP is in place for the host.
- SETSYS MAXDUMPTASKS(x) is specified with a value greater than 0.
- If the host is the CDQ MS, it is eligible to perform MS assignment responsibilities for the dump function.

Defaults

None.

Usage notes:

1. You cannot release the subfunction DUMP(AUTO) if the main function DUMP is held.
2. If the time is still within the dump window, automatic dump starts to process any functions that had not been completed when the dump function was held.

EVENTDRIVENMIGRATION: Releasing Interval and On-demand Migration

Explanation: **EVENTDRIVENMIGRATION** is an optional parameter specifying that DFSMSHsm should release both interval Migration and On-demand Migration.

Defaults: None.

Note: When you issue the RELEASE command, Interval or On-demand migration will not be started automatically. They will start at the next time interval or volume threshold event.

Note: RELEASE EVENDRIVENMIGRATION will only cause Interval and/or On-demand migration functions to be released. Other automatic migration functions may still be held.

EXPIREBV: Releasing the expire backup versions function

Explanation

EXPIREBV is an optional parameter specifying that DFSMSHsm release the expire backup versions function. DFSMSHsm starts processing any queued EXPIREBV commands.

Entering a RELEASE EXPIREBV command does not restart an EXPIREBV command that was in process at the time a HOLD EXPIREBV command was issued. To continue processing, you must reenter the EXPIREBV command. If the function is held when the command is reentered, the command will be saved by DFSMSHsm and processed once the function is released.

Defaults

None.

FRBACKUP: Releasing the fast replication backup or dump function

Explanation

FRBACKUP is an optional parameter that you specify to release the fast replication backup or dump functions. When you issue this command, DFSMSHsm starts processing any queued FRBACKUP, FRBACKUP DUMP, or FRBACKUP DUMPONLY commands.

This command will have the same effect if the system issuing the command is connected to a dump group in the HSMplex. Dump tasks from this system will now be able to be used by the group master scheduler system for FRBACKUP processing.

Defaults

None.

FRRECOV: Releasing the fast replication recover function

Explanation

FRRECOV is an optional parameter that you specify to release the fast replication recovery function. When you issue this command, DFSMSHsm starts processing any queued FRRECOV commands.

The parameters of the FRRECOV parameter are explained:

TAPE

Releases the recovery processing of FRBACKUP generated dump volumes.

DATASET

Is an optional parameter for HOLD FRRECOV. Data set recoveries are held when HOLD ALL, HOLD FRRECOV or HOLD FRRECOV(DATASET) is specified. The existing TAPE keyword will hold data set recoveries from tape.

The RELEASE FRRECOV command has the same effect when the system that issues the command is connected to a CVQ. However, this host volume recover tasks in the CVQ are used by only the master scheduler host when the following conditions are met:

- No other HOLD RECOVER is in place for the host.

- SETSYS MAXDUMPRECOVERTASKS(x) is specified with a value greater than 0.

If the host is the CVQ master scheduler, it is also eligible to perform MS assignment responsibilities for fast replication recovery requests.

Defaults

None.

HARDCOPY: Releasing the activity logs

Explanation

HARDCOPY is an optional parameter specifying that DFSMSHsm print four activity logs that have data. Printing occurs when ACTLOGTYPE is specified as SYSOUT on the SETSYS command.

If the activity logs are allocated to DASD, printing does not occur, but the current logs are closed and new copies of the logs are allocated for use. Allocating the activity logs to DASD allows you to view or browse the currently allocated activity logs for messages that may have passed too quickly from your screens.

The following activity logs can be printed with the RELEASE HARDCOPY command:

- Backup
- Dump
- Migration
- Command

After printing (or closing) these logs, DFSMSHsm allocates new activity logs.

Defaults

None.

Usage notes:

1. The activity log for Aggregate Backup and Recovery cannot be printed with the RELEASE HARDCOPY command. This activity log is written to either a SYSOUT or DASD data set as determined by the ABARSACTLOGTYPE parameter of the SETSYS command.

LIST: Releasing the list function

Explanation

LIST is an optional parameter specifying that DFSMSHsm release the list function. DFSMSHsm starts processing any queued LIST commands. Any list operation that was in process when list was held is not restarted.

Defaults

None.

LOG: Releasing logging

Explanation

LOG is an optional parameter specifying that DFSMSHsm begin recording in the DFSMSHsm log.

Defaults

None.

MIGRATION: Releasing space management

Explanation

MIGRATION is an optional parameter specifying whether DFSMSHsm should release automatic volume and automatic secondary space management plus command space management or only automatic volume and automatic secondary space management.

The MIGRATION parameter, without any subparameters, specifies that DFSMSHsm release command space management plus automatic volume and automatic secondary space management.

MIGRATION(AUTO) is a subparameter specifying that DFSMSHsm release only automatic volume and automatic secondary space management.

If tape migration has ended while DASD migration continues or if DASD migration has ended while tape migration continues, this parameter allows both DASD and tape to resume their acceptability as migration targets (for both MIGRATION alone and with AUTO).

Defaults

None.

Usage notes:

1. You cannot release the sub-function MIGRATION(AUTO) if the main function MIGRATION is held.
2. You can not release the sub-function AUTOEXPIRATION through the RELEASE MIGRATION(AUTO) command. You must use RELEASE AUTOEXPIRATION (RELEASE AEXP).
3. If the time is still within the automatic primary space management or automatic secondary space management processing window, any automatic primary space management or automatic secondary space management functions that were not completed when migration was held begin processing.

RECALL: Releasing recall and deletion

Explanation

RECALL is an optional parameter specifying whether DFSMSHsm should release automatic recall, recall by command, and deletion of a migrated data set from all volumes, from all tape volumes, or recalls that need tapes and were submitted by a TSO user.

The parameters of the RECALL parameter are explained:

TAPE

Releases held recalls from tape volumes.

TSO

Releases held recalls that required tapes that were submitted by a TSO user.

DASD

Releases held recalls from DASD volumes only, while maintaining the hold on recalls from tape volumes.

Defaults

If you specify the RECALL parameter with any other parameters, then the default is to release all recall and data set deletion tasks from all volumes.

Usage notes:

1. You cannot release the subfunction RECALL(TAPE(TSO)) if either the subfunction RECALL(TAPE) or the main function RECALL is held.
2. You cannot release the subfunction RECALL(TAPE) if the main function RECALL is held.
3. The RECALL(DASD) parameter does not follow the function-to-subfunction release relationship restriction described in the introduction topic [Chapter 62, “RELEASE command: Releasing all or some DFSMSHsm functions for processing,” on page 1029](#). The HOLD RECALL status is converted to a HOLD RECALL(TAPE) -- allowing recall from DASD to continue. There is no corresponding HOLD RECALL(DASD) command.

RECOVER: Releasing recovery and restore

Explanation

RECOVER is an optional parameter specifying whether DFSMSHsm should release all recovery processing or only tape data set recovery and restore processing.

The parameter of the RECOVER parameter is explained:

TAPEDATASET

Releases only tape data set recover and restore processing tasks.

The RELEASE FRRECOV command has the same effect when the system that issues the command is connected to a CVQ. However, this host volume recover tasks in the CVQ are used by only the master scheduler host when the following conditions are met:

If the host is the CVQ master scheduler, it is also eligible to perform MS assignment responsibilities for fast replication recovery requests.

- No other HOLD RECOVER is in place for the host.
- SETSYS MAXDUMPRECOVERTASKS(x) is specified with a value greater than 0.

Defaults

If you specify the RECOVER parameter with any other parameters, then the default is to release all recover and restore processing tasks.

Usage notes:

1. You cannot release the subfunction RECOVER(TAPEDATASET) if the main function RECOVER is held.

RECYCLE: Releasing the recycle function

Explanation

RECYCLE is an optional parameter specifying that DFSMSHsm release the recycle function. DFSMSHsm starts processing any queued recycle request. Any volumes that were being recycled when recycle was held are not restarted.

Defaults

None.

REPORT: Releasing the report function

Explanation

REPORT is an optional parameter specifying that DFSMSHsm release the report function. DFSMSHsm starts processing any queued report request. Any report that was in process when reporting was held is not restarted.

Defaults

None.

RMM: Releasing the DFSMSrmm function

Explanation

RMM is an optional parameter that enables the EDGTVEXT interface again after it was disabled because of an ABEND in DFSMSrmm.

Defaults

None.

TAPECOPY: Releasing the tape copy function

Explanation

TAPECOPY is an optional parameter specifying that the tape copy function is released. DFSMSHsm starts processing any queued TAPECOPY commands.

If the HOLD command has been issued while the TAPECOPY command is processing more than one volume, either from a volume list from the command or an input data set, processing stops after the current volume. TAPECOPY *does not* resume volume processing when the RELEASE command is issued. Therefore, you must reissue the TAPECOPY command to complete the processing of the remaining volumes in the list.

Defaults

None.

TAPEREPL: Releasing the tape replace function

Explanation

TAPEREPL is an optional parameter specifying that the tape replace function is released. DFSMSHsm starts processing any queued TAPEREPL commands.

If the HOLD command has been issued while the TAPEREPL command is processing more than one volume, either from a volume list from the command or an input data set, processing stops after the current volume. TAPEREPL *does not* resume volume processing when the RELEASE command is issued. Therefore, you must reissue the TAPEREPL command to complete the processing of the remaining volumes in the list.

Defaults

None.

Examples of how to code the RELEASE command

The following examples present different ways to code the RELEASE command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Printing the activity logs

In this example, DFSMSHsm prints activity logs that have data and allocates new activity logs.

```
RELEASE HARDCOPY
```

Releasing the aggregate backup function

In this example, DFSMSHsm releases the backup of aggregated data sets for processing.

```
RELEASE ABACKUP
```

Releasing the aggregate recovery function

In this example, DFSMSHsm releases the recovery of aggregated data sets for processing.

```
RELEASE ARECOVER
```

Releasing common queue functions

In this example, DFSMSHsm releases the common queue functions for processing.

```
RELEASE COMMONQUEUE
```

Releasing placement of recall requests in the CRQ

In this example, DFSMSHsm releases the CRQ placement function for processing.

```
RELEASE COMMONQUEUE(RECALL(PLACEMENT))
```

Releasing selection of recall requests in the CRQ

In this example, DFSMSHsm releases the CRQ selection function for processing.

```
RELEASE COMMONQUEUE(RECALL(SELECTION))
```

Releasing automatic volume and automatic secondary space management functions

In this example, DFSMSHsm releases the recall and deletion of a migrated data set and automatic volume and automatic secondary space management functions for processing. The MIGRATION parameter of the HOLD command was not specified previously.

```
RELEASE RECALL AUTOMIGRATION
```

Releasing the backup functions

In this example, DFSMSHsm releases the backup functions for processing.

```
RELEASE BACKUP
```

Releasing the expiration of backup versions

In this example, DFSMSHsm releases the expiration of backup versions for processing.


```
RELEASE EXPIREBV
```

Releasing the fast replication backup function

In this example, DFSMSHsm releases the fast replication backup for processing.

```
RELEASE FRBACKUP
```

Releasing the fast replication recover function

In this example, DFSMSHsm releases the fast replication recovery function for processing.

```
RELEASE FRRECOV
```

Releasing the list function

In this example, DFSMSHsm releases the list function for processing.

```
RELEASE LIST
```

Releasing all requests for tape recalls in the queue

In this example, DFSMSHsm releases all requests for tape recalls in the queue for processing.

```
RELEASE RECALL(TAPE)
```

Releasing all requests for tape recalls submitted by an interactive TSO user

In this example, DFSMSHsm releases all requests for tape recalls submitted by an interactive TSO user for processing.

```
RELEASE RECALL(TAPE(TSO))
```

Releasing the tape copy function

In this example, DFSMSHsm releases the tape copy function for processing.

```
RELEASE TAPECOPY
```

Releasing the tape replace function

In this example, DFSMSHsm releases the tape replace function for processing.

```
RELEASE TAPEREPL
```

Releasing automatic backup and command data set backups to tape

In this example, DFSMSHsm releases automatic backups and command data set backups directed to tape for processing.

```
RELEASE BACKUP(AUTO DSCOMMAND(TAPE))
```


Chapter 63. REPORT Command: Requesting reports based on DFSMSHsm statistics records

The REPORT command generates reports based on DFSMSHsm statistics records in the MCDS. You can get daily statistics reports and volume statistics reports for one function or for all DFSMSHsm functions, for one volume or for all volumes that have statistics records in the MCDS, and for statistics that have been created on a specific date or between two specific dates.

You can restrict the report to the functions DFSMSHsm performs, such as migration and backup, and you can restrict the report to specific dates.

You can request the following kinds of reports by using the REPORT command:

- Daily statistics report (DAILY)
- Volume statistics report (VOLUMES)

You can restrict the scope and format of the statistics reported as follows:

- Backup statistics report (FUNCTION BACKUP)
- Delete statistics report (FUNCTION DELETE)
- Migration statistics report (FUNCTION MIGRATION)
- Recall statistics report (FUNCTION RECALL)
- Recovery statistics report (FUNCTION RECOVER)
- Recycle statistics report (FUNCTION RECYCLE)
- Spill statistics report (FUNCTION SPILL)
- Transition statistics report (FUNCTION TRANSITION)
- Statistics other than function statistics (NOFUNCTION)

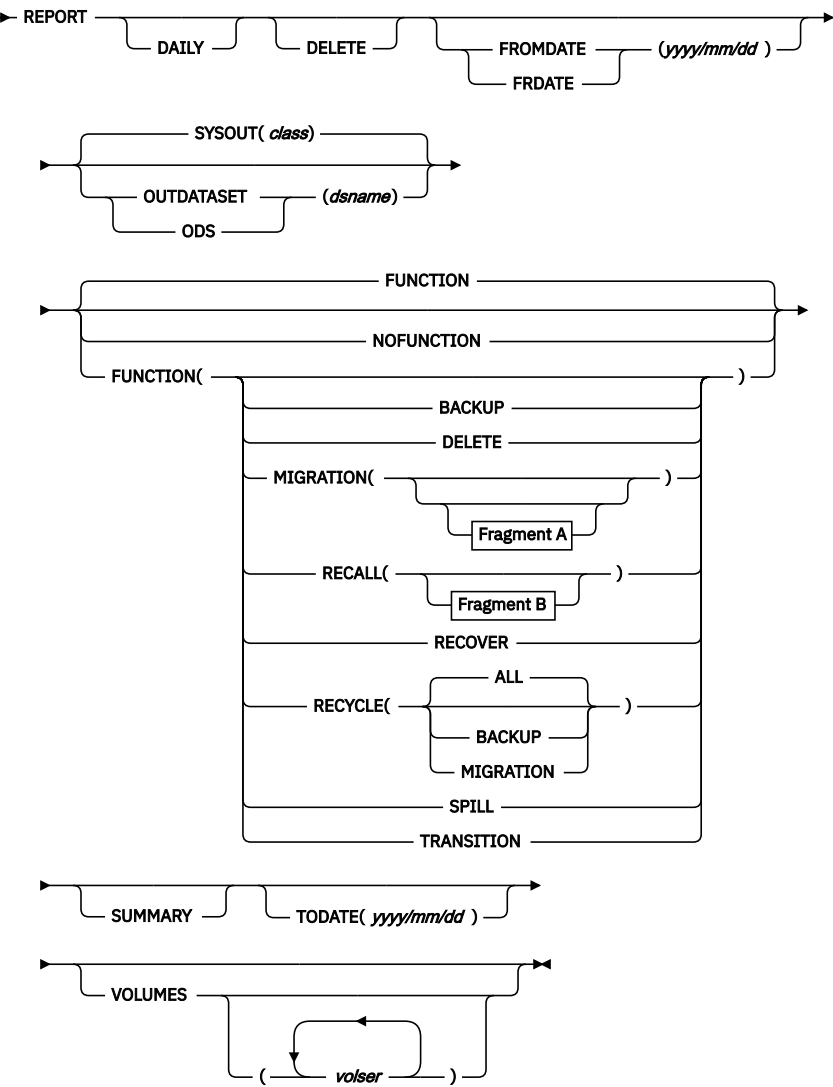
You can further restrict the scope of the statistics reports as follows:

- Statistics for dates on or after a specified date (FROMDATE)
- Statistics between two specified dates (FROMDATE and TODATE)
- Summary totals for the statistics reports (SUMMARY)

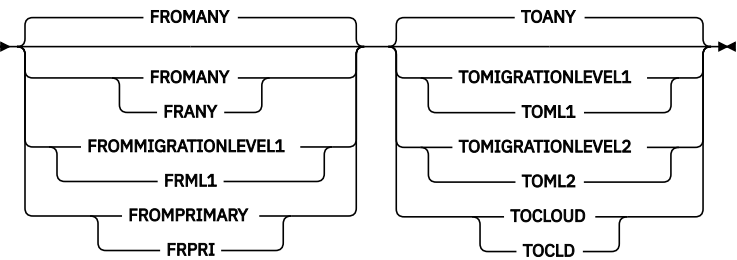
DFSMSHsm allows only one REPORT command to run at a time. [Chapter 78, “Using the REPORT command,”](#) on page 1401 contains sample reports of daily statistics and volume statistics.

For an alternative method of creating customized reports about HSM activities from SMF (FSR and WWFSR) records see [Chapter 14, “Obtaining information from DFSMSHsm,”](#) on page 559 and [z/OS DFSMSrmm Reporting](#).

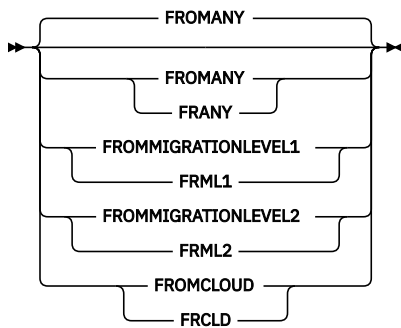
Syntax of the REPORT command



A: MIGRATION optional parameters



B: RECALL optional parameters



Summary of parameters

The following table is a summary of the combination of parameters you can specify with the REPORT command:

Parameter	Related Parameters
DAILY	All parameters except VOLUMES
DELETE	DAILY or VOLUMES
FROMDATE	DAILY, SUMMARY, TODATE, or VOLUMES
FUNCTION (and all subparameters of FUNCTION)	DAILY or VOLUMES
NOFUNCTION	DAILY or VOLUMES
OUTDATASET	DAILY or VOLUMES
SUMMARY	DAILY, FROMDATE, FUNCTION, TODATE, or VOLUMES
SYSOUT	DAILY or VOLUMES
TODATE	DAILY, FROMDATE, or VOLUMES
VOLUMES	All parameters except DAILY

Optional parameters of the REPORT command

This topic describes the optional parameters of the REPORT command.

DAILY: Requesting daily statistics

Explanation: DAILY is an optional parameter requesting a daily statistics report for all functions. See [Figure 162 on page 1405](#) for an example of the REPORT DAILY output.

Defaults: If you do not specify DAILY or VOLUMES, DFSMSHsm prints a daily statistics report.

Note:

1. If you want a report of daily statistics for all DFSMSHsm functions, specify REPORT DAILY FUNCTION. DFSMSHsm then reports statistics for the current date unless you also specify the FROMDATE and TODATE parameters.
2. For a daily report, the numbers printed on the report indicate the number of attempted DFSMSHsm functional processing requests. Functional processing is not attempted for the following invalid requests:
 - HMIGRATE requests for uncataloged data sets. The command function verifies that the data sets are cataloged before sending an MWE to DFSMSHsm.
 - HRECALL requests for uncataloged data sets. The command function verifies that the data sets are cataloged before sending an MWE to DFSMSHsm.

- Any request that attempts to migrate a volume that is not added to DFSMSHsm without specifying the UNIT parameter.

REPORT DAILY causes DFSMSHsm to update the daily statistics record (DSR) in the MCDS with information from the copy in storage where the command was processed. The report for today does not include any information about functions that were attempted in the last hour from any other DFSMSHsm host in the environment.

3. You can specify either DAILY or VOLUMES. If you specify both parameters, DFSMSHsm prints a volume statistics report.
4. MIGRATION and RECALL statistics do not include amounts for extent reductions.

DELETE: Deleting statistics records used as input to the report

Explanation: DELETE is an optional parameter specifying that DFSMSHsm delete the statistics records used as input to the report after DFSMSHsm finishes processing the REPORT command.

Defaults: If you do not specify DELETE with REPORT, DFSMSHsm does not delete the statistics records used as input to the report.

Note: This parameter is not related to the DELETE subparameter of the FUNCTION parameter.

FROMDATE: Requesting statistics of activity on or after the specified date

Explanation: FROMDATE(*yyyy/mm/dd*) is an optional parameter limiting the statistics report to activity that occurred on or after the date you specify. For *yyyy/mm/dd*, substitute the first date you want DFSMSHsm to report statistics. For example, if you specify FROMDATE(2001/02/01), DFSMSHsm reports statistics for Feb. 1, 2001 through the current date.

A leading zero is not required for a one-digit month or day.

Defaults: If you do not specify FROMDATE, the starting date defaults to the current date.

If you specify a FROMDATE that is later than TODATE, DFSMSHsm sets FROMDATE to the same date as TODATE.

Note:

1. You must specify *yyyy/mm/dd* if you specify FROMDATE.
2. If you specify that DFSMSHsm report the statistics for the current day, DFSMSHsm does not give you the latest statistics. The statistics records are kept in storage and accumulated with a control data set copy of the record only once an hour. Also, each DFSMSHsm host uses the same control data set record. New records are created each day for the DFSMSHsm activities that have occurred today. Therefore, use today's statistics report as an indication of trends of DFSMSHsm activity.
3. If FROMDATE is specified, it must not be after TODATE (specified or defaulted).
4. For REPORT VOLUMES, DFSMSHsm does not allow requesting a range of dates spanning the century. In other words, both FROMDATE and TODATE (whether specified or defaulted) must be before 1 January 2000, or both FROMDATE and TODATE (whether specified or defaulted) must be after 31 December 1999.

FUNCTION and NOFUNCTION: Requesting statistics by function

Explanation: FUNCTION and NOFUNCTION are mutually exclusive optional parameters specifying whether to report the statistics by function.

FUNCTION

specifies that statistics for each specified function are reported.

NOFUNCTION

specifies that function statistics are not reported.

Defaults: The default is FUNCTION. If you specify FUNCTION without any subparameters, DFSMSHsm reports the statistics for all functions.

Note: Because of the number of subparameters of the FUNCTION parameter, each subparameter is described separately.

FUNCTION(BACKUP): Requesting statistics of the backup function

Explanation: BACKUP is an optional subparameter of the FUNCTION parameter to request a statistics report of data sets that have been backed up to daily backup volumes. You can also use it to request a statistics report of EXPIREBV requests.

Defaults: None.

FUNCTION(DELETE): Requesting statistics of the delete function

Explanation: DELETE is an optional subparameter of the FUNCTION parameter to request a statistics report of the migrated data sets that have been deleted or data sets that have been scratched during data set deletion or data set retirement. Deletion of primary data sets includes only those deleted by the DELETEBYAGE (DBA) or DELETEIFBACKEDUP (DBU) functions. The DBA and DBU functions apply only to non-SMS-managed data sets.

Defaults: None.

FUNCTION(MIGRATION): Requesting statistics of the migration function

Explanation: MIGRATION is an optional subparameter of the FUNCTION parameter to request a statistics report about migration.

Use the subparameters of the MIGRATION parameter to select statistics reports about data sets migrated between specified levels of DFSMSHsm storage. For example, when you specify MIGRATION and the FROMPRIMARY and TOMIGRATIONLEVEL1 subparameters, you get a statistics report about only those data sets that DFSMSHsm migrated from primary volumes to migration level 1 volumes.

The following table shows the type of statistics report you get for migration:

Source Volume Subparameter	Target Volume Subparameter	Type of Migration on Statistics Report
FROMANY	TOANY	Primary to Level 1 Primary to Level 2 Primary to Cloud SUBSEQUENT MIGS
FROMANY	TOMIGRATIONLEVEL1	Primary to Level 1
FROMANY	TOMIGRATIONLEVEL2	Primary to Level 2 SUBSEQUENT MIGS
FROMANY	TOCLOUD	Primary to Cloud
FROMMIGRATIONLEVEL1	TOANY	SUBSEQUENT MIGS
FROMMIGRATIONLEVEL1	TOMIGRATIONLEVEL1	
FROMMIGRATIONLEVEL1	TOMIGRATIONLEVEL2	SUBSEQUENT MIGS
FROMPRIMARY	TOANY	Primary to Level 1 Primary to Level 2
FROMPRIMARY	TOMIGRATIONLEVEL1	Primary to Level 1
FROMPRIMARY	TOMIGRATIONLEVEL2	Primary to Level 2
FROMPRIMARY	TOCLOUD	Primary to Cloud

Defaults: When you specify MIGRATION without specifying subparameters, the defaults are FROMANY and TOANY.

Note:

1. When you specify FROMANY and TOMIGRATIONLEVEL2, DFSMSHsm includes data sets that migrate from migration level 2 volumes to different migration level 2 volumes in the statistics of data sets that migrate from migration level 1 volumes to migration level 2 volumes.
2. If you want information about movement from one tape migration level 2 volume to another tape migration level 2 volume, use the RECYCLE parameter.
3. SUBSEQUENT MIGS reflects all movements of data sets from migration level 1 to migration level 1, migration level 1 to migration level 2, and migration level 2 to migration level 2. This includes FREEVOL moves but does not include RECYCLE moves.

FUNCTION(RECALL): Requesting statistics of the recall function

Explanation: RECALL is an optional subparameter of the FUNCTION parameter to request a statistics report for the recall function. The parameters of the RECALL parameter are explained:

FROMANY

specifies that DFSMSHsm should create a statistics report of the data sets recalled from migration level 1 and migration level 2 volumes.

FROMCLOUD

specifies that DFSMSHsm should create a statistics report of the data sets recalled from cloud storage.

FROMMIGRATIONLEVEL1

specifies that DFSMSHsm should create a statistics report of the data sets recalled from migration level 1 volumes.

FROMMIGRATIONLEVEL2

specifies that DFSMSHsm should create a statistics report of the data sets recalled from migration level 2 volumes.

Defaults: When you specify RECALL without specifying subparameters, the default is FROMANY.

FUNCTION(RECYCLE): Requesting statistics of the recycle function

Explanation: RECYCLE is an optional subparameter of the FUNCTION parameter to request a statistics report of the data sets that have been recycled. The parameters of the RECYCLE parameter are explained:

BACKUP

specifies that DFSMSHsm should create a statistics report about the data sets that DFSMSHsm moved from tape backup volumes to tape spill backup volumes during recycle processing.

MIGRATION

specifies that DFSMSHsm should create a statistics report about the data sets that DFSMSHsm moved from the tape migration level 2 volumes to other tape migration level 2 volumes during recycle processing.

ALL

specifies that DFSMSHsm should create a statistics report about tape migration level 2 volumes and tape backup volumes.

Defaults: If you specify RECYCLE and do not specify a subparameter, the default is ALL.

FUNCTION(RECOVER): Requesting statistics of the recovery function

Explanation: RECOVER is an optional subparameter of the FUNCTION parameter to request a statistics report of the data sets that have been recovered from backup volumes.

Defaults: None.

FUNCTION(SPILL): Requesting statistics of the spill function

Explanation: SPILL is an optional subparameter of the FUNCTION parameter to request a statistics report of subsequent movement of backup versions.

Note: These statistics include data sets spilled from daily BACKUP volumes, data sets moved as a result of the FREEVOL BACKUPVOLUME command, and data sets moved from ML1 volumes to BACKUP volumes.

Defaults: None.

FUNCTION(TRANSITION): Requesting statistics of the transition function

Explanation: TRANSITION is an optional subparameter of the FUNCTION parameter to request a statistics report of class transition-related information.

Defaults: None.

OUTDATASET and SYSOUT: Specifying the output location

Explanation: OUTDATASET(*dsname*) and SYSOUT(*class*) are mutually exclusive, optional parameters specifying the output location for the statistics report.

OUTDATASET(*dsname*)

specifies the name of the data set where DFSMSHsm is to write the output data. For *dsname*, substitute the fully qualified name of the data set to receive the statistics report.

If the data set does not exist, DFSMSHsm dynamically allocates and catalogs an output data set with the following characteristics:

- Data set name specified (*dsname*)
- Record format of fixed-blocked with ANSI control characters (FBA)
- Logical record length of 121
- Data set is system reblockable
- Primary allocation of 20 tracks (see second note)
- Secondary allocation of 50 tracks (see second note)
- Unit of SYSALLDA (see second note)

If the data set already exists, DFSMSHsm will use the data set. The data set must have the following characteristics:

- The data set must be cataloged and on DASD.
- The data set record format must be FBA, and the logical record length must be 121.
- The data set is system reblockable.
- The user can choose the primary space allocation.
- If DFSMSHsm needs additional extents after the primary space allocation, DFSMSHsm uses a secondary space allocation of 50 tracks. (see second note)
- If the data set does not contain data, DFSMSHsm starts writing output data at the beginning of the data set.
- If the data set contains data, DFSMSHsm writes the output data after the existing data.

SYSOUT(*class*)

specifies that the statistics report is to be sent to the specified system output class. For *class*, substitute the alphanumeric character for the system output class.

Defaults: The default is SYSOUT, and *class* defaults to the current DFSMSHsm value for the SYSOUT parameter. You specify the current DFSMSHsm value with the SETSYS command. If you do not specify SYSOUT with the SETSYS command, the default is class A.

Note:

1. DFSMSHsm does not handle partitioned data set members individually. If you specify a partitioned data set with a member name as the output data set, the statistics report could be written over existing data. If you do not allocate an output data set with the required characteristics, I/O errors could occur on the output data set.
2. If you select the OUTDATASET option, you can use the PATCH command to change the unit name, primary allocation, and secondary allocation. If you select the SYSOUT option, you can use the PATCH command to change whether, and how, DFSMSHsm limits the lines of SYSOUT output. Refer to the [*z/OS DFSMSHsm Implementation and Customization Guide*](#) for the PATCH command.

SUMMARY: Requesting only the totals of the statistics

Explanation: SUMMARY is an optional parameter requesting that DFSMSHsm print a summary of statistics for the days you specified with the FROMDATE and TODATE parameters.

Defaults: None.

Note:

1. You specify REPORT DAILY FUNCTION SUMMARY to get the summary totals for the daily statistics for all functions. You specify REPORT VOLUME FUNCTION SUMMARY to get the summary totals for the volume statistics for all volumes and for all functions. You specify the VOLUMES parameter with a volume serial number to get summary statistics for only one volume. If you request a summary of specified functions, the summary contains the statistics totals for only the specified functions.
2. When you specify SUMMARY, you get only the summary report. If you do not specify SUMMARY, you get the statistics report you requested, and a summary at the end of the report if DFSMSHsm reported the statistics for more than one day.

TODATE: Requesting statistics of activity on or before the specified date

Explanation: TODATE(*yyyy/mm/dd*) is an optional parameter you use with FROMDATE to limit statistics reports to activity that occurred during that range of time. For *yyyy/mm/dd*, substitute the last date you want statistics reported.

A leading zero is not required for a one-digit month or day.

Defaults: If you do not specify TODATE, the ending date defaults to the current date.

Note:

1. You must specify *yyyy/mm/dd* if you specify TODATE.
2. If you specify that DFSMSHsm report the statistics for the current day, DFSMSHsm does not give you the latest statistics. The statistics records are kept in storage and accumulated with a control data set copy of the record only once an hour. Also, each DFSMSHsm host uses the same control data set record. New records are created each day for the DFSMSHsm activities that occurred today. Therefore, use today's statistics report as an indication of trends of DFSMSHsm activity.
3. If TODATE is specified, it must not be before FROMDATE (specified or defaulted).
4. For REPORT VOLUMES, DFSMSHsm does not allow requesting a range of dates spanning the century. In other words, both FROMDATE and TODATE (whether specified or defaulted) must be before 1 January 2000, or both FROMDATE and TODATE (whether specified or defaulted) must be after 31 December 1999.

For REPORT VOLUMES, DFSMSHsm will not report volume statistics for the current date unless that date is within the range of dates requested.

VOLUMES: Requesting volume statistics

Explanation: VOLUMES(*volser* ...) is an optional parameter requesting a volume statistics report for specified volumes or for all volumes with statistics records in the MCDS. For *volser*, substitute the serial number of the volume or a list of serial numbers of the volumes whose statistics you want reported.

You specify REPORT VOLUMES FUNCTION to get a list of the volume statistics for all functions for all volumes managed by DFSMSHsm. You specify VOLUMES with a volume serial number to get a statistics report for a specific volume. If you do not specify the FROMDATE and TODATE parameters, you receive a statistics report only for the current date.

See [Figure 163 on page 1410](#) for an example of the REPORT VOLUMES output.

Defaults: If you specify VOLUMES without *volser*, DFSMSHsm reports statistics for all volumes that have statistics records.

Note: Any data sets that existed on a volume before you used the ADDVOL command to add the volume to DFSMSHsm control are not included in the report unless DFSMSHsm audited the volume just before the report and no other data sets have been allocated to the volume since the audit.

The volume report information is obtained from the volume statistics record (VSR) in the MCDS. These records are written to the MCDS each hour. Any activity that has occurred since the last update of the VSR will not be reported.

Examples of how to code the REPORT command

For examples of the REPORT command, see [Chapter 78, “Using the REPORT command,” on page 1401](#).

Chapter 64. SETMIG command: Changing the space management status of non-SMS-managed data sets or volumes

The SETMIG command changes the space management status of a specific data set or group of data sets on level 0 volumes. It can also change the space management status of a specific primary volume. You use this command to temporarily change the type of space management for specific cases.

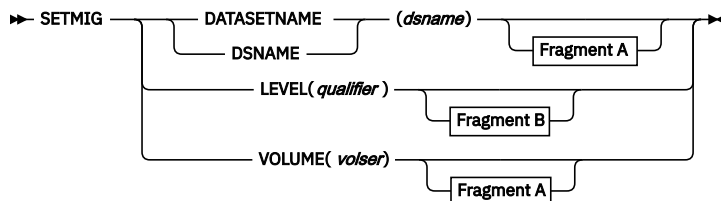
You can change the space management status in three ways:

- Data set migration only by command
- Command and automatic space management
- No command or automatic space management

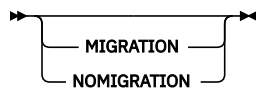
If you specify a value for DATASETNAME(*dsname*), you do not have to repeat the SETMIG command each time you start DFSMSHsm. But if you specify a value for LEVEL(*qualifier*) or VOLUME(*volser*), you must repeat the SETMIG command each time you start DFSMSHsm.

SETMIG is not supported for SMS-managed volumes or data sets except to remove or restore the SYS1 and HSM data set qualifier restriction when using the LEVEL parameter. If an SMS-managed volume is specified, an error message is issued and no function is performed. The SETMIG command is processed for an SMS-managed data set, but it has no effect on the processing of the data set, because the space management status is determined by a data set's management class attribute.

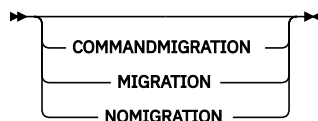
Syntax of the SETMIG command



A: DATASETNAME and VOLUME optional parameters



B: LEVEL optional parameters



Required parameters of the SETMIG command

This topic describes the required parameters of the SETMIG command.

DATASETNAME, LEVEL, and VOLUME: Changing the space management status

Explanation

DATASETNAME(*dsname*), LEVEL(*qualifier*), and VOLUME(*volser*) are mutually exclusive, required parameters to change the space management status of a data set, group of data sets, or volume.

DATASETNAME(*dsname*)

Changes the space management status of a data set on a level 0 volume.

dsname

Fully qualified name of the data set whose space management status is changing.

LEVEL(*qualifier*)

Changes the space management status of a group of data sets that have the same set of initial characters of the data set name. The data sets are on level 0 volumes.

qualifier

First qualifier of the data set name or any number of contiguous characters, beginning with the first character of the data set name for a group of data sets whose space management status is changing. In other words, any number of leading characters of a data set name can be specified, and the last set of characters can be a partial qualifier.

VOLUME(*volser*)

Changes the space management status of the primary volume.

volser

Serial number of the volume. If you specified the AUTOMIGRATION parameter of the ADDVOL command for this volume, you can use the VOLUME parameter of the SETMIG command to temporarily exclude the volume from automatic primary space management. Even if you temporarily exclude the volume from automatic primary space management, you can still issue commands to migrate a specific data set on the volume.

Defaults

None.

Usage notes:

1. DFSMSHsm does not handle partitioned data set members individually. If you specify a partitioned data set with a member name, DFSMSHsm changes the status of the entire partitioned data set.

Optional parameters of the SETMIG command

This topic describes the optional parameters of the SETMIG command.

COMMANDMIGRATION, MIGRATION, and NOMIGRATION: Specifying the way space management is controlled

Explanation

COMMANDMIGRATION, MIGRATION, and NOMIGRATION are mutually exclusive, optional parameters specifying how space management is controlled for a data set, a group of data sets, or all data sets on a volume.

COMMANDMIGRATION

valid only with the LEVEL parameter, specifies that DFSMSHsm allows space management by data set command for any of a group of data sets you identified with the LEVEL parameter. DFSMSHsm does not manage these data sets when it processes a volume.

MIGRATION

specifies that DFSMSHsm will do space management automatically or by command for a data set, a group of data sets, or all data sets on a volume. You specify the MIGRATION parameter only if you previously specified the COMMANDMIGRATION or NOMIGRATION parameter with the SETMIG command.

NOMIGRATION

specifies that DFSMSHsm will not do space management for the data set, group of data sets, or the volume; however, DFSMSHsm will do space management by data set command for data sets when this parameter is specified with the VOLUME parameter. If NOMIGRATION is specified with the data set name or LEVEL parameter, DFSMSHsm will not do space management by data set command.

Defaults

None.

Usage notes:

1. The COMMANDMIGRATION parameter applies only when you specify the LEVEL parameter. If you specify COMMANDMIGRATION when it does not apply, DFSMSHsm ignores it.
2. SETMIG with the MIGRATION option is only used to remove a previously entered SETMIG restriction.
3. If no SETMIG command is issued for a given data set, the data set is eligible for both command and automatic migration.
4. If a data set is specified in more than one LEVEL parameter, that data set is given the space management status that is the most restrictive. For example, given the following commands:

```
SETMIG LEVEL(A.B) MIGRATION
SETMIG LEVEL(A) COMMANDMIGRATION
```

any data sets that have the first two qualifiers of A.B are covered under LEVEL(A) COMMANDMIGRATION since it is more restrictive and the first SETMIG command is ignored. This is true regardless of the order in which the SETMIG commands are issued. The most restrictive SETMIG option is NOMIGRATION followed by COMMANDMIGRATION. The MIGRATION option basically indicates that there are no restrictions or that any previous restrictions are being removed.

Examples of how to code the SETMIG command

The following examples present different ways to code the SETMIG command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Changing the space management status of a specific data set

In this example, the space management status of a particular data set is changed to prevent space management by volume processing or data set command.

```
SETMIG DATASETNAME(FST1234.COMPARE.REPORT) NOMIGRATION
```

Changing the space management status of a group of data sets

In this example, the space management status of a group of data sets with the same first qualifier of the data set name is changed to prevent space management by volume processing or by command.

```
SETMIG LEVEL(JJT2143) NOMIGRATION
```


Changing the space management status of a volume

In this example, the space management status of a DFSMSHsm-managed volume is changed to allow automatic and command space management, which was previously restricted, to occur from the volume.

```
SETMIG VOLUME(VOL123) MIGRATION
```

Changing the space management status of a group of data sets to allow space management only by command

In this example, the space management status of a group of data sets is changed to allow space management of a data set only when requested by a command.

```
SETMIG LEVEL(MIL3691.REV3) COMMANDMIGRATION
```

This command would apply to all the data sets listed here:

- MIL3691.REV3
- MIL3691.REV3VER2.LISTING
- MIL3691.REV3456.TEST
- MIL3691.REV3TST.EXAM.LIST

Chapter 65. SETSYS command: Establishing or changing the values of DFSMSHsm control parameters

When you start DFSMSHsm, a subset of DFSMSHsm control parameters is established by default. You can override DFSMSHsm defaults by specifying one or more SETSYS commands in the ARCCMDxx PARMLIB member used when you start DFSMSHsm. You can issue the SETSYS command with specific parameter values after DFSMSHsm is started to change the current defaults. The changed values remain in effect until you restart DFSMSHsm. For an example of how to set up the ARCCMDxx PARMLIB member, see the DFSMSHsm installation verification procedure in *z/OS Program Directory* in the *z/OS Internet library* (www.ibm.com/servers/resourcelink/svc00100.nsf/pages/zosInternetLibrary).

If you do not specify the SETSYS command, DFSMSHsm does not do any automatic space management, backup, or dump. Therefore, if you want to take advantage of the automatic functions of DFSMSHsm, use the SETSYS command.

Do not confuse DFSMSHsm defaults with the SETSYS command defaults. Except for certain values, there are no SETSYS command defaults. A SETSYS command does not have required parameters, so unless you indicate a specific parameter value for the SETSYS command, the DFSMSHsm control parameter is the default. An exception to this is the CDSVERSIONBACKUP parameter.

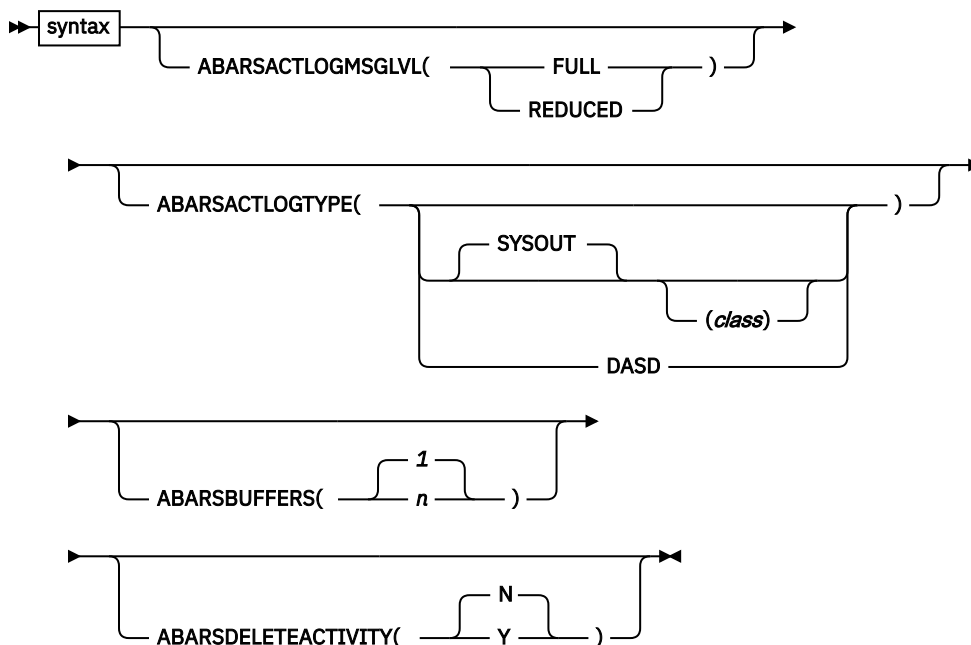
The following link provides a deep dive into the many knobs and dials available to configure DFSMSHsm uniquely to your environment.

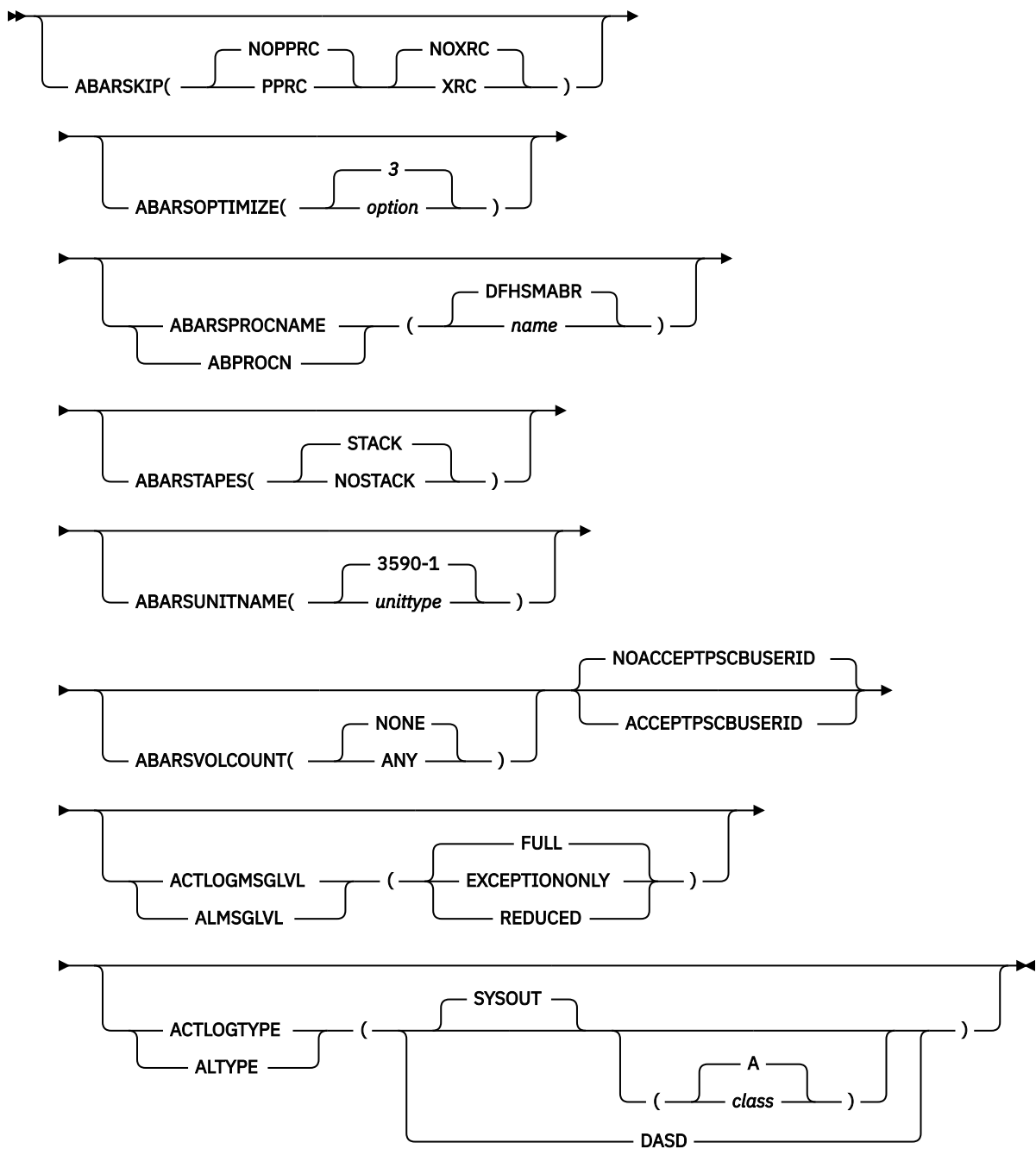
[DFSMSHsm Education Series-hsm SETSYS \(mediacenter.ibm.com/media/DFSMSHsm%20Education%20Series-hsm%20SETSYS/1_qh3d8942\)](http://mediacenter.ibm.com/media/DFSMSHsm%20Education%20Series-hsm%20SETSYS/1_qh3d8942)

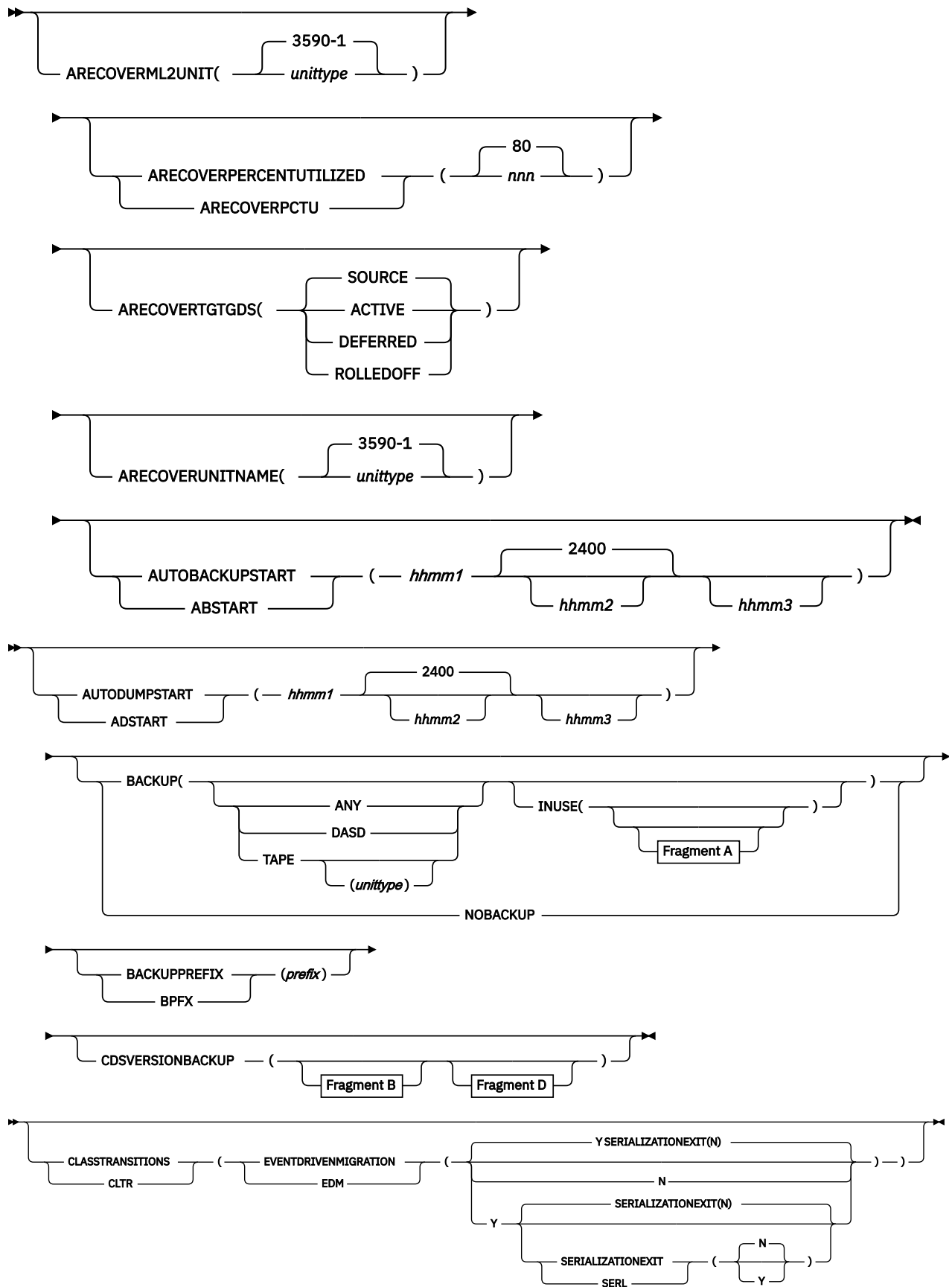
Note: Although the SETSYS command has no required parameters, you must specify at least one of the optional parameters to change any system parameters or default values.

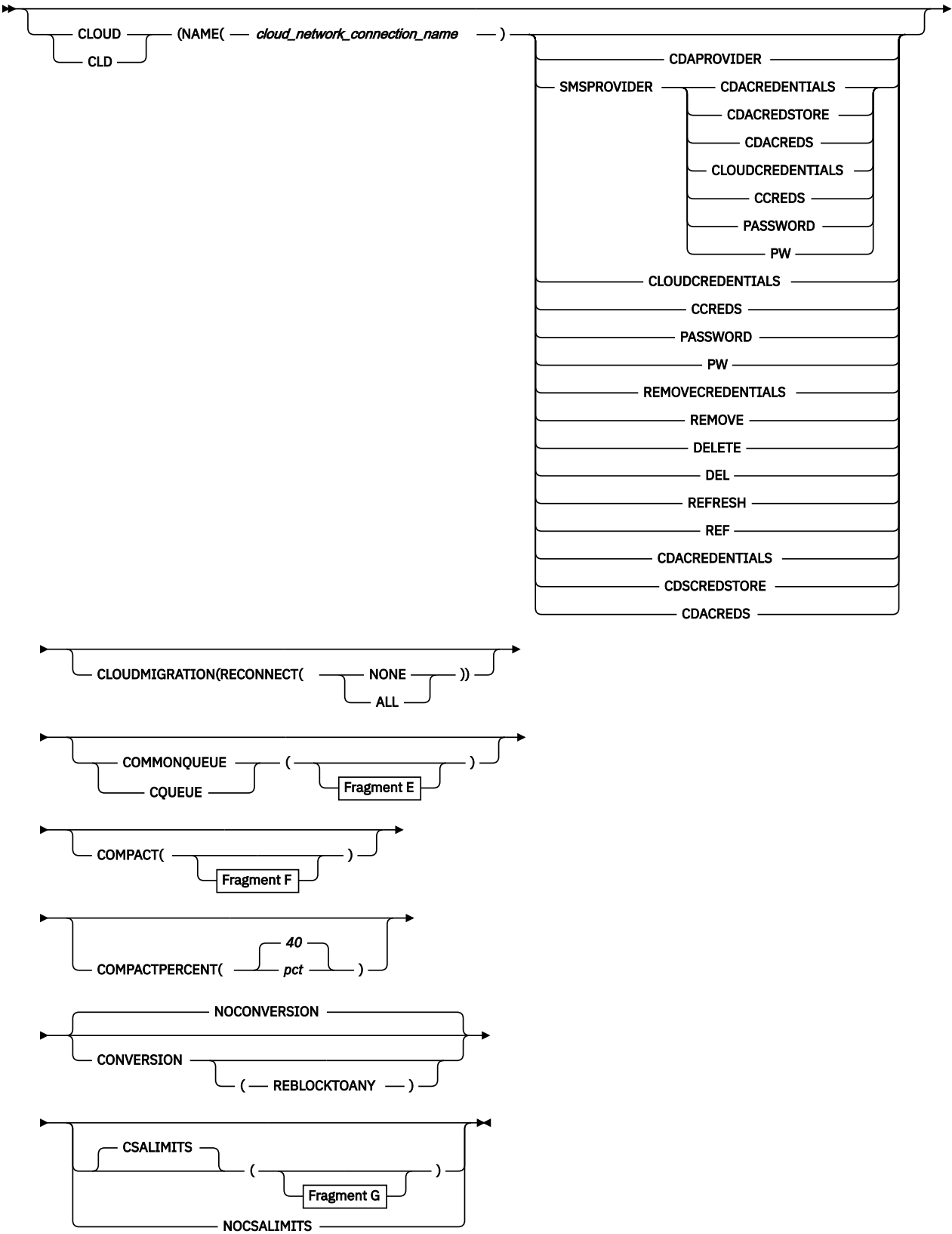
Syntax of the SETSYS command

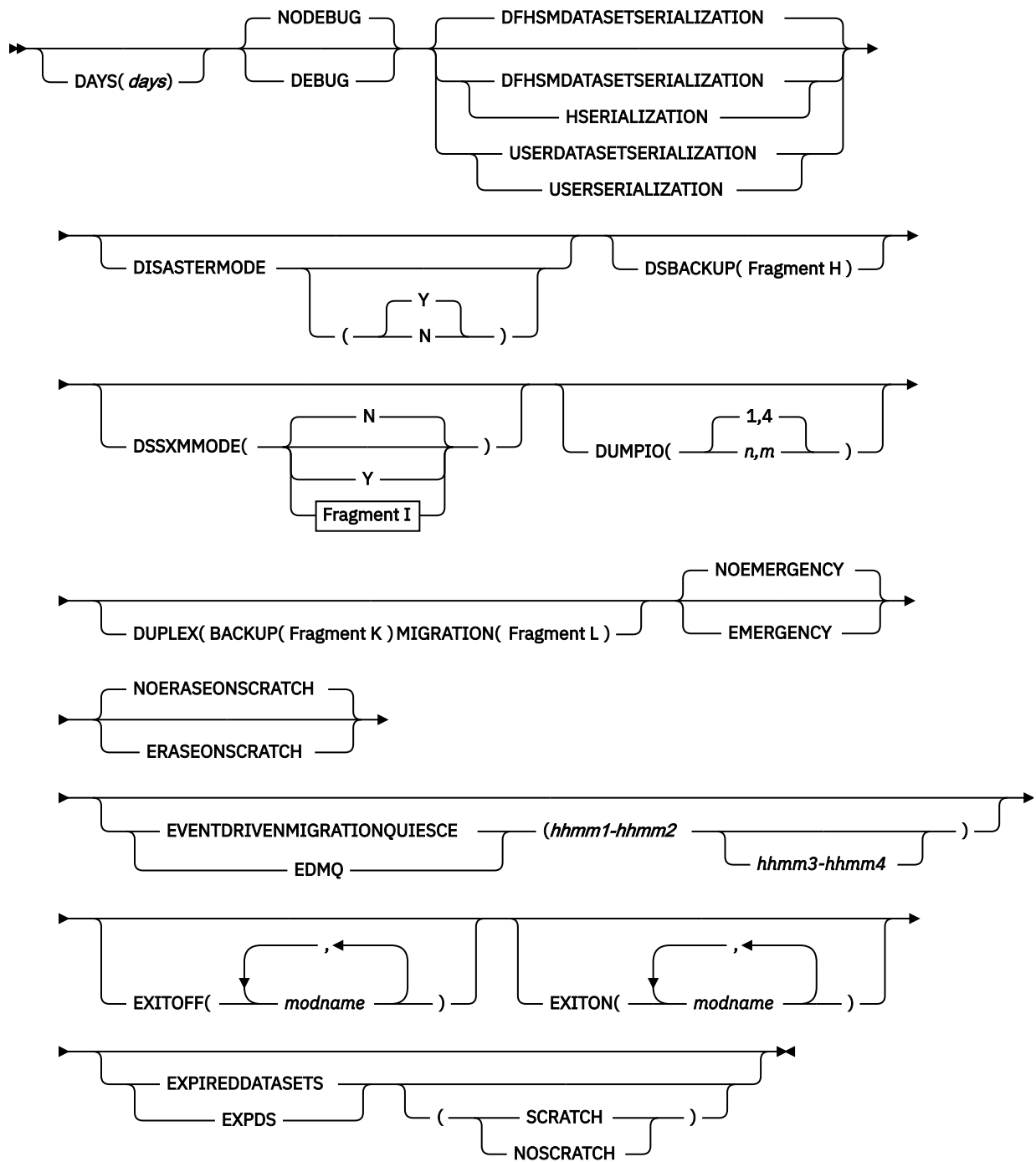
➤ SETSYS ➤

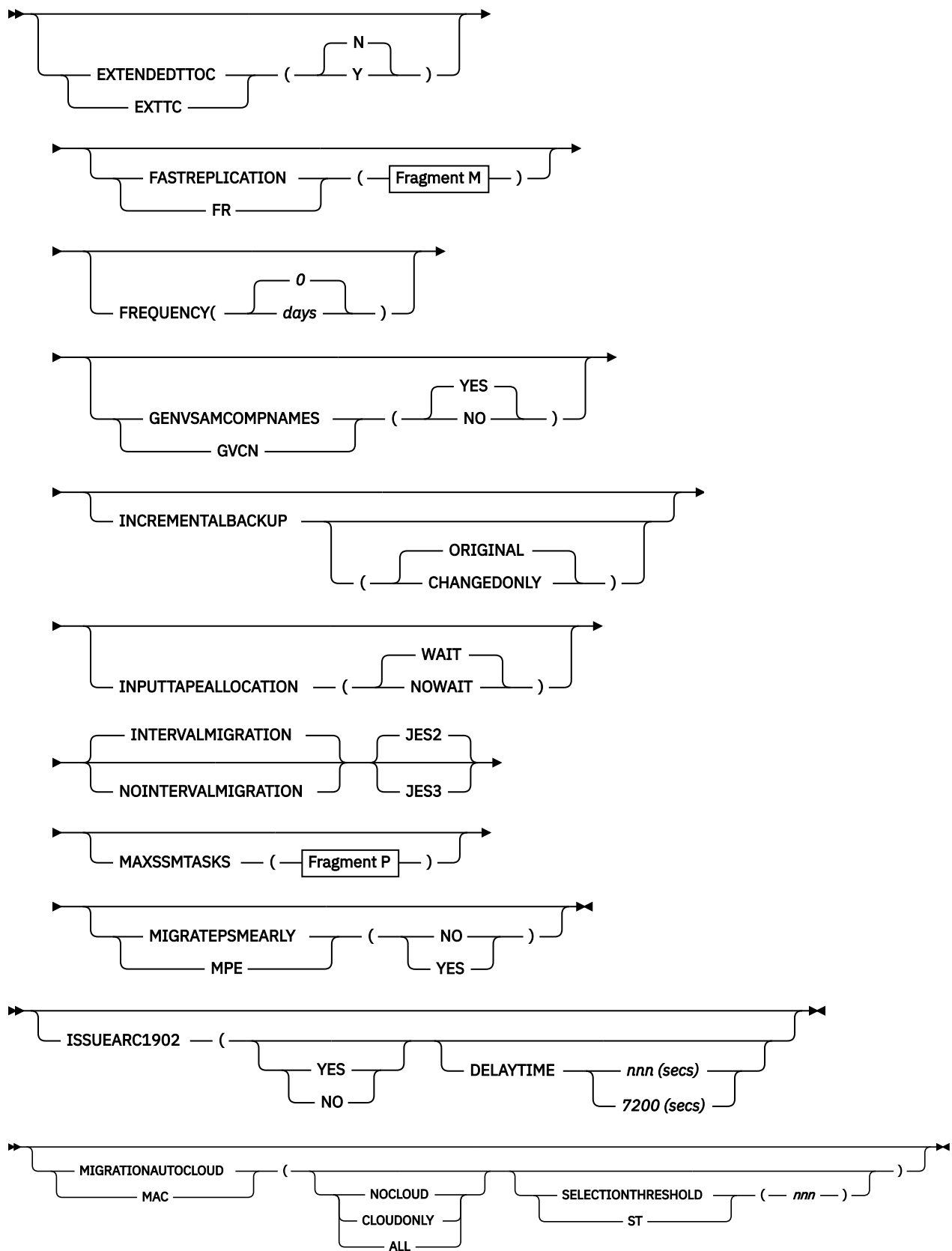


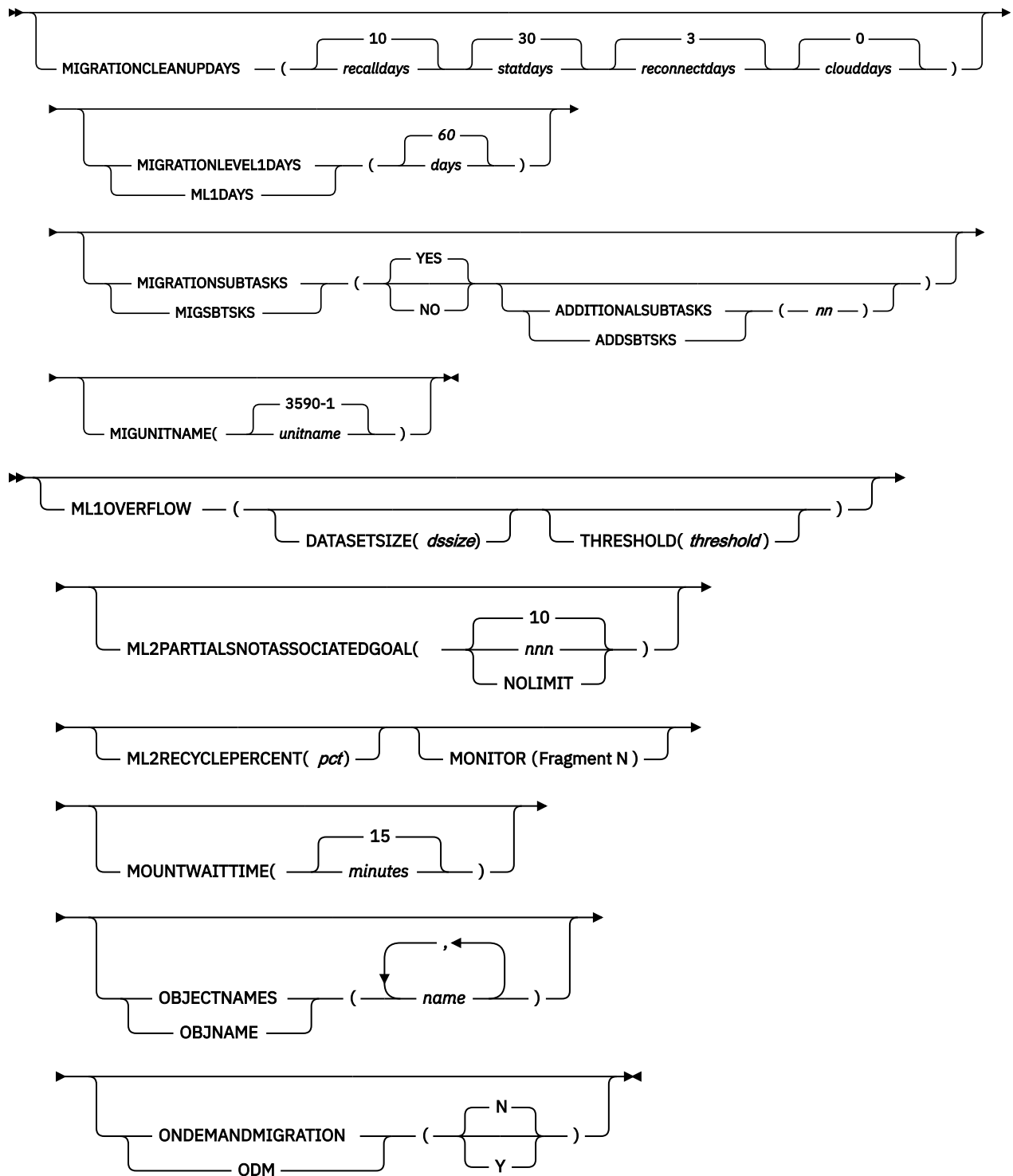


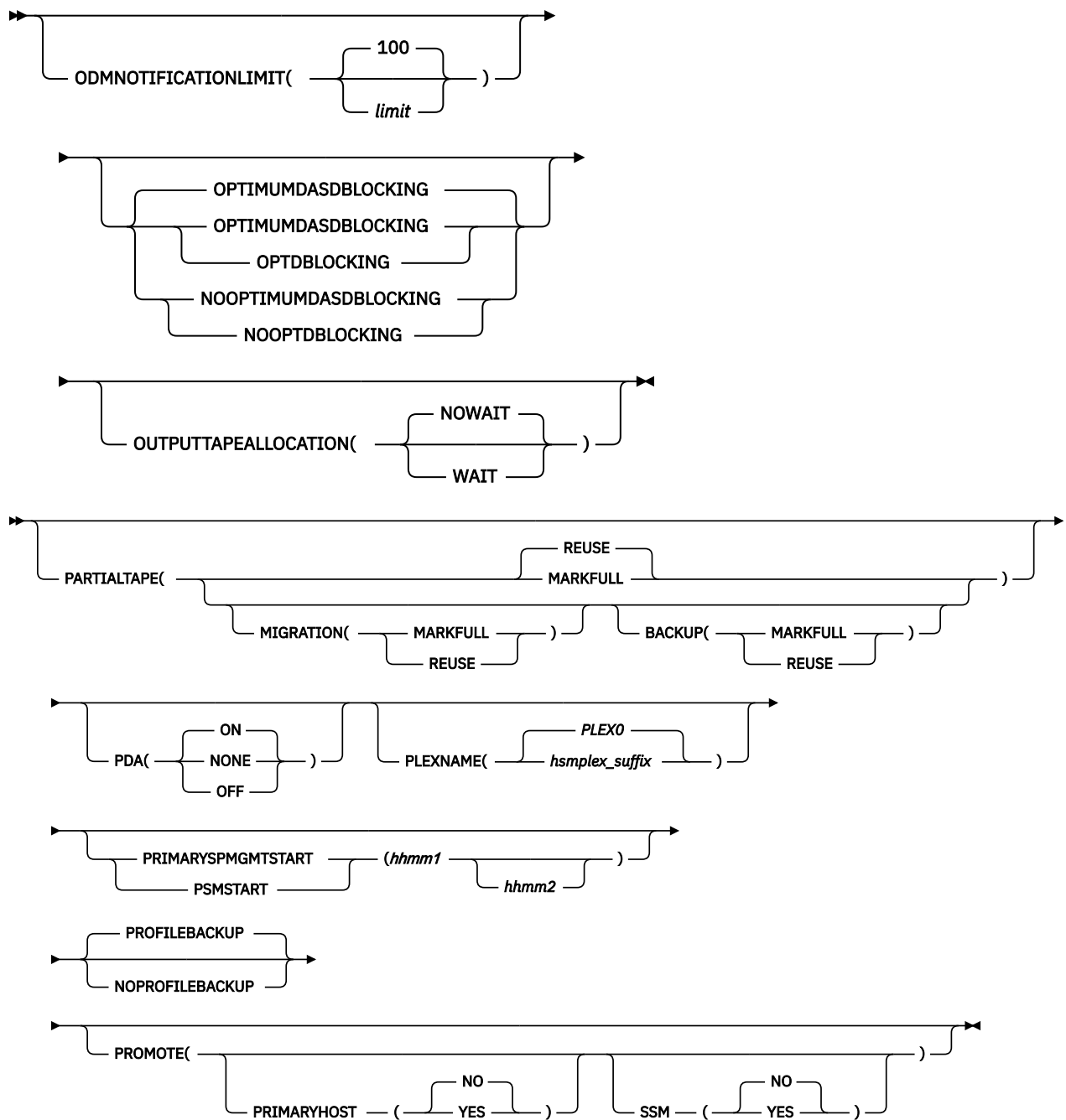


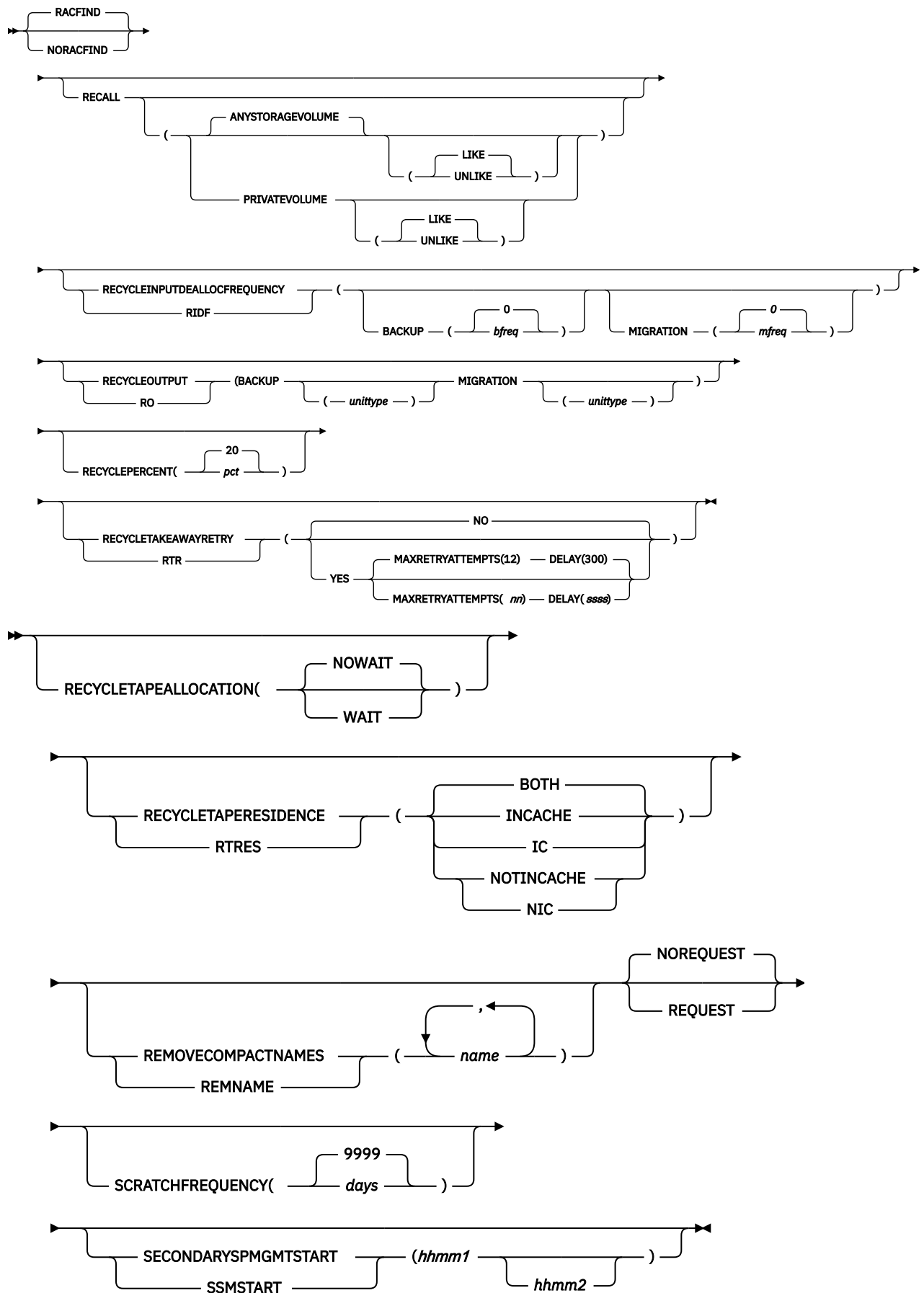


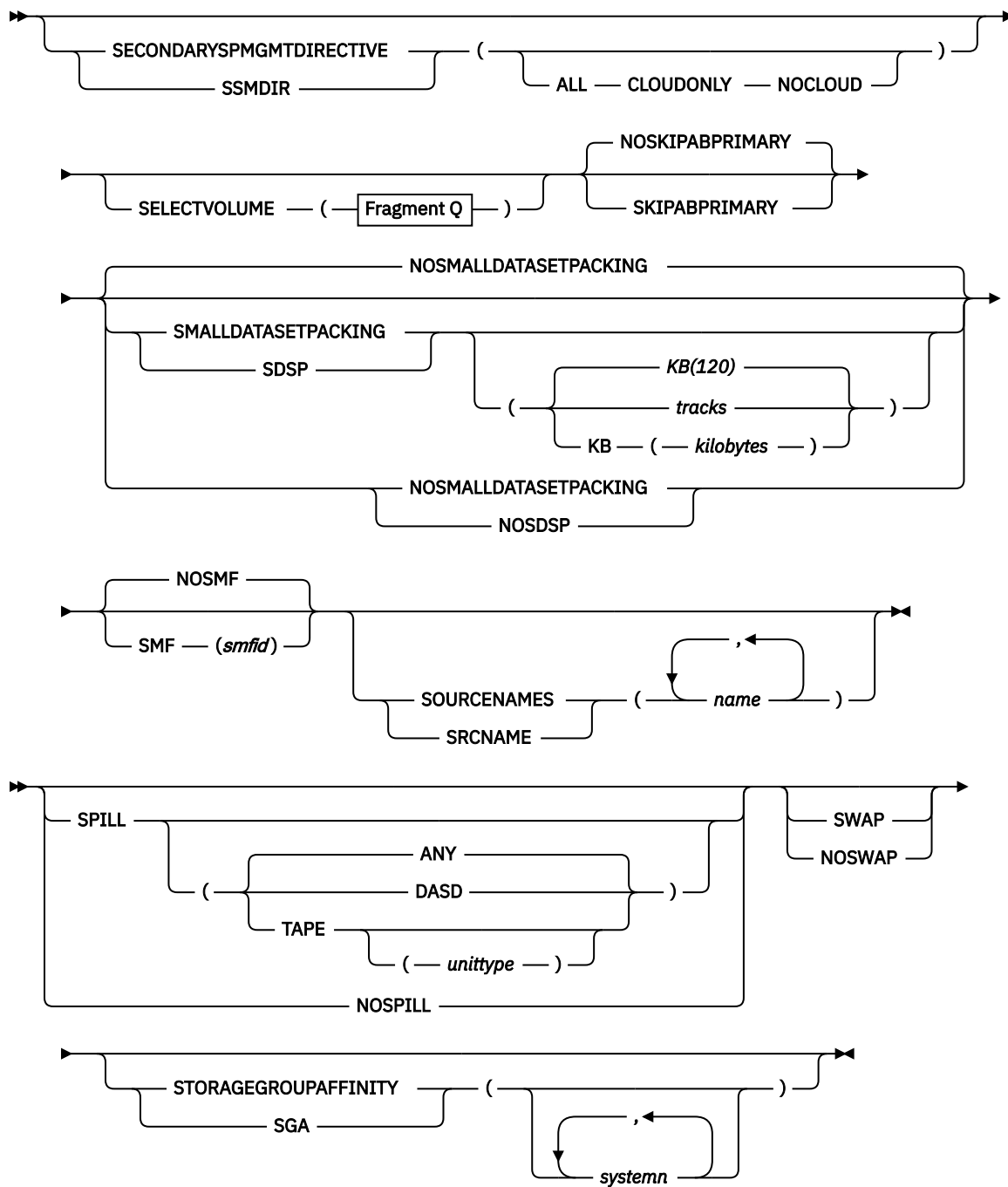


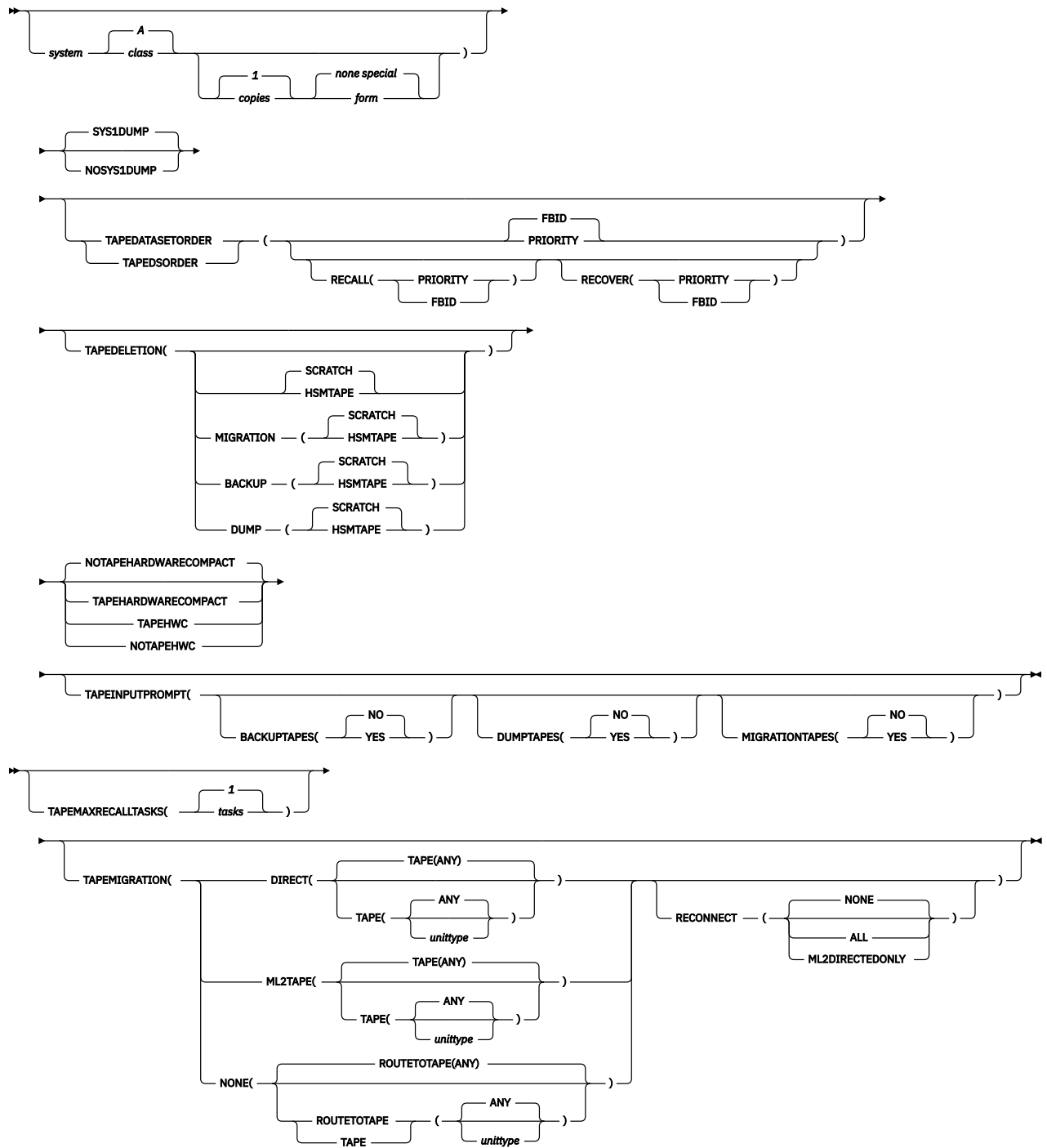


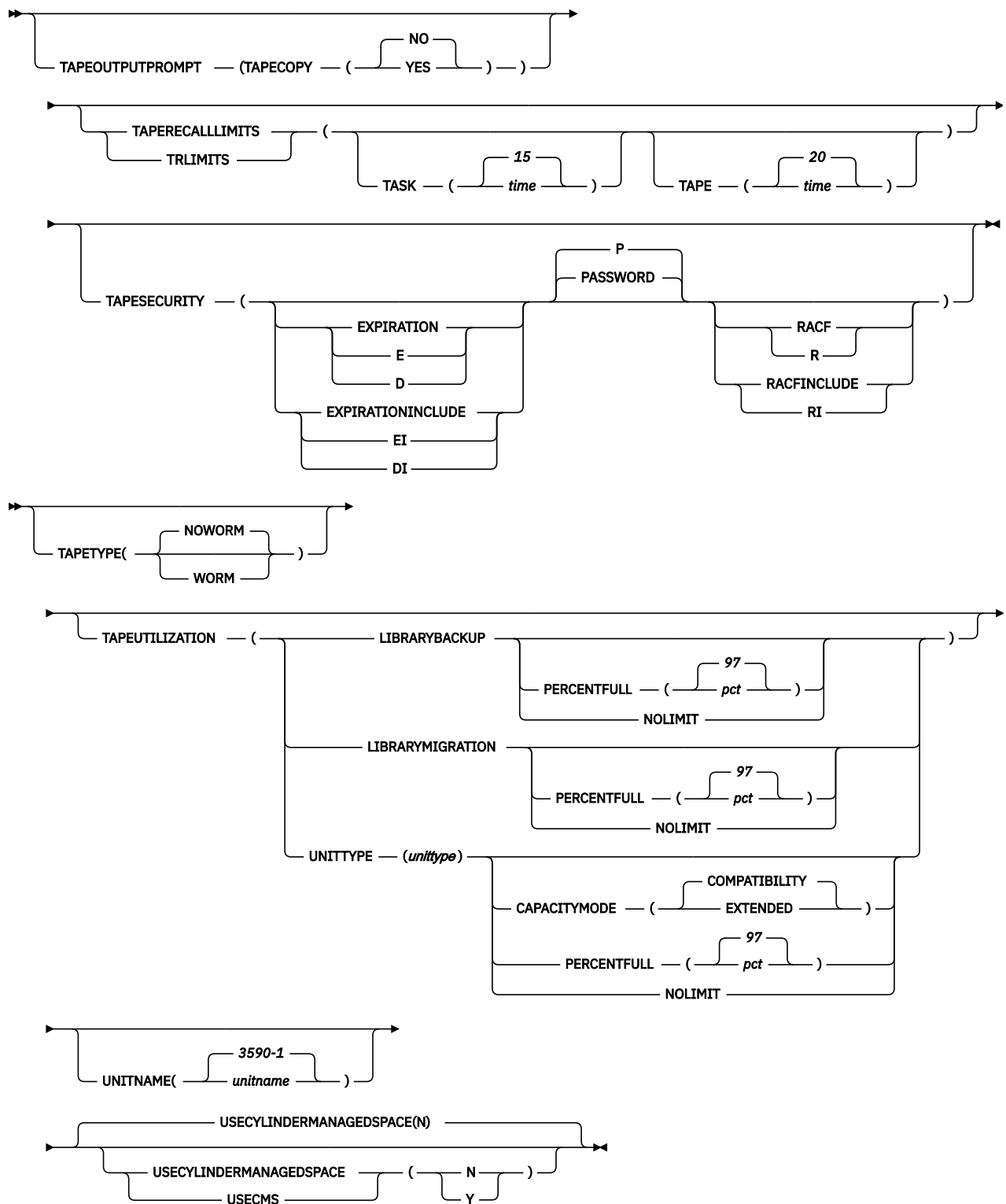


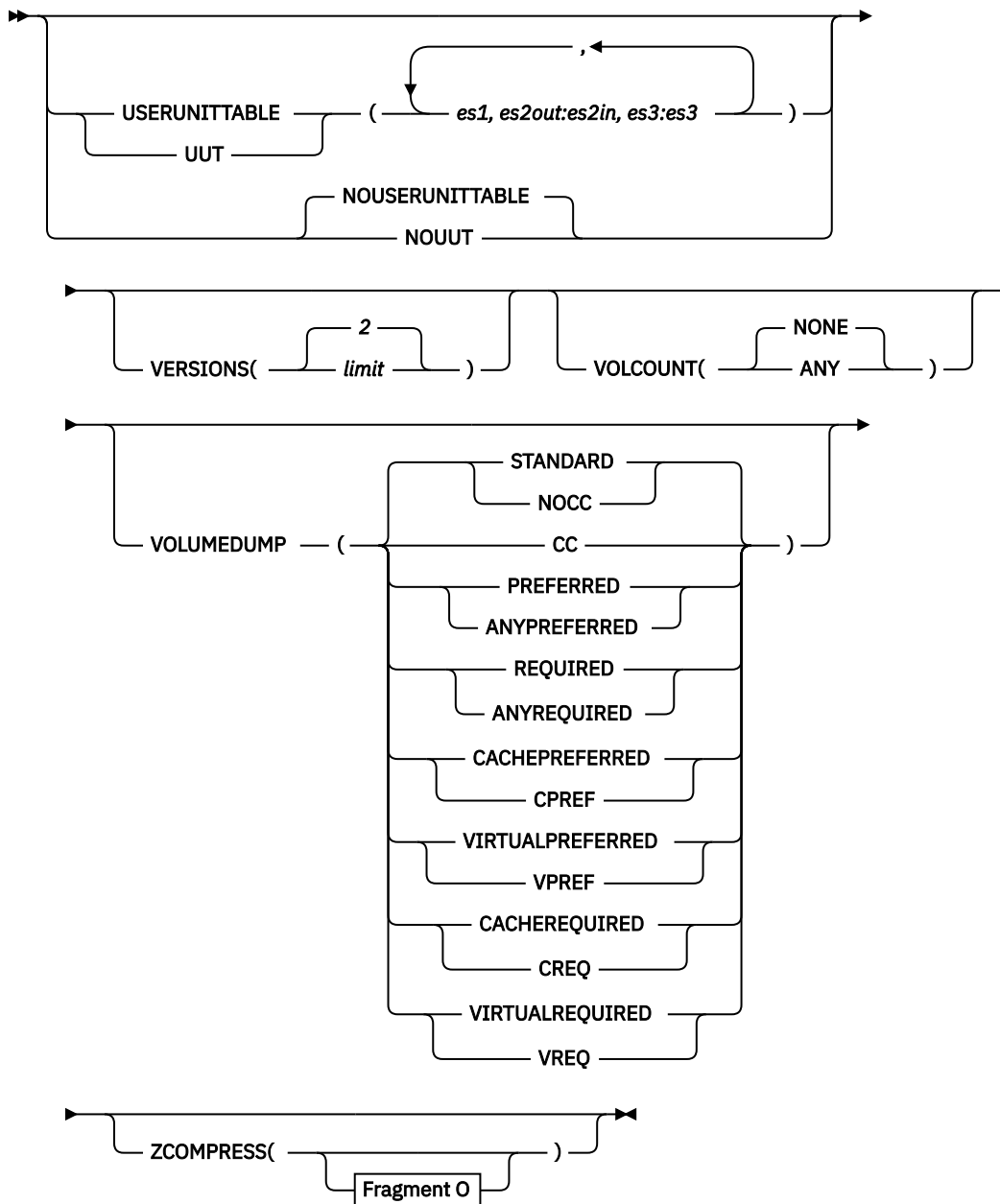




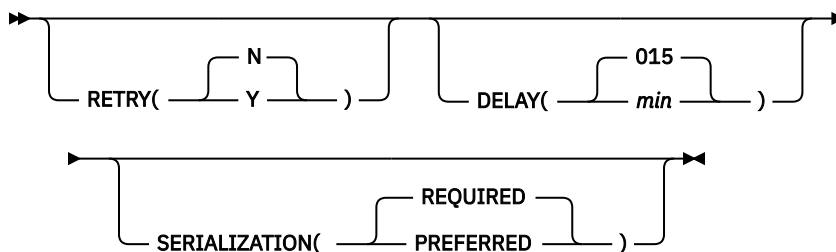




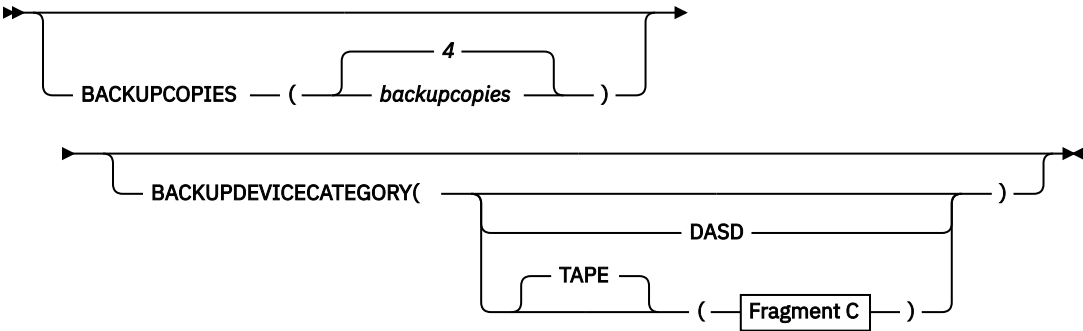




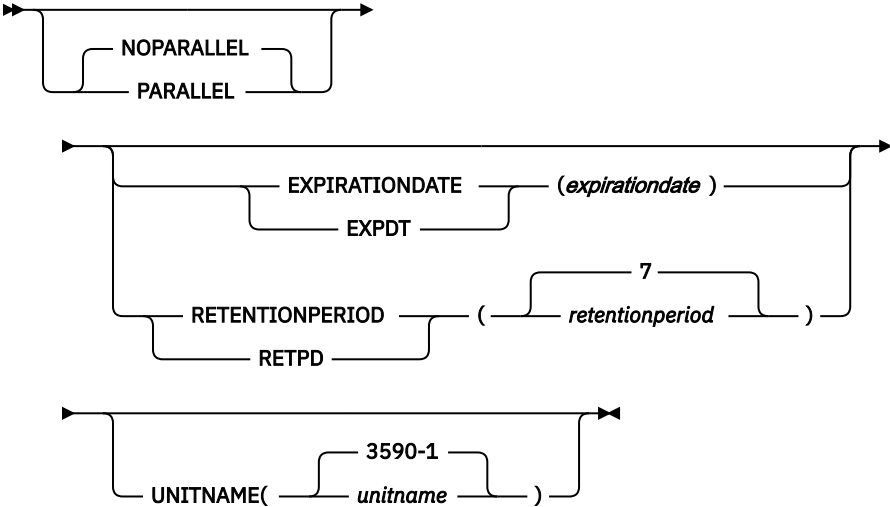
A: INUSE optional parameters



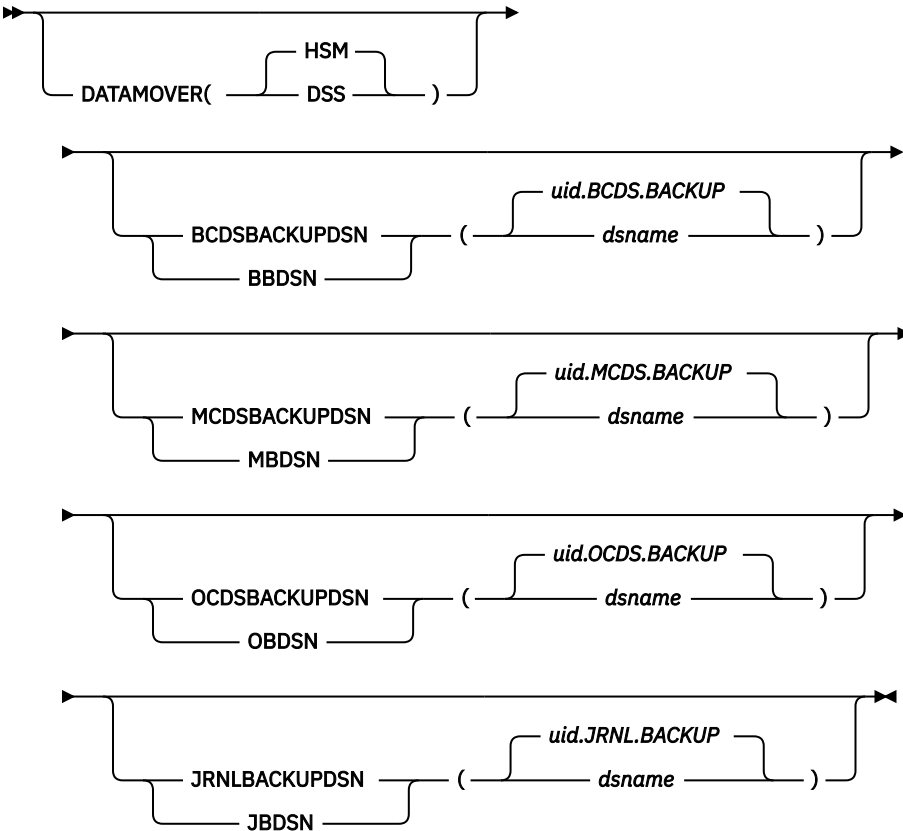
B: CDSVERSIONBACKUP optional parameters

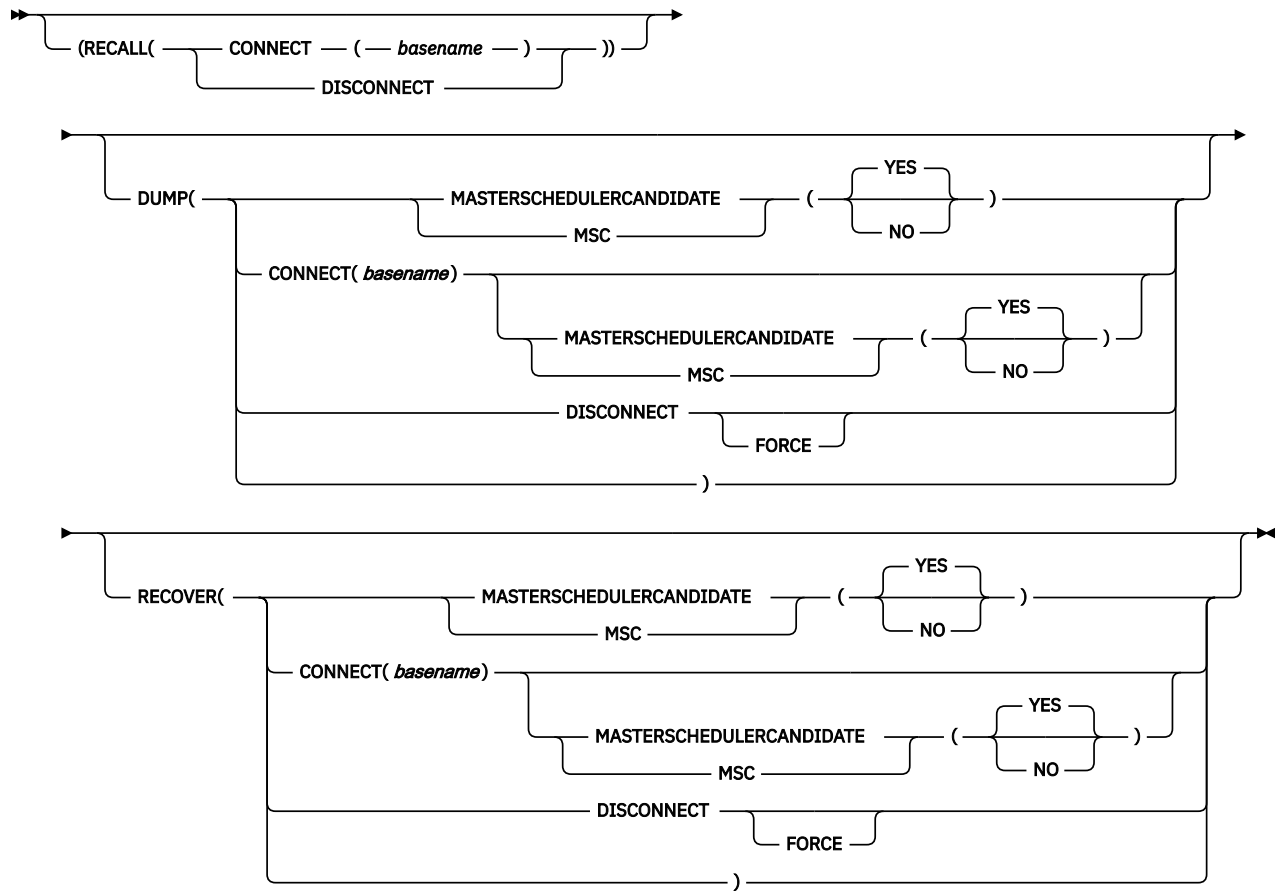
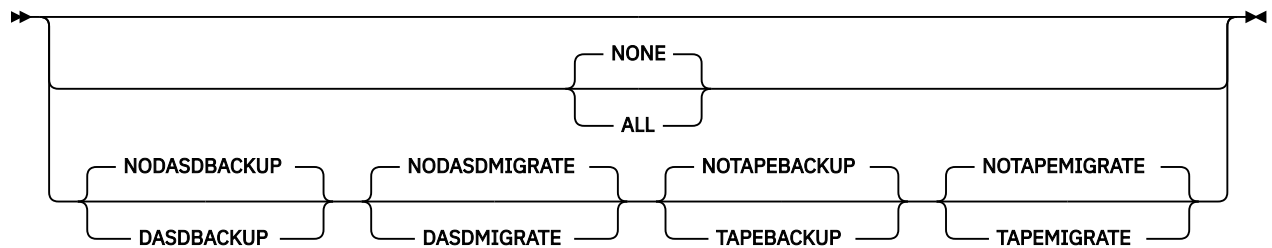
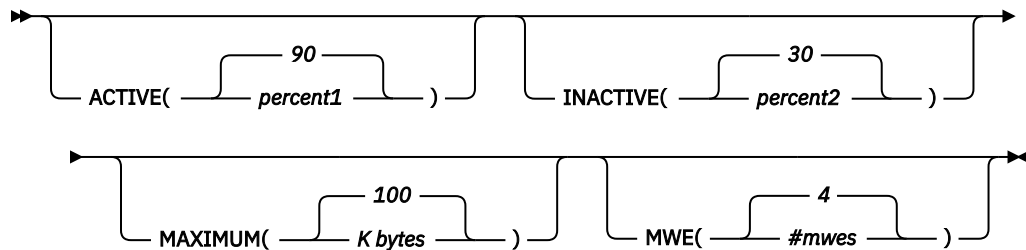


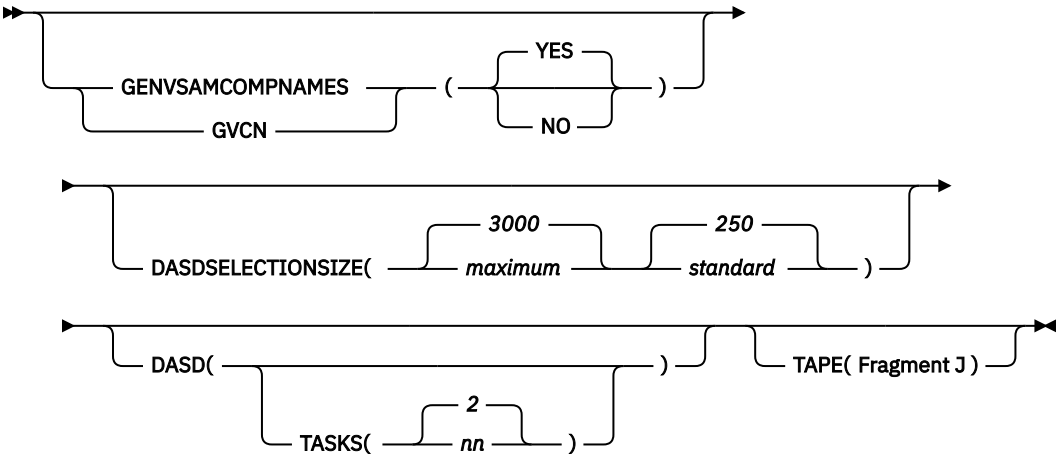
C: TAPE optional parameters



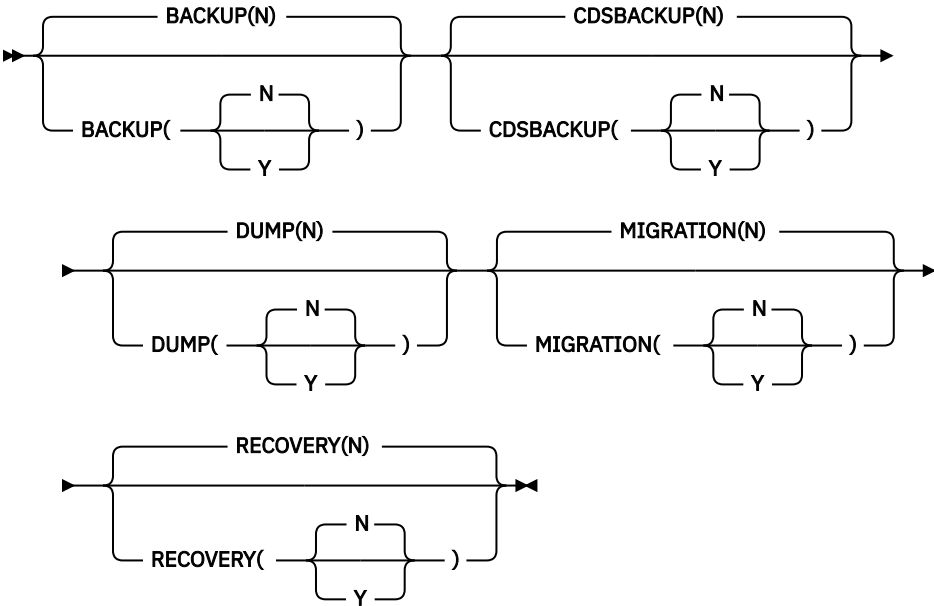
D: CDSVERSIONBACKUP optional parameters



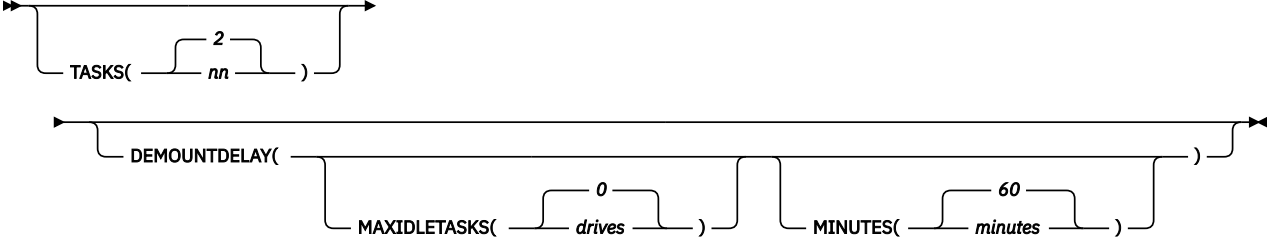
E: COMMONQUEUE optional parameters**F: COMPACT optional parameters****G: CSALIMITS optional parameters****H: DSBACKUP optional parameters**



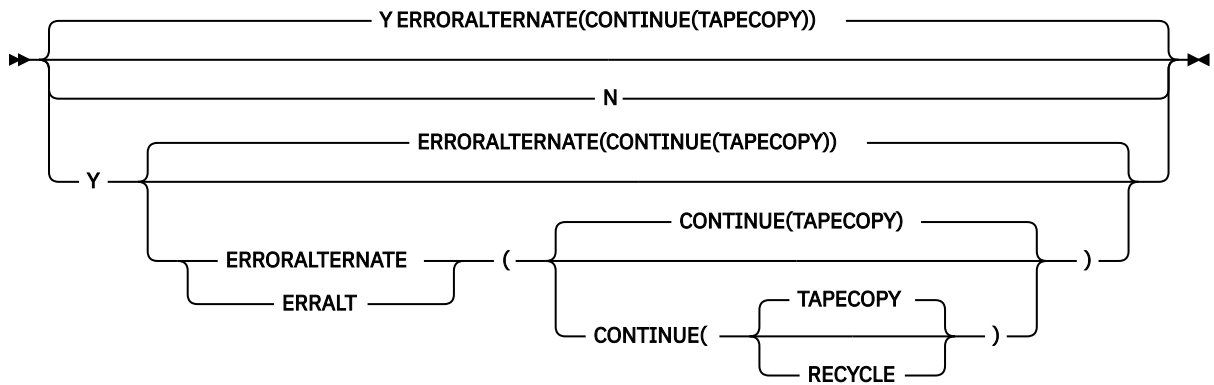
I: DSSXMMODE optional parameters



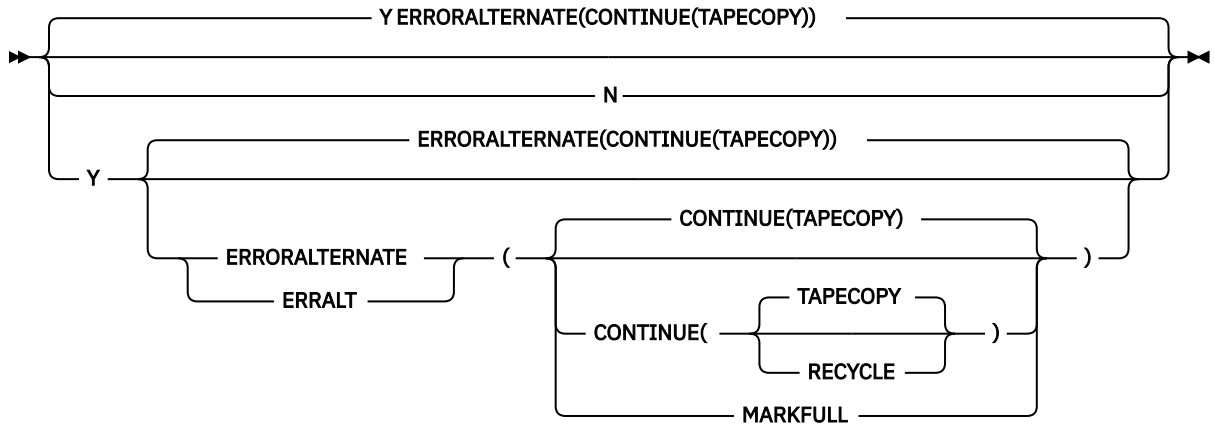
J: TAPE optional parameters



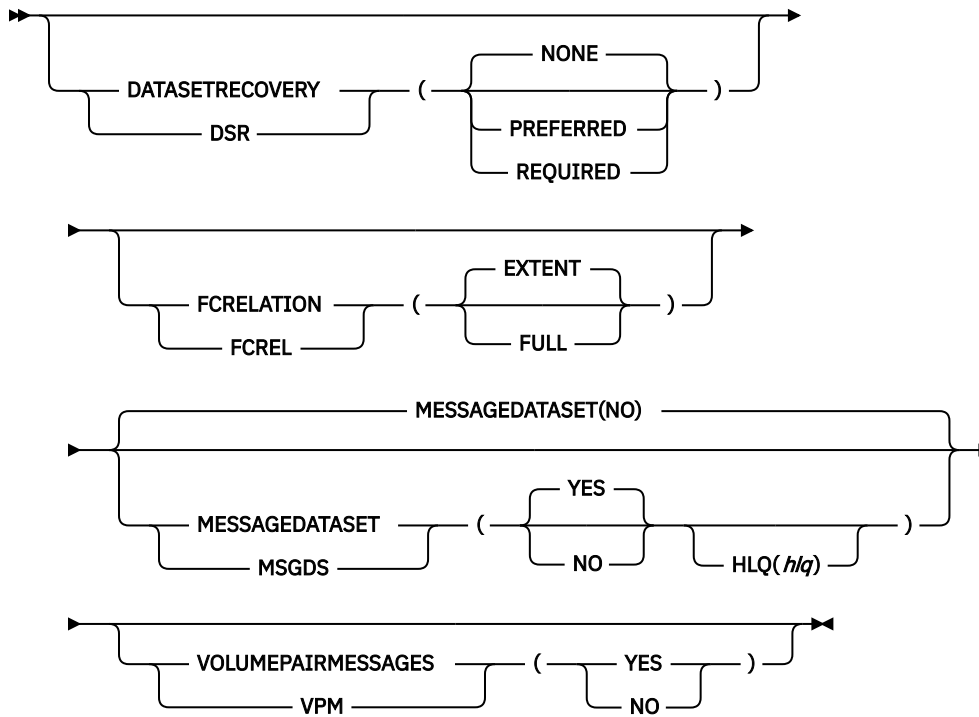
K: DUPLEX BACKUP optional parameters



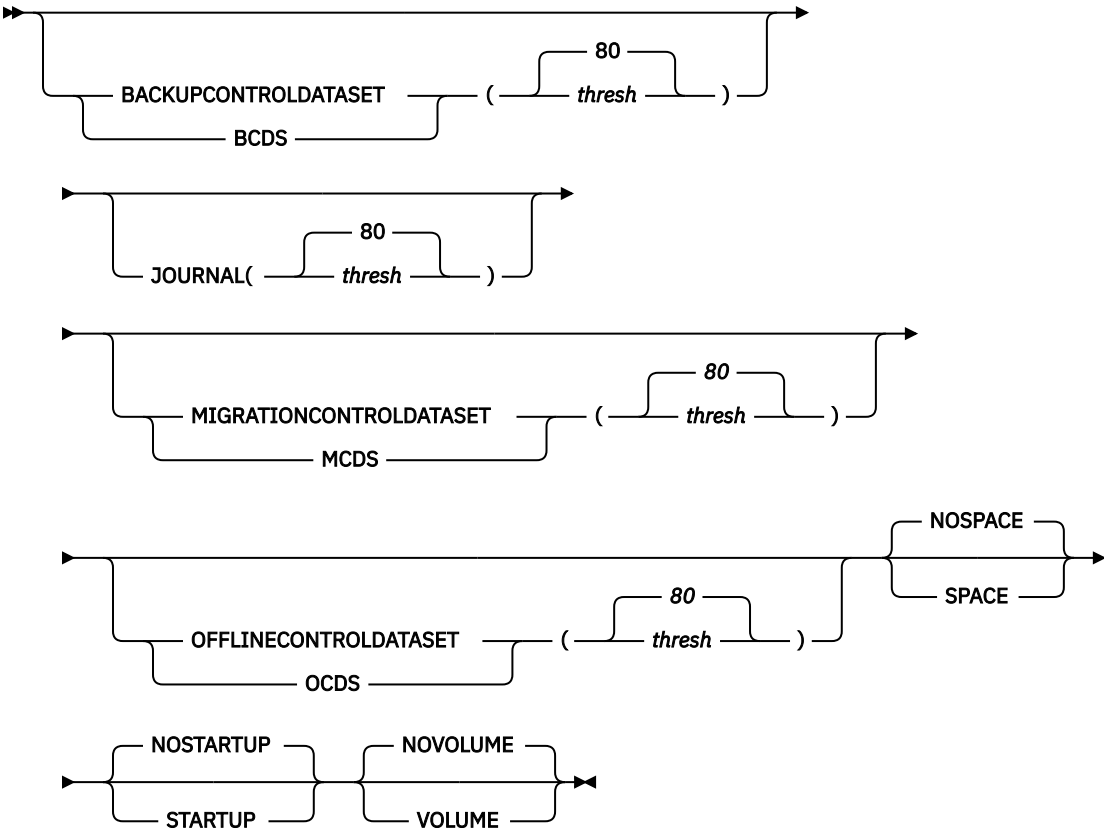
L: DUPLEX MIGRATION optional parameters



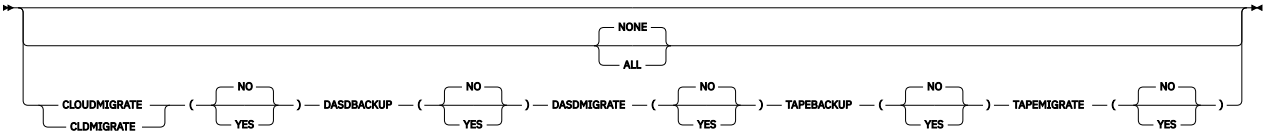
M: FASTREPLICATION optional parameters



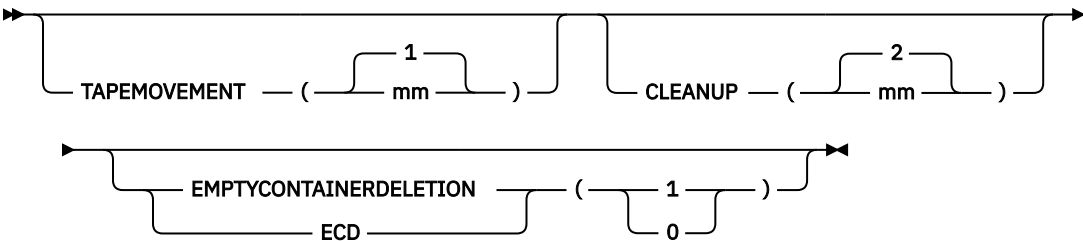
N: MONITOR optional parameters



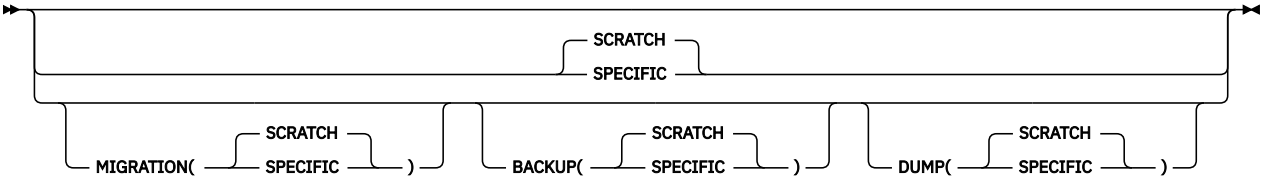
O: ZCOMPRESS optional parameters



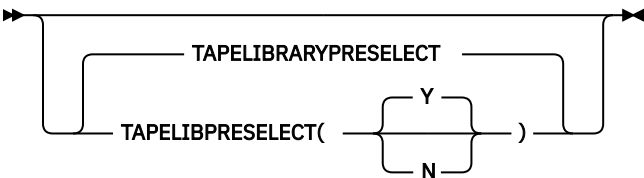
P: MAXSSMTASKS optional parameters



Q: SELECTVOLUME optional parameters



R: TAPELIBRARYPRESELECT optional parameter



Note: In a multiple address space for DFSMSHsm environment, SETSYS commands with ABARS parameters, like ABARSxxx, ARECOVERxxx, or MAXABARSADDRESSSPACE, or the SETSYS CSALIMITS command can be directed only to the host identified as the main host. DFSMSHsm ignores these commands if directed to a host identified as HOSTMODE=AUX.

SETSYS fast-path for DFSMSHsm functions

DFSMSHsm performs two main functions: availability management and space management. Availability management is used to make data available by automatically copying new and changed data sets to backup volumes and DASD volumes to dump tapes. Space management is used to manage DASD space by enabling inactive data sets to be moved off fast-access storage devices thus creating free space for new allocations. DFSMSHsm also provides other supporting functions that are essential to your installation's DFSMSHsm environment.

The following table illustrates the functions performed by DFSMSHsm for availability management, space management and these supporting functions.

Main functions	Supporting functions
Availability management: ABARS Backup Dump Fast replication Recover Space management: Class transitions Expiration Extent reduction Interval migration Migration On-demand migration Recall Recycle Secondary space management	CDS (backup, journaling, and monitoring) Logging and diagnosis MVS environment Security Tape environment

The SETSYS command and its parameters are used to establish a DFSMSHsm environment. When DFSMSHsm is installed, a default set of SETSYS parameters is used. You can specify one or more SETSYS commands in the ARCCMDxx PARMLIB member that is used during the startup of DFSMSHsm or you can issue SETSYS commands with specific parameter values after DFSMSHsm is started.

Table 58 on page 1076, Table 59 on page 1078 and Table 60 on page 1080 lists both the main and supporting functions performed by DFSMSHsm and the SETSYS parameters that affect them. The left column lists the functions and the right column lists the SETSYS parameters that affect these functions. The tables emphasize SMS-managed user data and exclude parameters that relate to non-SMS user data and backup to DASD. Refer to the SETSYS examples at the end of this SETSYS topic for common user scenarios that you can use for your installation.

Table 58. Availability management functions and SETSYS parameters

Availability Management Function	SETSYS Parameter
ABARS environment	ABARSACTLOGMSGVL
	ABARSACTLOGTYPE
	ABARSBUFFERS
	ABARSDELETEACTIVITY
	ABARSKIP
	ABARSPROCNAME
	ABARSTAPES
	ABARSUNITNAME
	EXITOFF (ED and TV)
	EXITON (ED and TV)
	MAXABARSADDRESSSPACE
ABACKUP	ABARSOPTIMIZE
	EXITOFF (BE and M2)
	EXITON (BE and M2)
ARECOVER	ABARSVOLCOUNT
	ARECOVERML2UNIT
	ARECOVERPERCENTUTILIZED
	ARECOVERTGTGDS
	ARECOVERUNITNAME
	EXITOFF (CR and SK)
	EXITON (CR and SK)

Table 58. Availability management functions and SETSYS parameters (continued)	
Availability Management Function	SETSYS Parameter
Backup	AUTOBACKUPSTART
	BACKUP
	BACKUPPREFIX
	COMPACT
	COMPACTPERCENT
	DSBACKUP
	DUMPIO
	DUPLEX
	EXITOFF (BD and SA)
	EXITON (BD and SA)
	INCREMENTALBACKUP
	MAXBACKUPTASKS
	ML1OVERFLOW
	USECYLINDERMANAGEDSPACE
	ZCOMPRESS
	See also the DEFINE command and the EXPIREBV command.
Dump	AUTODUMPSTART
	BACKUP
	BACKUPPREFIX
	COMMONQUEUE
	DUMPIO
	MAXDUMPTASKS
	VOLUMEDUMP
	See also the DEFINE command.
Fast replication	FASTREPLICATION
	MAXCOPYPOOLTASKS
	MAXDSRECOVERTASKS
	MAXTAPEDSRECOVERTASKS
	MESSAGEDATASET

<i>Table 58. Availability management functions and SETSYS parameters (continued)</i>	
Availability Management Function	SETSYS Parameter
Recover	COMMONQUEUE
	CONVERSION
	EXITOFF (CD and RP)
	EXITON (CD and RP)
	MAXDSRECOVERTASKS
	MAXDUMPRECOVERTASKS
	VOLCOUNT

<i>Table 59. Space management functions and SETSYS parameters</i>	
Space Management Function	SETSYS Parameter
Class transitions	CLASSTRANSITIONS
Expiration	EXPIREDDATASETS
Extent reduction	MAXEXTENTS
Interval migration	EVENTDRIVENMIGRATIONQUIESCE
	EXITOFF (MD and SA)
	EXITON (MD and SA)
	INTERVALMIGRATION
	NOINTERVALMIGRATION
	MAXINTERVALTASKS
	MIGRATIONAUTOCLOUD
	STORAGEGROUPAFFINITY
	See also the SETSYS parameters for the Migration function with the exception of the following parameters: EXPIREDDATASETS and PRIMARYSPMGMTSTART.

Table 59. Space management functions and SETSYS parameters (continued)	
Space Management Function	SETSYS Parameter
Migration	COMPACT
	COMPACTPERCENT
	DUMPIO
	DUPLEX
	EXITOFF (MD, MV, and SA)
	EXITON (MD, MV, and SA)
	EXPIREDDATASETS
	MAXINTERVALTASKS
	MAXMIGRATIONTASKS
	MIGRATEPREFIX
	MIGRATEPSMEARLY
	MIGRATIONAUTOCLOUD
	MIGRATIONSUBTASKS
	ML1OVERFLOW
	PRIMARYSPMGMTSTART
	SMALLDATASETPACKING
	STORAGEGROUPAFFINITY
	NOSMALLDATASETPACKING
	TAPEMIGRATION
	USECYLINDERMANAGEDSPACE
	ZCOMPRESS
On-demand migration	EVENTDRIVENMIGRATIONQUIESCE
	MAXINTERVALTASKS
	MIGRATIONAUTOCLOUD
	ODMNOTIFICATIONLIMIT
	ONDEMANDMIGRATION
	STORAGEGROUPAFFINITY

<i>Table 59. Space management functions and SETSYS parameters (continued)</i>	
Space Management Function	SETSYS Parameter
Recall	COMMONQUEUE
	CONVERSION
	EXITON (CD and RP)
	EXITOFF (CD and RP)
	MAXRECALLTASKS
	TAPEMAXRECALLTASKS
	TAPERECALLLIMITS
	VOLCOUNT
Recycle	MAXRECYCLETASKS
	ML2PARTIALSNOTASSOCIATEDGOAL
	ML2RECYCLEPERCENT
	RECYCLEINPUTDEALLOCFREQUENCY
	RECYCLEOUTPUT
	RECYCLEPERCENT
	RECYCLETAKEAWAYRETRY
	RECYCLETAPEALLOCATION
Secondary space management	EXITOFF (MM)
	EXITON (MM)
	MAXSSMTASKS
	MIGRATIONCLEANUPDAYS
	SECONDARYSPMGMTSTART

<i>Table 60. Supporting functions and SETSYS parameters</i>	
Supporting Function	SETSYS Parameter
CDS (backup, journaling, and monitoring)	CDSVERSIONBACKUP
	EXITOFF (CB)
	EXITON (CB)
	JOURNAL
	NOJOURNAL
	MONITOR

Table 60. Supporting functions and SETSYS parameters (continued)	
Supporting Function	SETSYS Parameter
Logging and diagnosis	ACTLOGMSGLVL
	ACTLOGTYPE
	DEBUG
	NODEBUG
	EMERGENCY
	NOEMERGENCY
	FASTREPLICATION(MESSAGEDATASET)
	MESSAGEDATASET
	MONITOR
	PDA
	SMF
	SYSOUT
	SYS1DUMP
MVS environment	CSALIMITS
	EXITOFF (IN and SD)
	EXITON (IN and SD)
	JES2
	JES3
	NOSWAP
	PLEXNAME
	PROMOTE
	DFHSMDATASETSERIALIZATION
	USERDATASETSERIALIZATION
Security	ACCEPTPSCBUSERID
	NOACCEPTPSCBUSERID
	ERASEONSCRATCH
	NOERASEONSCRATCH
	PROFILEBACKUP
	NOPROFILEBACKUP
	RACFIND
	NORACFIND
	TAPESECURITY

Table 60. Supporting functions and SETSYS parameters (continued)	
Supporting Function	SETSYS Parameter
Tape environment	COMPACT
	DISASTERMODE
	EXITOFF (TD, TE, and TV)
	EXITON (TD, TE, and TV)
	EXTENDEDTCOC(Y N)
	INPUTTAPEALLOCATION
	MOUNTWAITTIME
	OUTPUTTAPEALLOCATION
	PARTIALTAPE
	SELECTVOLUME
	TAPEDELETION
	TAPEHARDWARECOMPACT
	TAPEINPUTPROMPT
	TAPELIBRARYPRESELECT
	TAPEMIGRATION
	TAPEOUTPUTPROMPT
	TAPESPANSIZE
	TAPETYPE
	TAPEUTILIZATION
	UNITNAME
	USERUNITTABLE
	NOUSERUNITTABLE
	ZCOMPRESS

Relationship of SMS to SETSYS parameters

Table 62 on page 1083 lists the relationship of SETSYS parameters to SMS-managed DASD volumes and data sets. Parameters that apply equally to both SMS-managed and non-SMS-managed DASD volumes and data sets are not included.

All SETSYS parameters that affect tape data set names, tape volumes, and tape unit types are applicable to SMS-managed tape environments, and might have different meanings from non-SMS-managed tape environments, depending on the SMS implementation.

Table 61 on page 1082 list the abbreviations and their meaning are used in Table 62 on page 1083.

Table 61. SMS relationship abbreviations and their meaning		
Abbreviations	Meaning	Description
APP	Applicable	Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

Table 61. SMS relationship abbreviations and their meaning (continued)

Abbreviations	Meaning	Description
CAPP	Conditionally Applicable	Parameter applies in some instances to SMS-managed DASD volumes or data sets.
NAPP	Not Applicable	Parameter does not apply to SMS-managed DASD volumes or data sets.

Table 62. SETSYS parameter relationship to SMS-managed volumes and data sets

SMS relationship	SETSYS optional parameters
CAPP	ABARVOLCOUNT
NAPP	ARECOVERPERCENTUTILIZED
APP	BACKUP and NOBACKUP
APP	CLASSTRANSITIONS
APP	COMPACT
APP	COMPACTPERCENT
CAPP	CONVERSION and NOCONVERSION
NAPP	DAYS
APP	DEBUG and NODEBUG
APP	DFHSMDATASETSERIALIZATION and USERDATASETSERIALIZATION
APP	EMERGENCY and NOEMERGENCY
APP	ERASEONSCRATCH and NOERASEONSCRATCH
APP	EXITOFF
APP	EXITON
CAPP	EXPIREDDATASETS
APP	FASTREPLICATION
NAPP	FREQUENCY
APP	INCREMENTALBACKUP
APP	INTERVALMIGRATION and NOINTERVALMIGRATION
APP	MAXCOPYPOOLTASKS
APP	MAXDSRECOVERTASKS
APP	MAXEXTENTS
APP	MAXTAPEDSRECOVERTASKS
APP	MIGRATIONCLEANUPDAYS
NAPP	MIGRATIONLEVEL1DAYS
APP	OBJECTNAMES
CAPP	ODMNOTIFICATIONLIMIT
CAPP	ONDEMANDMIGRATION

Table 62. SETSYS parameter relationship to SMS-managed volumes and data sets (continued)	
SMS relationship	SETSYS optional parameters
APP	PROFILEBACKUP and NOPROFILEBACKUP
APP	RACFIND and NORACFIND
NAPP	RECALL
APP	REMOVECOMPACTNAMES
NAPP	SCRATCHFREQUENCY
APP	SKIPABPRIMARY and NOSKIPABPRIMARY
APP	SMF and NOSMF
CAPP	TAPEMIGRATION
NAPP	TAPEOUTPUTPROMPT
APP	USERUNITTABLE and NOUSERUNITTABLE
NAPP	VERSIONS
CAPP	VOLCOUNT
APP	ZCOMPRESS

Summary of DFSMSHsm default values

Table 63 on page 1084 is a summary of the defaults DFSMSHsm uses if you never specify a SETSYS command during or after startup.

Table 63. Summary of DFSMSHsm Default Values	
DFSMSHsm default value	Result
ABARSACTLOGMSGGLVL(FULL)	All DFSMSdss messages that are written to the ABARS activity log.
ABARSACTLOGTYPE SYSOUT(A)	ABARS activity logs are allocated to the SYSOUT class A.
ABARSDELETEACTIVITY(N)	The ABARS activity log is not automatically deleted.
ABARSKIP(NOPPRC NOXRC)	ABACKUP processing will not skip data sets on volumes that are protected by peer-to-peer remote copy or extended remote copy.
ABARSOPTIMIZE(3)	DFSMSHsm reads five tracks at a time.
ABARSPROCNAME(DFHSMABR)	DFSMSHsm uses DFHSMABR as the base procedure name on the internal START command to start the secondary address space.
ABARSTAPES(STACK)	The default is STACK and DFSMSHsm stacks output files onto a minimum number of tape volumes.
ABARSUNITNAME(3590-1)	ABARS selects 3590-1 units when invoking aggregate backup or aggregate recovery.
ABARSVOLCOUNT(NONE)	The default is NONE and DFSMSHsm does not pass the VOLCOUNT parameter to DFSMSdss.
ACTLOGMSGGLVL(FULL)	Messages are generated and logged for all activities.

Table 63. Summary of DFSMSHsm Default Values (continued)	
DFSMSHsm default value	Result
ACTLOGTYPE SYSOUT(A)	If you specified SYSOUT without a class, the output is class A, one copy, no special forms.
ARECOVERML2UNIT(3590-1)	ARECOVER selects 3590-1 units when recovering data sets to ML2 tape.
ARECOVERPERCENTUTILIZED(80)	When specifying this parameter, DFSMSHsm uses a default of 80% to fill non-SMS DASD recovery volumes. Otherwise, the valid values you can use range from 1 to 100.
ARECOVERTGTGDS(SOURCE)	The default is SOURCE.
ARECOVERUNITNAME(3590-1)	DFSMSHsm uses a unit of 3590-1 for aggregate recovery of data sets recovered to tape.
AUTOBACKUPSTART(0)	DFSMSHsm does not run automatic backup.
AUTODUMPSTART(0)	DFSMSHsm does not run automatic dump.
BACKUP(INUSE(RETRY(N) DELAY(15) SERIALIZATION(REQUIRED)))	DFSMSHsm does not retry a backup attempt that fails because the data set is in use. If RETRY(Y) is later specified, DFSMSHsm delays for 15 minutes before attempting a retry, and requires the data set to be serialized during the retry.
BACKUPCOPIES(4)	The default value is read from the multiple-host processor control record (MHCR). The record is updated with values that are specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, DFSMSHsm keeps four copies of the journal and control data sets.
BACKUPDEVICECATEGORY(TAPE)	The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, DFSMSHsm backs up the control data sets to tape.
BACKUPPREFIX	DFSMSHsm uses the UID you specified in the DFSMSHsm startup procedure.
BCDSBACKUPDSN(uid.BCDS.BACKUP)	The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, the backup data set name for the backup control data set is uid.BCDS.BACKUP.

Table 63. Summary of DFSMSHsm Default Values (continued)

DFSMSHsm default value	Result
CDSVERSIONBACKUP parameters listed separately: BACKUPCOPIES BACKUPDEVICECATEGORY BCDSBACKUPDSN DENSITY JRNBACKUPDSN MCDSBACKUPDSN NOPARALLEL OCDSBACKUPDSN RETENTION PERIOD UNITNAME	<p>The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, DFSMSHsm makes multiple backup versions of the control data sets and the journal data set.</p>
CLASSTRANSITIONS(EVENTDRIVENMIGRATION(Y SERIALIZATIONEXIT(N)))	<p>Class transitions will be performed during event-driven migration, but user data set serialization error exits will <i>not</i> be used.</p>
COMPACT(NONE)	<p>DFSMSHsm does not compact any data sets.</p>
COMPACTPERCENT(40)	<p>If you specify that DFSMSHsm compact data sets, DFSMSHsm uses a default of 40% to determine whether a data set will be compacted in the future. Otherwise, DFSMSHsm ignores this parameter.</p>
CSALIMITS(MWE(4) MAXIMUM(100) ACTIVE(90) INACTIVE(30))	<p>DFSMSHsm limits the common service area storage for MWE allocation. The NOWAIT MWE limit is 4, the maximum limit is 100 KB, the active limit is 90 KB, and the inactive limit is 30 KB.</p>
DATAMOVER	<p>The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, DFSMSHsm default for the type of data movement used when performing CDSVERSIONBACKUP, is DATAMOVER(HSM).</p>
DAYS(1) or DAYS(2)	<p>In a single-DFSMSHsm-host environment, the data sets must be at least one day old. In a multiple-DFSMSHsm-host environment, the data sets must be at least two days old.</p>
DFHSMDATASETSERIALIZATION	<p>Data set serialization of system resources is not provided by system facilities.</p>
DISASTERMODE(N)	<p>If you specify N, DFSMSHsm will request the original tape volume and makes no substitution.</p>

Table 63. Summary of DFSMSHsm Default Values (continued)	
DFSMSHsm default value	Result
DSBACKUP	For DASDSELECTIONSIZE, the default for <i>maximum</i> is 3000 KB and <i>standard</i> is 250 KB. For DASD(TASKS) or TAPE(TASKS), the default is 2. For DEMOUNTDELAY, the default for MAXIDLETASKS is 0 and MINUTES is 60. If GENVSAMCOMPNames(YES) GVCN(YES), and <i>dsname</i> is a VSAM base cluster and <i>newdsname</i> is either migrated or uncataloged, DFSMSHsm will process the request, and default names will be assigned to the VSAM data and index components. If GVCN(NO), and <i>dsname</i> is a base cluster and <i>newdsname</i> is migrated or uncataloged, the HBACKDS command will fail. If GVCN(YES NO) is specified, and <i>newdsname</i> is cataloged and not migrated, or if the data set to be backed up is non-VSAM, the HBACKDS command will fail.
DSSXMMODE(N)	DFSMSDss will not be loaded into its own address space for the DFSMSHsm function being invoked. Instead, DFSMSDss will be loaded in the DFSMSHsm address space.
DUMPIO(1,1)	DFSMSDss reads one track at a time.
DUPLEX	If you do not specify DUPLEX and either of its subparameters, duplexing does not occur.
EXITOFF	Installation exits remain in their current state.
EXITON	No installation exits are active.
EXPIREDDATASETS(NOSCRATCH)	DFSMSHsm ignores the expiration date of a data set and processes it as if the expiration date were not reached.
EXTENDEDITOC(N)	DFSMSHsm does not attempt to use extended TTOCs for migration tapes or backup tapes.
FASTREPLICATION(DATASETRECOVERY(NONE))	Traditional copy methods are used for data set level recovery. Fast replication is not used.
FASTREPLICATION(MESSAGE DATASET(NO))	DFSMSHsm does not log fast replication messages in message data sets
FASTREPLICATION(FCRELATION(EXTENT))	Extent-level FlashCopy relationships will be established during backup and recovery. Free space is not flashed.
FASTREPLICATION(VOLUMEPAIRMESSAGES(NO))	FlashCopy volume pairing debug message ARC1809I is disabled.
FREQUENCY(0)	DFSMSHsm backs up data sets every time volume backup runs.
INCREMENTALBACKUP(ORIGINAL)	DFSMSHsm creates an initial backup version (if one does not exist) for all non-VSAM and integrated catalog facility VSAM data sets regardless of the change bit in the format 1 DSCB.
INPUTTAPEALLOCATION(NOWAIT)	DFSMSHsm does not ask dynamic allocation to wait for an input tape unit to be allocated. Recall and recovery use input tape units.
INTERVALMIGRATION	DFSMSHsm does interval migration.

<i>Table 63. Summary of DFSMSHsm Default Values (continued)</i>	
DFSMSHsm default value	Result
ISSUEARC1902	Issue of ARC1902I is disabled and the delay time is set to default value of 7200 secs
JES2	DFSMSHsm uses job entry subsystem 2.
JOURNAL(SPEED)	DFSMSHsm puts the journal entries on a queue but does not wait for the record to be written before DFSMSHsm continues processing.
JRNLBKUPDSN(uid.JRNL.BACKUP)	The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, the backup data set name for the journal control data set is uid.JRNL.BACKUP.
MAXABARSADDRESSSPACE(1)	DFSMSHsm runs aggregate backup or recovery in one secondary address space.
MAXBACKUPTASKS(2)	DFSMSHsm concurrently runs two volume backup tasks.
MAXCOPYPOOLTASKS(FRBACKUP(15) FRRECOV(15) DSS(24))	DFSMSHsm concurrently processes a limited number of DFSMSdss invocation tasks for FRBACKUP and FRRECOV. The FRBACKUP limit is 15 and the FRRECOV limit is 15. The DSS limit is the number of volumes that DFSMSHsm passes to DFSMSdss to process concurrently per DFSMSdss invocation.
MAXDSRECOVERTASKS(2)	DFSMSHsm concurrently runs two data set recovery tasks.
MAXDSTAPERECOVERTASKS	DFSMSHsm uses the value of MAXDSRECOVERTASKS.
MAXDUMPRECOVERTASKS(1)	One volume recovery from dump task is processed at a time.
MAXDUMPTASKS(2)	DFSMSHsm concurrently processes two volume dump tasks.
MAXEXTENTS(0)	DFSMSHsm does not do extent reduction for active non-VSAM data sets.
MAXINTERVALTASKS(2)	Defaults to the value of MAXMIGRATIONTASKS.
MAXMIGRATIONTASKS(2)	DFSMSHsm defaults to a maximum of 2 migration tasks that can run concurrently.
MAXRECALLTASKS(15)	DFSMSHsm concurrently processes 15 recall tasks.
MAXRECYCLETASKS(2)	DFSMSHsm concurrently processes two tasks, each with one input and one output unit during RECYCLE processing
MAXSSMTASKS(CLEANUP(2) TAPEMOVEMENT(1))	DFSMSHsm concurrently processes a limited number of secondary space management tasks. The statistics and migration cleanup tasks limit is 2 and the automatic tape migration task limit is 1.

Table 63. Summary of DFSMSHsm Default Values (continued)	
DFSMSHsm default value	Result
MCDSBACKUPDSN(uid.MCDS.BACKUP)	The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, the backup data set name for the migration control data set is uid.MCDS.BACKUP.
MIGRATEPREFIX	DFSMSHsm uses the UID you specified in the DFSMSHsm startup procedure.
MIGRATIONAUTOCLOUD(NOCLOUD)	Eligible data sets are migrated to the non-cloud tier indicated by the management class.
MIGRATIONAUTOCLOUD (SELECTIONTHRESHOLD(40))	If the percentage of data sets that are migrated from a volume to the cloud is not greater than 40, this volume will be selected without checking for "free" threads. Otherwise, the volume will be selected if a "free CEC thread can be found to process the migration.
MIGRATIONCLEANUPDAYS(10 30 3 0)	DFSMSHsm keeps MCDS data set records for recalled data sets for 10 days. DFSMSHsm keeps daily and volume statistics records for 30 days. DFSMSHsm keeps MCDS data set records for reconnectable data sets for 3 days beyond their predicted remigration date. DFSMSHsm will not create "Retained" MCA records or keep track of old cloud migration copies by default.
MIGRATIONLEVEL1DAYS(60)	DFSMSHsm does not migrate the data set from a migration level 1 volume to a migration level 2 volume if someone has recalled the data set within the last 60 days.
MIGRATIONSUBTASKS(NO)	DFSMSHsm does not use migration subtasks.
MIGUNITNAME(3590-1)	DFSMSHsm uses a unit of 3590-1 to mount scratch tapes during migration.
ML1OVERFLOW(DATASSETSIZE(2000000))	If the calculated size of a data set being backed up or migrated is equal to or greater than 2000000 K bytes, DFSMSHsm prefers the OVERFLOW ML1 volume with the least amount of free space that still has enough remaining space for the data set. If the calculated size of the data set is less than 2000000 K bytes, DFSMSHsm prefers the NOOVERFLOW ML1 volume with the maximum amount of free space and least number of users.
ML2PARTIALSNOTASSOCIATEDGOAL	If the number of ML2 partial tapes not associated with any migration or recycle output task is more than 10 at the time of a RECYCLE ML2 command, DFSMSHsm recycles the least full of the excess, leaving the 10 with the highest percentages written.
ML2RECYCLEPERCENT	This parameter has no default, however, if it is not specified, the default functionally used is the value specified for RECYCLEPERCENT. If RECYCLEPERCENT is not specified, its default is 20%.

Table 63. Summary of DFSMSHsm Default Values (continued)

DFSMSHsm default value	Result
MONITOR(BACKUPCONTROLDATASET(80) JOURNAL(80) MIGRATIONCONTROLDATASET(80) OFFLINECONTROLDATASET(80) NOSPACE NOSTARTUP NOVOLUME)	DFSMSHsm prints space and volume information only in the DFSMSHsm log. DFSMSHsm sends warning messages to the operator when the journal data set or any control data set is 80% full.
MOUNTWAITTIME(15)	DFSMSHsm waits 15 minutes for an operator to mount the tape before it issues another message asking if the tapes can be mounted.
NOACCEPTPSCBUSERID	DFSMSHsm does not attempt to retrieve the user ID for TSO batch requests.
NOBACKUP	DFSMSHsm does not back up or recover any data sets.
NOCONVERSION	DFSMSHsm does not reblock data sets when it recalls or recovers them.
NODEBUG	DFSMSHsm processes all data sets and volumes.
NOEMERGENCY	DFSMSHsm processes all data sets normally.
NOERASEONSCRATCH	DFSMSHsm does not check for erase of DFSMSHsm-owned data sets. DFSMSHsm does not ask RACF for the erase status of the user's data set when backup versions and migration copies are scratched from DFSMSHsm-owned DASD volumes. The data set is deleted but the DASD residual data is not overwritten by data management.
NOOPTIMUMDASDBLOCKING	DFSMSHsm writes to DFSMSHsm-owned DASD in 2 KB blocks.
NOPARALLEL	The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, DFSMSHsm will backup the control data sets and the journal serially.
NOREQUEST	DFSMSHsm does not ask operator permission when starting automatic volume space management, automatic secondary space management, automatic backup, and automatic dump at the requested time.
NOSKIPABPRIMARY	DFSMSHsm automatically backs up data sets on primary volumes.
NOSMALLDATASETPACKING	DFSMSHsm does not pack small data sets into SDSP data sets on migration level 1 volumes.
NOSMF	DFSMSHsm does not write SMF records.
NOTAPEHARDWARECOMPACT	Improved data recording capability (IDRC) is not used when 3480X is used as the output device.
NOUNLOAD	DFSMSHsm does not unload your virtual volumes.

Table 63. Summary of DFSMSHsm Default Values (continued)	
DFSMSHsm default value	Result
NOUSERUNITTABLE	DFSMSHsm does not use esoteric tape unit names.
OBJECTNAMES	DFSMSHsm uses the general compaction table.
OCDSBACKUPDSN(uid.OCDS.BACKUP)	The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, the backup data set name for the offline control data set is uid.OCDS.BACKUP.
ODMNOTIFICATIONLIMIT(100)	Message ARC1901E is issued to the operator's console when the number of volumes concurrently queued for on-demand migration exceeds 100 volumes.
ONDEMANDMIGRATION(N)	On-demand migration is disabled.
OUTPUTTAPEALLOCATION(NOWAIT)	DFSMSHsm does not ask dynamic allocation to wait for an output tape unit to be allocated. Migration, backup, and dump use output tape units.
PARTIALTAPE(REUSE)	DFSMSHsm marks the tape volume full only when the volume reaches the percent full specified with the SETSYS TAPEUTILIZATION parameter.
PDA(ON)	DFSMSHsm defaults to ON if the subparameters NONE or OFF are not specified. Tracing begins at DFSMSHsm startup and continues until the OFF subparameter is specified or DFSMSHsm is stopped or ABENDs.
PLEXNAME(PLEX0)	DFSMSHsm uses a default HSMplex name of ARCPLEX0.
PRIMARYSPMGMTSTART(0)	DFSMSHsm does not run automatic primary space management.
PROFILEBACKUP	DFSMSHsm creates a backup RACF discrete profile for every cataloged RACF-indicated data set that it backs up.
PROMOTE	No host in an HSMplex is eligible to take over the Primary or Secondary Space Management functions of a failed host.
RACFIND	DFSMSHsm puts RACF-indication on migration copies and backup versions.
RECALL(ANYSTORAGEVOLUME)	DFSMSHsm recalls nonpooled data sets to an online primary volume with the storage use attribute and primary volume attribute of automatic recall.
RECYCLEINPUTDEALLOCFREQUENCY(BACKUP MIGRATION)	DFSMSHsm uses zero (0) for both the BACKUP and MIGRATION subparameters. This will keep an input unit allocated until the recycle processing has completed or as long as input can be read.
RECYCLEOUTPUT(BACKUP MIGRATION)	DFSMSHsm uses any available tape.
RECYCLEPERCENT(20)	The tape volume cannot contain more than 20% valid data when DFSMSHsm generically recycles a volume.

<i>Table 63. Summary of DFSMSHsm Default Values (continued)</i>	
DFSMSHsm default value	Result
RECYCLETAKEAWAYRETRY(NO)	DFSMSHsm does not automatically generate a new RECYCLE command for a tape when the original recycle must terminate due to the takeaway process, or when the tape is in use by the another DFSMSHsm task.
RECYCLETAPEALLOCATION(NOWAIT)	DFSMSHsm does not ask dynamic allocation to wait for a tape unit to be allocated during recycle processing.
RETENTIONPERIOD(7)	The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, DFSMSHsm keeps the backup copies of the DFSMSHsm control data sets for 7 days.
SCRATCHFREQUENCY(9999)	DFSMSHsm keeps list data sets for 9,999 days before it scratches them.
SECONDARYSPMGMTSTART(0)	DFSMSHsm does not run automatic secondary space management.
SELECTVOLUME(SCRATCH)	DFSMSHsm chooses scratch tapes when an end-of-volume occurs on a tape volume while DFSMSHsm is using it.
SOURCENAMES	DFSMSHsm uses the general compaction table.
SPILL(ANY)	DFSMSHsm does spill processing on any type of spill backup volume that can be used for output.
SWAP NOSWAP	The system resource manager decides whether or not to make the DFSMSHsm address space able to be swapped.
SYSOUT(A 1)	DFSMSHsm prints to class A, makes one copy, and uses no special form.
SYS1DUMP	DFSMSHsm writes the output of a DFSMSHsm dump to a system dump data set.
TAPEDELETION(SCRATCHTAPE)	Tapes become scratch tapes after DFSMSHsm recycles them.
TAPEINPUTPROMPT(TAPES(YES))	DFSMSHsm issues action messages to the operator asking if needed input tapes are available.
TAPEMAXRECALLTASKS(1)	DFSMSHsm cannot process more than 1 tape recall task. The rest are DASD recall tasks. This parameter is a subset of the MAXRECALLTASKS parameter.
TAPEMIGRATION(NONE (ROUTETOTAPE(ANY)) RECONNECT(NONE))	DFSMSHsm selects the first available migration tape when a tape migration level 2 volume is required. If no tapes are available, DFSMSHsm selects a scratch tape with the unit name specified or defaulted from the MIGUNITNAME parameter of the SETSYS command. DFSMSHsm does not perform reconnection.

Table 63. Summary of DFSMSHsm Default Values (continued)	
DFSMSHsm default value	Result
TAPEOUTPUTPROMPT (TAPECOPY(NO))	If you specify N, DFSMSHsm issues no message to the operator during TAPECOPY processing indicating which type of target tape is to be mounted on a particular device.
TAPERECALLLIMITS(TASK(15) TAPE(20))	DFSMSHsm allows a task to continue processing from a single mount of a tape for at least 15 minutes and/or until a tape recall request of higher priority exists which can not be started until a task is freed up. Then the task is freed up for other recalls. DFSMSHsm allows a task to have a tape allocated and mounted for at least 20 minutes and/or until another host needs that mounted tape for a higher priority recall. Then the tape is freed up by this host which allows other hosts an opportunity to use that tape.
TAPESECURITY(PASSWORD)	DFSMSHsm uses password protection for its tape volumes.
TAPESPANSIZE(500)	DFSMSHsm can leave unused up to 500 megabytes (MB) of tape during attempts to eliminate spanning data sets.
TAPETYPE(NOWORM)	DFSMSHsm does not use WORM media for non-ABARS functions.
TAPEUTILIZATION PERCENTFULL(97)	DFSMSHsm forces an end-of-volume on a backup or migration tape volume when it is 97% full.
TAPEUTILIZATION CAPACITYMODE(COMPATIBILITY)	If you do not specify CAPACITYMODE(EXTENDED) for an esoteric capable of CAPACITYMODE switching, the drives operate in CAPACITYMODE(COMPATIBILITY) for output. The drives use CAPACITYMODE(COMPATIBILITY) or CAPACITYMODE(EXTENDED) as necessary for input.
UNITNAME(3590-1)	CDSVERSIONBACKUP: The default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist, or the parameter (or subparameter) has never been specified, DFSMSHsm uses a unit of 3590-1 to allocate scratch tapes during backup. UNITNAME: DFSMSHsm uses a unit of 3590-1 to allocate scratch tapes during backup.
USECYLINDERMANAGEDSPACE(N)	Cylinder-managed space is not used for migration copies and backup versions. For allocation, DFSMSHsm does not specify an EATTR value.
VERSIONS(2)	DFSMSHsm keeps two backup versions of each data set.
VOLCOUNT(NONE)	The default is NONE and DFSMSHsm does not pass the VOLCOUNT parameter to DFSMSdss.
VOLUMEDUMP(NOCC)	Concurrent copy will not be used by DFSMSdss when dumping volumes.

Table 63. Summary of DFSMSHsm Default Values (continued)

DFSMSHsm default value	Result
ZCOMPRESS(NONE)	The default is NONE and the SETSYS COMPACT settings will be used.

Optional parameters of the SETSYS command

References to "all" data sets in the descriptions of the SETSYS parameters mean *both* SMS-managed and non-SMS-managed data sets for those parameters that are applicable to SMS.

ABARSACTLOGMSGLVL: Specifying which DFSMSdss messages are written to the log

ABARSACTLOGMSGLVL is used to specify which DFSMSdss messages are written to the respective ABARS activity log.

FULL

Specifies that all DFSMSdss messages are intercepted and written to the ABARS activity log.

REDUCED

Specifies that only DFSMSdss warning or error messages are displayed in the ABARS activity log. No informational messages will be written.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

ABARSACTLOGTYPE: Specifying the ABARS activity log output data set type

ABARSACTLOGTYPE is used to specify the ABARS activity log output data set type.

Subparameter	Explanation
SYSOUT(class)	Messages are written out to a spool data set. For <i>class</i> , substitute an alphanumeric character for the class ABARS is to use for output.
DASD	<p>DFSMSHsm dynamically allocates DASD data sets with a unit name of SYSALLDA and a size of 20 tracks for primary allocation and 50 tracks for secondary allocation. Activity logs have names in the following form:</p> <pre>mcvtactn.Hmcvthost.function.agname.Dyyddd.Thhmmss</pre> <p>where:</p> <ul style="list-style-type: none"> • mcvtactn = Activity log high-level qualifier. The default is HSMACT. • Hmcvthost = DFSMSHsm host ID from the PROC statement, preceded by H. • function = Either ABACKUP or ARECOVER. • agname = The name of the aggregate group being backed up. • Dyyddd = Year and day of allocation, preceded by D. • Thhmmss = Hour, minute, and second of allocation, preceded by T.

Note: SYSOUT has the advantage of not utilizing DASD space, whereas DASD has the advantage of being machine readable and readily available. Outputting to DASD will allow the ABACKUP activity log to be backed up as part of the aggregate backup output.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: SYSOUT is the initial default setting. If class is not specified, the default will be the SETSYS ACTLOGTYPE SYSOUT class. If ACTLOGTYPE has been set to DASD, the ABARSACTLOGTYPE class defaults to the SETSYS SYSOUT class, which in turn defaults to “A”.

If SETSYS ABARSACTLOGTYPE has not been issued, each time SETSYS ACTLOGTYPE is issued, it makes the corresponding change to the setting of ABARSACTLOGTYPE.

Note: Once SETSYS ABARSACTLOGTYPE is issued, any subsequent SETSYS ACTLOGTYPE command does not affect the ABARSACTLOGTYPE.

ABARSBUFFERS: Specifying the number of I/O buffers for ABARS

ABARSBUFFERS(*n*) is used to specify the number of I/O buffers to be used by ABARS when DFSMSHsm is used to read or write data sets during aggregate backup and recovery processing. This parameter does not affect the number of buffers used when DFSMSdss is invoked to DUMP or RESTORE data sets.

n specifies a number in the range of 1 to 9 indicating the number of buffers used for I/O processing. This number is internally multiplied by two (a buffer for input and a buffer for output).

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If the number of buffers is not specified by the SETSYS ABARSBUFFERS(*n*) command, the default is one.

ABARSDELETEACTIVITY: Allowing DFSMSHsm to automatically delete the ABARS activity log

ABARSDELETEACTIVITY(Y | N) is used to select whether or not DFSMSHsm deletes the ABARS activity log during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing.

Y specifies that DFSMSHsm is to delete the ABARS activity log during ABARS processing. N specifies that no automatic deletion is performed by DFSMSHsm.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: The ABARS activity log is not automatically deleted during ABARS processing.

ABARSKIP: Specifying which types of data sets ABACKUP should skip

ABARSKIP is used to specify which types of data sets that ABACKUP processing should skip. If a data set in an aggregate group is on a volume or volumes that are protected by peer-to-peer remote copy, specifying ABARSKIP(PPRC) causes ABACKUP processing to skip the data set and not back it up. If a data set is on a volume or volumes protected by extended remote copy, specifying ABARSKIP(XRC) causes ABACKUP processing to skip the data set and not back it up.

Specifying ABARSKIP(NOPPRC) or ABARSKIP(NOXRC) causes ABACKUP processing to not skip the corresponding data set and it is backed up.

SMS relationship: Parameter has the same meaning when it is applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: The DFSMSHsm default is NOPPRC and NOXRC.

ABARSOPTIMIZE: Adjusting performance when backing up level 0 DASD specified in INCLUDE list

ABARSOPTIMIZE(*option*) is used to control the value that the ABACKUP function specifies in the OPTIMIZE parameter when invoking DFSMSdss to backup level 0 DASD data sets.

The valid specifications for the *option* value are:

- If value is 1, then DFSMSdss reads one track at a time
- If value is 2, then DFSMSdss reads two tracks at a time
- If value is 3, then DFSMSdss reads five tracks at a time
- If value is 4, then DFSMSdss reads one cylinder at a time

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: The default value for *option* is 3.

ABARSPROCNAME: Specifying the base procedure name to start the ABARS secondary address space

ABARSPROCNAME(*name*) is used to specify the JCL procedure name used on the START command issued by DFSMSHsm to start the ABARS secondary address space. For *name*, specify a procedure name that conforms to standard MVS JCL naming conventions.

DFSMSHsm appends *ABARSnnst* as an identifier to this procedure name when starting ABARS secondary address spaces. The value *nn* changes each time a new address space is started. The value will range from 1 to 64, so that each address space is started with a unique procedure name. The value *st* is derived from the current clock time when the procedure is started.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify a procedure name on any SETSYS command, the DFSMSHsm default is DFHSMABR.

ABARSTAPES: Allowing DFSMSHsm to stack the ABACKUP output files

ABARSTAPES(STACK | NOSTACK) is used to indicate to DFSMSHsm to stack the ABACKUP output files onto a minimum number of tape volumes during ABACKUP processing.

Subparameter	Explanation
STACK	DFSMSHsm stacks the ABACKUP output files onto a minimum number of tape volumes. STACK also indicates to the ARECOVER function that the ABACKUP output files are stacked and if ARECOVER determines that the ABACKUP output files are not stacked, ARECOVER attempts to recover the output files as if NOSTACK was specified.
NOSTACK	DFSMSHsm does not stack the ABACKUP output files. The ABACKUP function operates as it did in prior releases. NOSTACK also indicates to the ARECOVER function that the ABACKUP output files are not stacked and if ARECOVER determines that the ABACKUP output files are stacked, ARECOVER attempts to recover the output files as if STACK was specified.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: The default is STACK.

ABARSUNITNAME: Specifying a unit name for ABARS

ABARSUNITNAME(*unittype*) is used to specify the unit to be used when invoking aggregate backup or aggregate recovery.

Tape Library Relationship: A UNITNAME can be used for filtering in the ACS routines. If you specify the *unittype* for this command, it will be sent to the ACS routines in SMS for the ABARS function.

unittype specifies the tape unit to be allocated for the control file, instruction/activity log, and data file. The valid types of tape units are:

- 3400-3, 3400-4, 3400-5, 3400-6, 3400-9
- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the USERUNITTABLE parameter of the SETSYS command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. Specify 3400-9 when your 3480 Magnetic Tape Subsystem simulates your 3420 tape drives.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If *unitname* is not specified by the SETSYS ABARSUNITNAME(*unitname*) or the UNIT parameter on the ABACKUP or ARECOVER command, it defaults to 3590-1.

ABARSVOLCOUNT: Specifying the method for determining volume count for SMS target data set allocations done by DFSMSdss

ABARSVOLCOUNT(ANY | NONE) is an optional parameter that affects how DFSMShsm will invoke DFSMSdss during the ARECOVER of data sets that were dumped by DFSMSdss. This parameter affects only the ARECOVER of data sets dumped from primary volumes (LO data sets). Refer to *z/OS DFSMSdss Storage Administration* for more specific information about the restrictions and limitations of the DFSMSdss VOLCOUNT parameter.

Subparameter	Explanation
ANY	DFSMShsm passes the VOLCOUNT(ANY) parameter to DFSMSdss during the ARECOVER of data dumped by DFSMSdss (LO data sets). This results in the allocation of the target data set on as many volumes as required, to a maximum of 59.
NONE	The VOLCOUNT parameter is not passed to DFSMSdss and DFSMShsm operates as it did prior to the VOLCOUNT support being added where the source volume count must be maintained (for example, no single volume data sets can be extended to multiple volumes). If, for example, ABARSVOLCOUNT(ANY) had been previously specified, issuing SETSYS ABARSVOLCOUNT(NONE) resets the associated DFSMShsm indicators.

The ABARSVOLCOUNT parameters of ANY or NONE are ignored if specified for data types to which they do not apply. Refer to the *z/OS DFSMShsm Implementation and Customization Guide* to determine data set types moved by DFSMSdss for DFSMShsm.

SMS relationship: Parameter applies in some instances to SMS-managed DASD volumes or data sets.

SETSYS default: There is no SETSYS default.

DFSMSHsm default: If you do not specify SETSYS ABARSVOLCOUNT, the default is NONE.

ACCEPTPSCBUSERID and NOACCEPTPSCBUSERID: Allowing non-RACF installations to issue authorized commands in a TMP environment

Explanation: **ACCEPTPSCBUSERID** | **NOACCEPTPSCBUSERID** are mutually exclusive, optional parameters you use to specify whether DFSMSHsm should have a way of issuing authorized DFSMSHsm commands in a TSO batch environment in installations not having RACF. When RACF is not installed, authorized DFSMSHsm commands issued in a pure batch environment fail because of a lack of authorization.

ACCEPTPSCBUSERID specifies that an installation not having RACF will have DFSMSHsm retrieve the user ID for TSO batch requests from the protected step control block (PSCB) and associate the user ID with the request. Thus, the current authorization checking for commands can be used to determine if the user ID is an authorized DFSMSHsm user ID. If no user ID is present in the PSCB, ****BATCH*** will be used. An installation must ensure that the user ID is placed in the PSCB if **ACCEPTPSCBUSERID** is specified. If an installation does not issue the SETSYS **ACCEPTPSCBUSERID** command and RACF is not installed, DFSMSHsm will not attempt to retrieve the user ID for TSO batch requests and all DFSMSHsm authorized-user commands will fail.

NOACCEPTPSCBUSERID specifies that DFSMSHsm will not attempt to retrieve the user ID for TSO batch requests when RACF is not installed.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is **NOACCEPTPSCBUSERID**.

Notes:

1. The SETSYS **ACCEPTPSCBUSERID** command applies only to TSO batch requests on systems where RACF is not installed.
2. When SETSYS **SYS1DUMP** is entered in the DFSMSHsm startup command member in a PARMLIB, there is a possibility of an ABEND occurring during DFSMSHsm initialization prior to this command being processed. If an ABEND occurs in this window and a dump listing is desired, include a **SYSUDUMP DD** card in the DFSMSHsm startup procedure.

ACTLOGMSGLVL: Specifying message level for recording data sets

Explanation: **ACTLOGMSGLVL(FULL | EXCEPTIONONLY | REDUCED)** is an optional parameter specifying the message log level for the activity log required for recording data sets processed by volume and level functions. Specifically, this parameter determines which ARC0734I data set movement messages will be written to the activity log. Only the issuance of the ARC0734I message is affected.

Subparameter	Explanation
FULL	Messages are generated and logged for all activities.
EXCEPTIONONLY	Messages are generated and logged for activities with a nonzero return code.
REDUCED	The original space management message or backup message is generated, but messages for subsequent movement and cleanup are suppressed.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify any of the subparameters with the ACTLOGMSGLVL parameter, the DFSMSHsm default is FULL.

Notes:

1. ACTLOGMSGLVL is not supported for secondary address spaces.
2. Any messages excluded from the activity logs by the use of ACTLOGMSGLVL will also be excluded from the HSMLOGX/Y.

ACTLOGTYPE: Specifying the type of output data set to receive messages

Explanation: **ACTLOGTYPE(SYSOUT(class) | DASD)** is an optional parameter specifying the activity logs as either a SYSOUT data set or a DASD data set. The content of the log is free-form and can be scanned to produce a desired output.

Subparameter	Explanation
SYSOUT(class)	Messages are written out to a spool data set. For <i>class</i> , substitute an alphanumeric character for the class DFSMSHsm is to use for output.
DASD	<p>DFSMSHsm dynamically allocates DASD data sets with a unit name of SYSALLDA and a size of 20 tracks for primary allocation and 50 tracks for secondary allocation. Activity logs have names in the following form:</p> <pre>mcvtactn.Hmcvthost.function.Dyyddd.Thhmmss or mcvtactn.HUmcvthost.function.Dyyddd.Thhmmss</pre> <p>where:</p> <ul style="list-style-type: none"> • <i>mcvtactn</i> = Activity log high-level qualifier, the default is HSMACT • <i>Hmcvthost</i> = DFSMSHsm host ID from the PROC statement, preceded by H • <i>HUmcvthost</i> = DFSMSHsm Filemode host ID from the PROC statement, preceded by HU • <i>function</i> = Either CMDLOG, BAKLOG, DMPLOG, or MIGLOG • <i>Dyyddd</i> = Year and day of allocation, preceded by D • <i>Thhmmss</i> = Hour, minute, and second of allocation, preceded by T

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: The SYSOUT class established by the SETSYS SYSOUT command.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is SYSOUT. If you do not specify *class* for SYSOUT, the DFSMSHsm default is A, one copy, and no special forms.

Notes:

1. If the SYSOUT class is modified with the SETSYS ACTLOGTYPE command, existing open SYSOUT activity logs are closed and new SYSOUT activity logs are opened using the output class specified with the command. For the DFSMSHsm secondary address space, the new output class is used for subsequent secondary address spaces as they are started.
2. ACTLOGTYPE is supported for secondary address spaces.
3. If you need to replace HSMACT as the activity log high-level qualifier, refer to [Tuning DFSMSHsm in z/OS DFSMSHsm Implementation and Customization Guide](#).

ARECOVERML2UNIT: Specifying the unit when recovering migrated data sets to nonspecific volumes

Explanation: **ARECOVERML2UNIT(*unittype*)** specifies the type of tape unit to use in a nonspecific volume request when recovering migrated data sets to ML2 tape volumes. For *unittype*, substitute a valid unit type or a valid esoteric unit name. The following are valid types of tape units: 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, 3480, 3480X, 3490, and 3590-1. You specify 3400-9 when your 3480 Magnetic Tape Subsystem simulates your 3420 tape drives. You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails.

When in a mixed tape drive environment you must use a customer defined esoteric *unittype* since the generic will contain a mix of machine types. For example, if your tape environment has both 3590 Model H drives and 3592 Model J drives defined as 3590, then the "3590-1" generic will contain both of these machine types. If you intend to recover ML2 tapes on the 3592 Model J drives, create an esoteric exclusively for them, define the esoteric to DFSMSHsm via the SETSYS USERUNITTABLE command, and define that esoteric as the ARECOVERML2UNIT.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

DFSMSHsm default: If you do not specify this parameter, the DFSMSHsm default is 3590-1.

ARECOVERPERCENTUTILIZED: Specifying the percentage of DASD space for non-SMS-managed volumes

Explanation: **ARECOVERPERCENTUTILIZED(*nnn*)** is an optional parameter that allows you to change the percentage to which DFSMSHsm allows DFSMSdss to fill non-SMS-managed DASD recovery volumes. For *nnn*, substitute an integer number between 1 and 100.

SMS relationship: Parameter does not apply to SMS-managed DASD volumes or data sets.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 80% (080).

ARECOVERTGTGDS: Passing TGTGDS parameter to DFSMSdss during restore processing

Explanation: **ARECOVERTGTGDS(*option*)** is used to pass to DFSMSdss the TGTGDS parameter. This parameter provides greater flexibility managing SMS-managed generation data sets that are being restored to level 0 DASD.

The valid values for *option* are as follows:

- DEFERRED specifies that the target data set is to be assigned the DEFERRED status.
- ACTIVE specifies that the target data set is to be assigned the ACTIVE status, for example, rolled into the GDG base.
- ROLLEDOFF specifies that the target data set is to be assigned the ROLLEDOFF status.
- SOURCE specifies that the target data set is to be assigned the same status as that of the source data set.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: The default is SOURCE.

ARECOVERUNITNAME: Specifying tape unit type for aggregate recovery

Explanation: **ARECOVERUNITNAME(*unittype*)** is an optional parameter specifying the tape unit type for data sets being recovered to user tape volumes during an aggregate recovery.

For *unittype*, substitute the type of tape unit. The following are valid types of tape units:

- 3400-3, 3400-4, 3400-5, 3400-6, 3400-9
- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. If you specify:

- 3400-9, your 3480 Magnetic Tape Subsystem simulates your 3420 tape drives.
- 3480 or 3480X, you use all the functions of the 3480 Magnetic Tape Subsystem.
- 3490, you use all the functions of the 3490 Magnetic Tape Subsystem.
- 3590-1, you use all the functions of the 3590 Magnetic Tape Subsystem.

You specify 3480 or 3480X when you use all the functions of the 3480 or 3480X Magnetic Tape Subsystem.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify *unittype* the default is 3590-1.

AUTOBACKUPSTART: Specifying when automatic backup is to start

Explanation: **AUTOBACKUPSTART((*hhmm1 hhmm2 hhmm3*)** is an optional parameter specifying the time when automatic backup can start. The first time you specify (*hhmm1*) is the time you plan for DFSMSHsm to start automatic backup. The second time you specify (*hhmm2*) is the latest time DFSMSHsm can start automatic backup. The third time you specify (*hhmm3*) is the time after which DFSMSHsm will not start automatic backup on additional volumes.

If DFSMSHsm does not start automatic backup between the planned start time and the latest start time, no automatic backup is run that day.

For *hhmm1*, substitute the planned time for DFSMSHsm to start automatic backup. If you specify *hhmm1* as zero, DFSMSHsm cannot start automatic backup.

Note: Time is based on the 24-hour clock and is expressed as a four-digit decimal number in the format *hhmm*. The hour is represented by *hh* and the minutes by *mm*. For example, 1:15 p.m. is 1315 and midnight is 2400. The maximum value you can specify for *hh* is 24, the maximum value you can specify for *mm* is 59, and the maximum value you can specify for *hhmm* is 2400.

For *hhmm2*, substitute the latest time DFSMSHsm can start automatic backup. If *hhmm2* is set to zero, this is the same as not including the *hhmm2* parameter and DFSMSHsm runs automatic backup processing until all volumes have been backed up. If you want automatic backup to begin after midnight, the value for the latest start time can be smaller than the planned start time. For example, you can specify 2330 for *hhmm1* and 0100 for *hhmm2*.

For *hhmm3*, substitute the time after which DFSMSHsm can no longer start automatic backup of additional volumes. If *hhmm3* is set to zero, this is the same as not including the *hhmm3* parameter. If you do not specify *hhmm3*, DFSMSHsm runs automatic backup processing until all volumes have been backed up.

If you want automatic backup to begin after midnight, *hhmm2*, the latest start time, can be smaller than *hhmm1*, the planned start time. As an example, you can specify 2330 for *hhmm1*, 0100 for *hhmm2*, and 0330 for *hhmm3*. DFSMSHsm assumes that *hhmm2* is a later time than *hhmm1* and *hhmm3* is a later time than *hhmm2*.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you do not specify a value for *hhmm2*, the SETSYS default is 2400.

DFSMShsm default: If you do not specify this parameter on any SETSYS command, the DFSMShsm default is no automatic backup.

Note:

1. If you do not specify a nonzero value for *hhmm1* during this start up of DFSMShsm, DFSMShsm does not do automatic backup.
2. The maximum value you can specify for *mm* is 59, and the maximum value you can specify for *hhmm1*, *hhmm2*, or *hhmm3* is 2400.
3. You use the BACKUP parameter of the DEFINE command to specify on which days DFSMShsm is to run automatic backup. The AUTOBACKUPSTART parameter specifies the time when automatic backup can start. If automatic backup does not start before the latest start time, DFSMShsm does not start automatic backup until the next day in the cycle that automatic backup is scheduled to run. If automatic backup did not start today, you can use AUTOBACKUPSTART to change the planned and latest start times.
4. DFSMShsm serializes access to your data sets and volumes for a short period of time while it runs automatic backup. Therefore, choose a time for automatic backup when the computing system is not very active and there is not much interactive user activity.
5. If you want the operator to decide whether DFSMShsm should start automatic backup, specify the REQUEST parameter and the AUTOBACKUPSTART parameter.
6. If you specify *hhmm3* as zero, automatic backup will process all eligible primary volumes.
7. The SETSYS command fails if *hhmm1* and *hhmm2* are equal or if *hhmm2* and *hhmm3* are equal.
8. DFSMShsm does not honor the value for *hhmm3* until the volume phase of the AUTOBACKUP function.

AUTODUMPSTART: Specifying when automatic dump is to start

Explanation: **AUTODUMPSTART(*hhmm1 hhmm2 hhmm3*)** is an optional parameter specifying the time when automatic dump can start. The first time you specify (*hhmm1*) is the time you plan for DFSMShsm to start automatic dump. The second time you specify (*hhmm2*) is the latest time DFSMShsm can start automatic dump. The third time you specify (*hhmm3*) is the time after which DFSMShsm will not start automatic dump on additional volumes.

If DFSMShsm does not start automatic dump on volumes by the *hhmm2* time, no automatic dumping is performed that day. Volumes being processed when the *hhmm3* time is reached are not affected.

For *hhmm1*, substitute the planned time for DFSMShsm to start automatic dump. If you specify *hhmm1* as zero, DFSMShsm does not start automatic dump.

Note: Time is based on the 24-hour clock and is expressed as a four-digit decimal number in the format *hhmm*. The hour is represented by *hh* and the minutes by *mm*. For example, 1:15 p.m. is 1315 and midnight is 2400. The maximum value you can specify for *hh* is 24, the maximum value you can specify for *mm* is 59, and the maximum value you can specify for *hhmm* is 2400.

For *hhmm2*, substitute the latest time DFSMShsm can start automatic dump. If *hhmm2* is set to zero, this is the same as not including the *hhmm2* parameter and DFSMShsm runs automatic dump processing until all volumes have been dumped. If you want automatic dump to begin after midnight, the value for the latest start time can be smaller than the planned start time. For example, you can specify 2330 for *hhmm1* and 0100 for *hhmm2*.

For *hhmm3*, substitute the time after which DFSMShsm can no longer start the automatic dumping of additional volumes. If *hhmm3* is set to zero, this is the same as not including the *hhmm3* parameter. If you do not specify *hhmm3*, DFSMShsm runs automatic dump processing until all volumes have been dumped.

If you want automatic dumping to begin after midnight, *hhmm2*, the latest start time, can be smaller than *hhmm1*, the planned start time. As an example, you can specify 2330 for *hhmm1*, 0100 for *hhmm2*, and

0330 for *hhmm3*. DFSMSShsm assumes that *hhmm2* is a later time than *hhmm1* and *hhmm3* is a later time than *hhmm2*.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you do not specify a value for *hhmm2*, the SETSYS default is 2400.

DFSMSShsm default: If you do not specify this parameter on any SETSYS command, the default is no automatic dump.

Note:

1. If you do not specify a nonzero value for *hhmm1* during this start up of DFSMSShsm, DFSMSShsm does not do automatic dumping.
2. The maximum value you can specify for *mm* is 59, and the maximum value you can specify for *hhmm1*, *hhmm2*, or *hhmm3* is 2400.
3. You use the DUMPCYCLE parameter of the DEFINE command to specify on which days DFSMSShsm is to run automatic dump. The AUTODUMPSTART parameter specifies the time when automatic dump can start. If automatic dump does not start before the latest start time, DFSMSShsm does not start automatic dump until the next day in the cycle that automatic dump is scheduled to run. If automatic dump did not start today, you can use AUTODUMPSTART to change the planned and latest start times.
4. AUTODUMPSTART has no meaning unless the BACKUP parameter is specified on a SETSYS command during this DFSMSShsm startup.
5. An explicit command must be issued to halt the automatic dump process before normal completion.
6. If you want the operator to decide whether DFSMSShsm should start automatic dump, specify the REQUEST parameter and the AUTODUMPSTART parameter.
7. As with automatic backup, DFSMSShsm serializes access to data sets and volumes for a short period of time. Therefore, select a time for automatic dump when the computing system is not very active and there is not much interactive user activity (usually at night).
8. Automatic dump usually starts at the planned start time. Assume, however, that DFSMSShsm is not running at the planned start time or the operating system is lost while automatic dump is running. When DFSMSShsm restarts, it checks whether the latest start time has passed. If it has not passed, DFSMSShsm restarts automatic dump. If the time has passed and you want automatic dump to run today, you can use this parameter to respecify the time when automatic dump can start.
9. If you specify *hhmm3* as zero, automatic dump processes all eligible primary volumes.
10. The SETSYS command will fail if *hhmm1* and *hhmm2* are equal.
11. DFSMSShsm can run automatic dump multiple times a day in a test environment. If automatic dump runs to completion and you want to start automatic dump again, use the following procedure:
 - a. Issue the SETSYS AUTODUMPSTART command to define a new start window.
 - b. Specify a start window that has a planned start time after the time that automatic dump last ended.
 - c. Do not confuse the time that automatic dump last ended with the specified latest start time.

Automatic dump starts immediately if the current time is in the new start window. If the current time is not in the new start window, automatic dump starts automatically when the planned start time occurs.

AUTOMIGRATIONSTART: Specifying the start time for automatic primary space management

The use of the SETSYS AUTOMIGRATIONSTART parameter is nonfunctional and its use has been replaced by other functions. The AUTOMIGRATIONSTART parameter is still supported but is no longer documented.

BACKUP and NOBACKUP: Specifying whether you intend to use DFSMSHsm backup, recovery, and dump

Explanation: **BACKUP** | **NOBACKUP** are mutually exclusive, optional parameters specifying whether you intend to use any of the following DFSMSHsm functions: backup, dump, recovery, and expiring backup versions. The **BACKUP** parameter allows you to specify restrictions as to what kind of volumes are allowed to be used for daily backup.

The target device type used for volume backups can come from multiple sources. There are the **BACKUPDEVICECATEGORY** options on the **BACKVOL** and **ADDVOL** commands and there are the **SETSYS BACKUP** options of **ANY** | **DASD** | **TAPE**. The target device type can apply to the backup from SMS DASD when specified on a **BACKVOL** command or with the **SETSYS BACKUP** command. For non-SMS DASD, the target device type can be specified by any **BACKVOL**, **ADDVOL**, or **SETSYS BACKUP** command.

Tape Library Relationship: A **UNITNAME** can be used for filtering in the ACS routines. If you specify the **unittype** for this command, it will be sent to the ACS routines in SMS for the **BACKUP** function.

BACKUP specifies that you intend to use some or all of the following DFSMSHsm functions: backup (automatic or command), dump (automatic or command), recovery, and expire backup versions.

To do the following functions	Use this DFSMSHsm command
Backup	HBACKDS, BACKDS, or BACKVOL
Automatic backup	SETSYS AUTOBACKUPSTART and DEFINE BACKUP
Dump	BACKVOL and DEFINE DUMPCLASS
Automatic dump	SETSYS AUTODUMPSTART, DEFINE DUMPCLASS, and DEFINE DUMPCYCLE
Recovery	HRECOVER or RECOVER
Expire backup versions	EXPIREBV

The **SETSYS BACKUP** options **ANY** | **DASD** | **TAPE**(*unittype*) are there to control the target of the backup output. The **SETSYS BACKUP** options apply only if the **BACKUPDEVICECATEGORY** option is not specified on the **BACKVOL** command (SMS or non-SMS) or on an **ADDVOL** command (non-SMS only).

Subparameter	Explanation
ANY	During volume backup processing, unless overridden by a BACKVOL command or by an ADDVOL of a non-SMS volume command, any type of daily backup volume (DASD or Tape) defined to DFSMSHsm by the ADDVOL can be used for output.
DASD	During volume backup processing, unless overridden by a BACKVOL command or by an ADDVOL of a non-SMS volume command, only DASD daily backup volumes defined by the DFSMSHsm ADDVOL command can be used for output.
TAPE	During volume backup processing, unless overridden by a BACKVOL command or by an ADDVOL of a non-SMS volume command, only tape daily backup volumes defined to DFSMSHsm via the ADDVOL command can be used for output.

Subparameter	Explanation
TAPE(<i>unittype</i>)	<p>For volume backup processing to non-SMS tape, only tape daily backup volumes that can be mounted and written on the specified type <i>unittype</i> can be used for output. For SMS tape, the specified type <i>unittype</i> will be passed to the ACS routines. A corresponding <i>unittype</i> restriction for dump output can be specified with the DEFINE DUMPCLASS command.</p> <p>For <i>unittype</i>, a generic or esoteric unit name can be specified. The following generic unit names are acceptable: 3480, 3480X, 3490 and 3590-1. If an esoteric unit name is specified, it must have been previously defined to DFSMSHsm using the USERUNITTABLE parameter of the SETSYS command. You specify 3480 or 3480X when you use all the functions of the 3480 or 3480X Magnetic Tape Subsystem.</p>

Related reading: For more information, see [“Selecting daily backup volumes”](#) on page 207.

INUSE, with its subparameters, specifies how DFSMSHsm should handle backing up a data set that is found to be in use (open for update) during volume backup processing. The following subparameters can be used with the INUSE parameter:

Subparameter	Explanation
RETRY(Y)	DFSMSHsm will retry once to back up a data set if the first attempt fails because (1) the data set is in use, or (2) a BWO candidate suffers a CA/CI split and the backup is discarded.
RETRY(N)	DFSMSHsm does not retry a backup if the first attempt fails because the data set is in use.
DELAY(<i>min</i>)	DFSMSHsm delays for <i>min</i> minutes before retrying a backup attempt which failed because a data set was in use. For <i>min</i> , specify a value from 0 to 999.
SERIALIZATION (PREFERRED)	If DFSMSHsm retries backing up a data set (after some delay) and finds the data set still (or again) in use, it should back up the data set anyway.
SERIALIZATION (REQUIRED)	If DFSMSHsm retries backing up a data set (after some delay) and finds the data set still (or again) in use, it should fail the backup.

NOBACKUP specifies that you do not intend to use DFSMSHsm backup, dump, recovery, or expire backup versions. When you specify this parameter, automatic backup and automatic dump are not run, and DFSMSHsm fails any backup, dump, recovery, or expire backup versions commands.

If you specify the NOBACKUP parameter while DFSMSHsm functions are running, the following will occur:

If . . .	Then . . .
Volume backup is running	The volume backup operation ends after the data set currently being processed has been backed up.
Volume dump is running	The volume dump operation ends after the volume currently being processed has been dumped.
Automatic backup is running	Automatic backup prematurely ends after all volume backup operations in process have ended.
Automatic dump is running	Automatic dump prematurely ends after all volume dump operations in process have ended.

If . . .	Then . . .
EXPIREBV is running	EXPIREBV prematurely ends after all expire backup versions operations in process have ended.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: The following table describes the SETSYS BACKUP and SETSYS NOBACKUP default.

If . . .	Then . . .
You specify BACKUP without any subparameters	Any previous backup output specification and specifications for handling data sets in use are retained.
You specify RETRY without a value	The SETSYS default is N.
You specify DELAY without a value	The SETSYS default is 15.
You specify SERIALIZATION without a value	The SETSYS default is REQUIRED.

DFSMSHsm default: The following table describes the SETSYS BACKUP and SETSYS NOBACKUP DFSMSHsm default.

If . . .	Then . . .
You do not specify either BACKUP or NOBACKUP on any SETSYS command	The DFSMSHsm default is NOBACKUP. If BACKUP has been specified but never (within this start-up of DFSMSHsm) with any of the ANY DASD TAPE options specified, then the default option of ANY is used. If an option of BACKUP (either ANY DASD TAPE) has been specified already but <i>not</i> on this issuance of the BACKUP command, then the command is treated as if the option ANY was specified. This means that the ANY option is the default option for BACKUP.
You do not specify INUSE on any SETSYS BACKUP command	DFSMSHsm does not retry any data set backup if the first attempt finds that the data set is currently in use (that is, the backup fails).
You do not specify the RETRY subparameter on any SETSYS BACKUP command	DFSMSHsm does not retry any data set backup if the first attempt finds that the data set is currently in use (that is, the backup fails).
You do not specify the DELAY subparameter but you specify RETRY(Y)	DFSMSHsm delays for 15 minutes before doing a retry.
You do not specify the SERIALIZATION subparameter on any SETSYS command	If DFSMSHsm attempts a retry of a backup and finds the data set still in use, it will fail the backup.

Note:

1. If DFSMSHsm has scheduled a retry of a backup, and you then change the value of the DELAY subparameter, DFSMSHsm will use the most recent DELAY value to determine the time for attempting the retry.

2. Installation exit ARCBDEXT, if enabled, can override the RETRY and SERIALIZATION options for a given data set.
3. BWO data sets with CA/CI split will not be retried if SERIALIZATION(PREFERRED) has been specified.
4. When a backup retry is internally scheduled for a data set, it is the equivalent of a BACKDS command being issued. The settings used on the SETSYS DSBACKUP command affect where the backup is directed. Requests will be directed to tape or DASD based on the settings used in the DASDSELECTIONSIZE parameter.
5. If you specify BACKUP(ANY) on the SETSYS command, DFSMSHsm selects the first nonfull, unallocated, daily backup volume assigned to the current day in the backup cycle. If you want DFSMSHsm to use tape as a daily backup volume and you do not specify restrictions limiting the selection to tape, you must define at least one tape for each of the appropriate days in the backup cycle by using the ADDVOL command. DFSMSHsm does not select a scratch tape if it cannot find any daily or unassigned, nonfull backup volumes that are already defined to DFSMSHsm. If DASD daily backup volumes are not being used in your environment, specify SETSYS BACKUP(TAPE) rather than SETSYS BACKUP(ANY). This allows the flexibility of backing up to a scratch tape if an eligible daily backup volume is not found. You do not have to define tapes as daily backup volumes with the ADDVOL command if you restrict daily backup volumes to tape by:
 - Specifying BACKUP(TAPE) on a SETSYS command.
 - Specifying BACKUPDEVICECATEGORY(TAPE) on a BACKVOL command.
 - Specifying BACKUPDEVICECATEGORY(TAPE) on the ADDVOL command of the nonSMS volume being backed up.

When daily backup volumes are restricted to the tape in one of these ways, DFSMSHsm selects a scratch tape if an eligible daily backup volume is not found.

BACKUPPREFIX: Specifying the prefix for the backup version name or the dump copy data set name

Explanation: **BACKUPPREFIX(prefix)** is an optional parameter specifying the prefix (high-level qualifier) of the data set name DFSMSHsm generates when it backs up a data set or dumps a volume. *prefix* must be defined as 1 to 7 alphanumeric characters; the first character must be alphabetic only.

Attention

Changing this parameter after the DFSMSHsm environment is set could result in failures during the recall and recovery of data sets from tape and SDSs.

The **backup version** name has the following format:

```
prefix.BACK.Tssmmhh.user1.user2.Xydd
```

If patch-enabled, the following dataset name will be generated:

```
prefix.BACK.Tcccchh.user1.user2.Xydd
```

where:

- *prefix* is replaced with the prefix you specify with this command.
- BACK indicates that this is a backup version.
- Tssmmhh is the time when DFSMSHsm backed up the data set; ss is the second, mm is the minute, and hh is the hour. If a duplicate exists after this name is generated, DFSMSHsm changes the first character of the time stamp.
- Tcccchh is the time when DFSMSHsm backed up the data set; cccc is the time in hundredths of seconds from the beginning of the hour converted to four alphabetic characters and hh is hours. For more information, see [“Generating names of backup versions of data sets” on page 204](#).

- *user1* and *user2* are replaced with the first two qualifiers of the data set name. The data set name can be 44 characters long.
- *Xydd* is the date when DFSMSHsm backed up the data set. DFSMSHsm replaces the X with a letter that represents the decade. A–J have the following numeric values:

A–1,	B–2,	C–3,	D–4,	E–5
F–6,	G–7,	H–8,	I–9,	J–0

For example, you specify BACKUPPREFIX(HSMBAK). Your data set name is TERRY.CLIST.TEXT. DFSMSHsm backs up the data set on 30 December 1991 at 9:15 a.m. The backup version name is:

```
HSMBAK.BACK.T001509.TERRY.CLIST.I0364
```

The **dump copy** data set name has the following format:

```
prefix.DMP.dclass.Vvolser.Dyyddd.Tssmmhh
```

where:

- *prefix* is replaced with the prefix you specify with this command
- DMP indicates that this is a dump version
- *dclass* is the dump class name
- *Vvolser* is the source volume serial number
- *Dyyddd* is the date when DFSMSHsm dumped the data set. *yy* is the last two digits of the year of the dump date stamp. *ddd* is the Julian day of the dump date stamp.
- *Tssmmhh* is the time when DFSMSHsm dumped the data set; *ss* is the second, *mm* is the minute, and *hh* is the hour.

If you request it, DFSMSHsm makes a dump VTOC copy data set for each dump generation that is made for a volume.

The **dump VTOC copy** data set name has the following format:

```
prefix.DUMPVTOC.Tssmmhh.Vvolser.Dyyddd
```

where:

- *prefix* is replaced with the prefix you specify with this command
- DUMPVTOC indicates that this is a dump VTOC version
- *Tssmmhh* is the time when DFSMSHsm dumped the data set; *ss* is the second, *mm* is the minute, and *hh* is the hour.
- *Vvolser* is the source volume serial number
- *Dyyddd* is the date when DFSMSHsm dumped the data set. *yy* is the last two digits of the year of the dump date stamp. *ddd* is the Julian day of the dump date stamp.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is the UID parameter you specified in the DFSMSHsm startup procedure. This UID is used as the first qualifier for data set names generated by DFSMSHsm during the BACKUP or MIGRATION functions. Changing this UID after the DFSMSHsm environment is set could result in failures during the recall and recovery of data sets from tape and SDSPs. [z/OS DFSMSHsm Implementation and Customization Guide](#) explains the UID and the DFSMSHsm startup procedure.

CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets

Explanation: **CDSVERSIONBACKUP** is an optional parameter that you use to specify whether or not DFSMSHsm should back up the control data sets to multiple data sets. The multiple backup copies give you a history of the control data sets and the journal data set. The number of backup copies kept is determined by the BACKUPCOPIES subparameter.

In a multiple-DFSMSHsm-host environment, all hosts sharing DFSMSHsm control data sets must specify the same options for the SETSYS CDSVERSIONBACKUP command, as the information is stored in the MCDS. The last DFSMSHsm host to issue the SETSYS CDSVERSIONBACKUP command controls how all the hosts back up the CDSs. An alternative to having all DFSMSHsm hosts specify the same options for the command is to have only one host (usually the primary host) specify the command. If you use the same ARCCMDxx PARMLIB member for all hosts, you can use the ONLYIF command in that member to restrict the SETSYS parameter to one host.

If you specify CDSVERSIONBACKUP while DFSMSHsm is backing up the control data sets and journal data set, the parameter does not take effect until after DFSMSHsm has finished backing up the control data sets and journal data set.

The subparameters of the CDSVERSIONBACKUP parameter follow this discussion in alphabetical order. The following are optional subparameters:

- BACKUPCOPIES
- BACKUPDEVICECATEGORY
- DATAMOVER
- BCDSBACKUPDSN
- MCDSBACKUPDSN
- OCDSBACKUPDSN
- JRNLBACKUPDSN

Note: The use of the NOCDSVERSIONBACKUP parameter has been disabled.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If any of these parameters or subparameters are not specified, the default value is read from the multiple-host processor control record (MHCR). The record is updated with values specified on the SETSYS CDSVERSIONBACKUP command every time the command is issued. If the MHCR record does not exist or the parameter (or subparameter) has never been specified, the DFSMSHsm defaults are as follows:

- BACKUPCOPIES is 4
- BACKUPDEVICECATEGORY is TAPE
- DATAMOVER is HSM
- NOPARALLEL
- UNITNAME specified on the SETSYS UNITNAME command
- RETENTIONPERIOD (7)
- BCDSBACKUPDSN is *uid*.BCDS.BACKUP. The *uid* is the UID parameter you specified in the DFSMSHsm startup procedure.
- MCDSBACKUPDSN is *uid*.MCDS.BACKUP. The *uid* is the UID parameter you specified in the DFSMSHsm startup procedure.
- OCDSBACKUPDSN is *uid*.OCDS.BACKUP. The *uid* is the UID parameter you specified in the DFSMSHsm startup procedure.

- JRNLBACKUPDSN is *uid*.JRNL.BACKUP. The *uid* is the UID parameter you specified in the DFSMSHsm startup procedure.

CDSVERSIONBACKUP (BACKUPCOPIES): Specifying the number of backup copies to keep for each control data set and journal data set

Explanation: **BACKUPCOPIES**(*backupcopies*) is an optional subparameter of the CDSVERSIONBACKUP parameter specifying the number of backup copies to be cataloged and inventoried by DFSMSHsm for each of the control data sets and the journal data set. For *backupcopies*, specify a decimal number from 1 to 9999 indicating the number of backup copies to be maintained. When the number of copies exceeds this limit, DFSMSHsm removes the oldest from the catalog and if on tape, invokes the ARCTVEXT installation exit when it has been activated by the SETSYS EXITON command.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: See the DFSMSHsm defaults for “CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets” on page 1109.

CDSVERSIONBACKUP (BACKUPDEVICECATEGORY): Specifying the backup device to which the control data sets and journal will be backed up

Explanation: **BACKUPDEVICECATEGORY** is an optional subparameter of the CDSVERSIONBACKUP parameter that specifies the type of device that receives the backup copies of the DFSMSHsm control data sets and the journal data set when DFSMSHsm backs them up.

Subparameter	Explanation
DASD	DFSMSHsm places the backup copies on DASD.
TAPE	DFSMSHsm places the backup copies on scratch tapes.
PARALLEL NOPARALLEL	<p>Control data set backups to tape are done in parallel (PARALLEL) or one at a time (NOPARALLEL). Backup of the CDSs is always performed in parallel if the output device is DASD. If the output is to tape, DFSMSHsm can be directed to backup the data sets in parallel by using the PARALLEL subparameter. If you prefer to use only a single tape drive at a time, use the NOPARALLEL subparameter.</p> <p>PARALLEL is the subparameter indicating that all control data sets and the journal should be backed up in parallel. By specifying this subparameter, you can cause a separate tape drive to be allocated for each control data set and the journal. Note that if DSS is the datamover, PARALLEL must be used.</p> <p>NOPARALLEL is the subparameter indicating that the control data sets and the journal should be backed up serially. When NOPARALLEL is specified, the control data sets are backed up one at a time followed by the backup of the journal. By specifying this parameter, you can cause only one tape drive to be allocated at a time for all the data set backups. NOPARALLEL is the default.</p>

Subparameter	Explanation
DENSITY (<i>density</i>)	You select the density of the scratch tape that DFSMSHsm is to mount when it backs up the control data sets and the journal data set. For <i>density</i> , specify a value of 2, 3, or 4. The decimal number 2 indicates a tape density of 32 bytes/mm (800 bpi); 3 represents a density of 63 bytes/mm (1600 bpi), and 4 represents a density of 246 bytes/mm (6250 bpi). The DENSITY subparameter applies only to reel-to-reel technology and if specified for cartridge technology, DFSMSHsm ignores it. DENSITY must be specified if the unit name specified with the UNITNAME subparameter is a reel-type tape.
EXPIRATIONDATE (<i>expirationdate</i>) RETENTIONPERIOD (<i>retentionperiod</i>)	<p>You select when the backup copies are no longer to be managed by your tape management product. They result in an expiration date being placed in the tape volume label.</p> <p>For <i>yyyyddd</i>, <i>yyyy</i> and <i>ddd</i> are the year and day of the year for the expiration date of the dump copies. For <i>yyyy</i>, specify a year between the current year and 2155. For <i>ddd</i>, specify a day between 001 and 366.</p> <p>For <i>yy</i>, specify a year between the current year and 99; the <i>yy</i> value is added to 1900 to get the year. For <i>ddd</i>, specify a day between 001 and 366.</p> <p>For <i>retentionperiod</i>, specify a 1- to 4-digit number representing the number of days DFSMSHsm must keep the backup copy before DFSMSHsm can delete it.</p>
UNITNAME (<i>unitname</i>)	You select the type of unit where DFSMSHsm is to allocate the scratch tape when DFSMSHsm backs up the control data sets and the journal data set. For <i>unitname</i> , the types of units you can request for allocating the scratch tape are 3400-3, 3400-4, 3400-5, 3400-6, 3400-9, 3480, 3480X, 3490, 3590-1, or an esoteric name you specify in the USERUNITTABLE parameter of the SETSYS command.

The SETSYS UNITNAME command must be issued prior to issuing the SETSYS CDSVERSIONBACKUP command to identify the type of tape unit used during backup or dump processing. If an esoteric tape unit name is used, the SETSYS USERUNITTABLE command must be issued prior to the SETSYS CDSVERSIONBACKUP command.

If you specify . . .	Then . . .
3400-9	Your 3480 Magnetic Tape Subsystem simulates your 3420 tape drives.
3480 or 3480X	You use all the functions of the 3480 Magnetic Tape Subsystem.
3490	You use all the functions of the 3490 Magnetic Tape Subsystem.
3590-1	You use all the functions of the 3590 Magnetic Tape Subsystem.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: See the DFSMSHsm defaults for “CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets” on page 1109.

Note:

1. Past dates are not recognized for *expirationdate*.

2. Values for *expirationdate* of 1999365 and 1999366 are recognized as “never expire” dates. Values of 1998000 and 1999000 are recognized as meaningful to certain tape management systems.

CDSVERSIONBACKUP (DATAMOVER): Specifying the type of data movement used when performing CDSVERSIONBACKUP

Explanation: **DATAMOVER(HSM | DSS)** is an optional subparameter of the CDSVERSIONBACKUP parameter specifying the type of data movement to be used when performing CDSVERSIONBACKUP of the DFSMSHsm control data sets. DFSMSdss must be specified as the datamover for CDS backup when you are using RLS mode.

Subparameter	Explanation
HSM	The DFSMSHsm control data sets are backed up using the AMS EXPORT function. The journal data set is backed up using sequential I/O.
DSS	The DFSMSHsm control data sets are backed up by invoking the DFSMSdss logical dump function for each data set. The journal is backed up using sequential I/O. Using DFSMSdss as the datamover allows you to exploit the concurrent copy function. Control data sets and the journal are always backed up in parallel.

When you use DFSMSdss rather than DFSMSHsm as the data mover, concurrent copy can be used and the CDSs will be validated as they are backed up. Concurrent copy allows the CDS backup to complete as soon as the journal has been backed up so that DFSMSHsm can begin processing requests before the CDS backups are completed.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: See the DFSMSHsm defaults for “[CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets](#)” on page 1109.

CDSVERSIONBACKUP (BCDSBACKUPDSN): Specifying the initial characters of the backup data set name for the backup control data set

Explanation: **BCDSBACKUPDSN(dsname)** is an optional subparameter of the CDSVERSIONBACKUP parameter specifying the initial characters of the data set name to be used when allocating the backup data set for the BCDS. For *dsname*, specify the initial characters of the data set name. You can use up to 35 initial characters including the periods (.); however, the 35 initial characters cannot end with a period. If the BCDS is defined as a multicluster, then only 31 characters can be used. DFSMSHsm appends a final qualifier of .Dnnnnnnn or .Vnnnnnnn to the backup name you choose. DFSMSHsm uses the total name to allocate the backup data set for the BCDS.

Note: With the multicluster CDS support, up to four names for the BCDS may be required, depending on how you have split the BCDS. DFSMSHsm generates these names using the user-supplied name (or default) and appending .DSx before the final qualifier. x will be a number 1–4 representing each cluster. For example, if the backup data set name for the BCDS is:

```
DFHSM.BCDS.BACKUP
```

and you have split the BCDS into 4 clusters, then the generated data set names will be:

```
DFHSM.BCDS.BACKUP.DS1.D0000000
DFHSM.BCDS.BACKUP.DS2.D0000000
DFHSM.BCDS.BACKUP.DS3.D0000000
DFHSM.BCDS.BACKUP.DS4.D0000000
```


If the BCDS is not defined as a multicluster, the user-defined name is used as is.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: See the DFSMSHsm defaults for [“CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets”](#) on page 1109.

CDSVERSIONBACKUP (MCDSBACKUPDSN): Specifying the initial characters of the backup data set name for the migration control data set

Explanation: **MCDSBACKUPDSN(dsname)** is an optional subparameter of the CDSVERSIONBACKUP parameter specifying the initial characters of the data set name when allocating the backup data set for the MCDS. For *dsname*, specify the initial characters of the data set name. You can use up to 35 initial characters, including the periods (.); however, the 35 initial characters cannot end with a period. If the MCDS is defined as a multicluster, then only 31 characters can be used. DFSMSHsm appends a final qualifier of .Dnnnnnnn or .Vnnnnnnn to the backup name you choose. DFSMSHsm uses the total name to allocate the backup data set for the MCDS.

Note: With the multicluster CDS support, up to four names for the MCDS may be required, depending on how you have split the MCDS. DFSMSHsm generates these names using the user-supplied name (or default) and appending .DSx before the final qualifier. x will be a number 1–4 representing each cluster. For example, if the backup data set name for the MCDS is:

```
DFHSM.MCDS.BACKUP
```

and you have split the MCDS into 2 clusters, then the generated data set names will be:

```
DFHSM.MCDS.BACKUP.DS1.D0000000
DFHSM.MCDS.BACKUP.DS2.D0000000
```

If the MCDS is not defined as a multicluster, the user-defined name is used as is.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: See the DFSMSHsm defaults for [“CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets”](#) on page 1109.

CDSVERSIONBACKUP (OCDSBACKUPDSN): Specifying the initial characters of the backup data set name for the offline control data set

Explanation: **OCDSBACKUPDSN(dsname)** is an optional subparameter of the CDSVERSIONBACKUP parameter specifying the initial characters of the data set name when allocating the backup data set for the OCDS. For *dsname*, specify the initial characters of the data set name. You can use up to 35 initial characters, including the periods (.); however, the 35 initial characters cannot end with a period. DFSMSHsm appends a final qualifier of .Dnnnnnnn or .Vnnnnnnn to the backup name you choose. DFSMSHsm uses the total name to allocate the backup data set for the OCDS.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: See the DFSMSHsm defaults for [“CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets”](#) on page 1109.

CDSVERSIONBACKUP (JRNLBACKUPDSN): Specifying the initial characters of the backup data set name for the journal data set

Explanation: **JRNLBACKUPDSN**(*dsname*) is an optional subparameter of the CDSVERSIONBACKUP parameter specifying the initial characters of the data set name when allocating the backup data set for the journal data set. For *dsname*, specify the initial characters of the data set name. You can use up to 35 initial characters, including the periods (.); however, the 35 initial characters cannot end with a period. DFSMSHsm appends a final qualifier of .Dnnnnnnn or .Vnnnnnnn to the backup name you choose. DFSMSHsm uses the total name to allocate the backup data set for the journal data set.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: See the DFSMSHsm defaults for “CDSVERSIONBACKUP: Specifying whether the control data sets should be backed up to multiple data sets” on page 1109.

CLASSTRANSITIONS: Specifying if class transitions will be performed during event-driven migration

Explanation: **CLASSTRANSITIONS** is an optional parameter that specifies settings for class transitions during interval migration and on-demand migration windows. The EVENTDRIVENMIGRATION subparameter, when set to Y (YES, the default), indicates that data sets will be class transitioned during event-driven migration. A setting of N (NO) indicates that class transitions will *not* occur during event-driven migration processing and only occur during primary space management.

The SERIALIZATIONEXITS subparameter, when set to Y (YES) indicates that the user data set serialization error exits will be used during event driven migration. A setting of N (NO, the default) indicates that the user data set serialization error exits will *not* be used during event-driven migration.

SMS relationship: This keyword only applies to SMS managed volumes.

SETSYS default: Same as DFSMSHsm default.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, class transitions will be performed during event-driven migration, but user data set serialization error exits will *not* be used.

CLOUDMIGRATION: Specifying the type of migration to CLOUD

CLOUDMIGRATION is an optional parameter specifying that you want DFSMSHsm to attempt to use fast subsequent migration when migrating to the CLOUD.

The following are required and optional keywords for the CLOUD parameter:

RECONNECT

Data sets eligible for fast subsequent migration can be reconnected to the CLOUD from which they were recalled. A data set can be reconnected if all fast subsequent migration requirements are satisfied. For more information about these requirements, see [Fast Subsequent Migration \(reconnection\) eligibility requirements](#).

NONE

Data sets are not eligible for fast subsequent migration and are not reconnected to the CLOUD from which they were recalled.

ALL

All data sets that are eligible for standard migration and satisfy fast subsequent migration eligibility requirements are reconnected to the CLOUD that the data set was recalled from; including data sets that would have otherwise been migrated to an ML1 or ML2 volume.

Alias: CLM

CLOUD: Specifying CLOUD properties

CLOUD is an optional parameter that allows specification of properties specific to a particular Object Storage that DFSMSHsm uses to store full volume dump and migration copies of data sets. If specified in a DFSMSHsm parmlib member, this parameter is ignored.

The following are required and optional keywords for the CLOUD parameter:

NAME(*cloud_network_connection_name*)

cloud_network_connection_name is the name of a defined SMS Cloud Construct or a CDA provider file or a CDA provider file. This is an optional parameter when REFRESH is specified. Otherwise, this is a required parameter.

If the specified name does not exist in the current SMS configuration nor in the CDA providers directory, the command does not succeed.

CDACREDENTIALS

is an optional parameter that specifying this optional keyword requests that DFSMSHsm obtain the password for the *cloud_network_connect_name* from the CDA (cloud data access) where the password is entered by the security administrator. A *cloud_network_connection_name* previously defined by CCREDS is transitioned to use a password obtained from CDA with the issuance of this command with the CDACREDS keyword.

Also, the reissuance of the command with this keyword for an existing *cloud_network_connect_name* refreshes the stored cloud properties entry with the credentials that are reobtained from CDA. Refreshing these stored entries also have the stored credentials reencrypted. This technique is used to update the DFSMSHsm stored cloud properties when the cloud password is changed by the security administrator or when a cloud network connection name is added to the SMS configuration.

CDACREDS option stores up to 248 *cloud_network_connection_name* entries and CCREDS option stores 7 entries, the combination allows 255 *cloud_network_connection_name* entries to be stored by DFSMSHsm.

Aliases: CDACREDSTORE, CDACREDS

CLOUDCREDENTIALS

Specifying this optional keyword requests that DFSMSHsm request the password associated with the Object Storage account from the storage administrator using WTOR. The reply to the WTOR is suppressed on the console. If the password is mixed case, and the system has a dual-case console, enclosing the password in single quotation marks ensures that it is passed as-is to DFSMSHsm. DFSMSHsm will store the password for future requests to the Cloud Storage. The cloud password must be secured. It is recommended that the security administrator who is administering the cloud storage accounts is the one who enters this password.

A *cloud_network_connection_name* previously defined by CDACREDS is transitioned to use a password obtained from the WTOR with the reissuance of the command with this keyword.

DFSMSHsm can only store seven cloud property entries that are specified with the CCREDS keyword. A transition to a WTOR specified password might not succeed because of the limited number of cloud property entries that can be stored by DFSMSHsm. With the CDACREDS keyword, up to 248 *cloud_network_connection_name* property entries are stored where the security administrator entered the password with CDA. For these reasons, you should request CDACREDS with the SETSYS command.

Aliases: CCREDS, PASSWORD, PW

REMOVECREDENTIALS

Specifying this optional keyword requests that DFSMSHsm delete the information that is related to the Object Storage account from its record. You can use this keyword to remove old invalid entries. The FIXCDS command to display the MHCR record can be used to display the Object Storage accounts known to DFSMSHsm.

Aliases: REMOVE, DELETE, DEL

REFRESH

Specifying this optional keyword requests that DFSMSHsm reload the Cloud credentials into memory from the Migration Control Data Set (MCDS). This option is used on other DFSMSHsm hosts in a sysplex after SETSYS CLOUD(<name> CDACREDENTIALS) was used on one system to change the stored password. You can also use this option if you want DFSMSHsm to re-validate that OMVS is available for DFSMSHsm.

Aliases: REF

CDAPROVIDER and SMSPROVIDER

CDAPROVIDER and SMSPROVIDER are mutually exclusive keywords specifying whether DFSMSHsm should look for cloud information in a CDA provider file or an SMS network connection construct. When REMOVE or REFRESH is specified along with CDAPROVIDER or SMSPROVIDER, CDAPROVIDER and SMSPROVIDER keywords are ignored.

- **CDAPROVIDER** specifies cloud provider properties are defined in a CDA provider file. DFSMSHsm looks for a CDA provider file with the specified CLOUD name. When a valid, nonempty provider file is found, DFSMSHsm will use CDA API for cloud operations. If a valid CDA provider file is not found, the SETSYS CLOUD command fails.

When DFSMSHsm is obtaining information from a CDA provider file, the **tctType** key field is checked to determine how to connect to the cloud object store and transfer data.

- **TAPE-OBJECT**: When tctType value specifies “TAPE-OBJECT” in the CDA provider file, it describes a TS7700 advanced object store. DFSMSHsm supports multiple endpoints for TS7700 advanced object store if the functionality has been enabled. All data is moved through the DS8000 cloud proxy using either FICON or Ethernet connections.
- **TCT**: When tctType value species “TCT” in the CDA provider file, data will be transferred using TCT. All user data is moved through the DS8000 cloud proxy using FICON connections. All metadata and utility operations go through the storage cloud via Ethernet instead of using the DS8000 proxy.
- When tctType is not specified in the CDA provider file, the connection type is **direct to cloud**. DFSMSHsm uses CDA API to transfer both metadata and user data extents via Ethernet connections.

Alias: CDAPROV

- **SMSPROVIDER** specifies cloud provider properties are defined in an SMS network connection construct and DFSMSHsm should not look for a valid, non-empty CDA provider file. When SMSPROVIDER is specified, either CCREDS or CDACREDS must also be specified.

Alias: SMSPROV

Defaults: When neither CDAPROVIDER nor SMSPROVIDER is specified, DFSMSHsm does the following when the designated cloud is a TS7700 TAPE-OBJECT store:

- When multiple endpoints support functionality has been enabled, DFSMSHsm first looks for a valid, non-empty CDA provider file with the name of a TS7700 TAPE-OBJECT store in order to provide multiple endpoints support. If a valid, non-empty CDA provider file is not found, DFSMSHsm will continue to look for an SMS network connection construct for the specified cloud, and multiple endpoints support is unavailable.
- When multiple endpoints support is not enabled, DFSMSHsm uses cloud information defined in the SMS network connection construct.

Related Reading

- For more information about multiple endpoints support, see [Allowing multiple endpoints support for TCT TS7700 advanced object store \(tape object\) in z/OS DFSMSHsm Implementation and Customization Guide](#).
- For more information about using CDA provider files and associated key credentials, see [DFSMSHsm and cloud storage in z/OS DFSMSHsm Implementation and Customization Guide](#).

Restriction: DFSMSHsm can only store seven CLOUD property entries when the CLOUDCREDENTIALS option is used. Up to 248 cloud property entries can be stored with the CDACREDENTIALS keyword. The combination allows 255 cloud property entries to be stored by DFSMSHsm.

Aliases: CLD

COMMONQUEUE: Specifying settings for the common queues

Explanation: **COMMONQUEUE** is an optional parameter that specifies settings for the DFSMSHsm common queues.

The subparameters of the **COMMONQUEUE** parameter follow this discussion.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: None.

COMMONQUEUE (DUMP(CONNECT(*basename*))): Specifying the XCF functional group name suffix for remote dump processing

Explanation: **DUMP(CONNECT(*basename*))** is an optional subparameter of the **COMMONQUEUE** parameter. It specifies the XCF functional group name suffix that this host uses for remote dump processing. When it is specified, this system is allowed to send and receive dumps to and from other systems in the HSMplex that were started with the same *basename*. For *basename*, substitute four characters. The four characters can be numeric, uppercase alphabetic, and national characters. The XCF group name generated by the system from this parameter is **ARCD*basename***. For example, if you specify **DMP1** for *basename*, the resulting XCF group name that this host will become a member of is **ARCD**DMP1****.

This system becomes a candidate for being the common dump queue master scheduler when the **MSC** keyword is not specified, or specified as **MSC** with the **CONNECT** or **MSC(YES)** on another separate command. If this master scheduler candidate system is selected during DFSMSHsm startup or during dump to be the master scheduler of this functional group, this system is considered to be the enabled master scheduler for this functional group of systems.

Subsequent SETSYS commands after a **CONNECT** can be issued to change the **MASTERSCHEDULERCANDIDATE** state of this member of the dump group. A specification of **MSC(NO)** will remove this member as a master scheduler candidate, and if the member was currently active as the master scheduler, it will be stopped. The system will reassign the master scheduler to another dump group member that was defined as a candidate, if one is available. New incoming work will be assigned to run in this new master scheduler member. Active and queued work will be allowed to complete. A specification of **MSC(YES)** will allow this member to become the master scheduler if needed by the dump group.

If it is known at startup that a system is not a candidate for being the master scheduler, but the system needs to be a member of the dump group, the **MSC(NO)** and **CONNECT(*basename*)** subparameters should be specified together in the **DUMP** keyword.

Remote dump processing in the group will not be performed if all of the group members are not allowed to be a master scheduler with the **MSC(NO)** specification.

Use the **HOLD DUMP** command to prevent additional command and automatic dump processing tasks being started by this system. See the **HOLD DUMP** command ([“DUMP: Preventing dumps” on page 902](#)) for more information.

When the **SETSYS COMMONQUEUE(DUMP(CONNECT(*basename*)))** command is specified, the preference by the system will be to utilize the available dump tasks as defined by **MAXDUMPTASKS** on all the dump group members in the HSMplex. The maximum and minimum stack value on the dump class or **BACKVOL** command determines the utilization of the available tasks in the dump group.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: If the DUMP(CONNECT) subparameter is not specified, no XCF dump groups will be defined and remote processing of dumps will not be performed.

Note: A system must disconnect itself from using XCF groups before it attempts to connect to a new XCF group.

For more information: Refer to [“Common dump queue” on page 466](#).

COMMONQUEUE(DUMP(DISCONNECT)): Specifying that a system disconnect itself from using the currently connected XCF functional group for dump processing

Explanation: DUMP(DISCONNECT) is an optional sub-parameter of the COMMONQUEUE parameter which specifies that a system disconnect itself from using the currently connected XCF functional group for dump processing. If the system is the enabled master scheduler for this group, it will be stopped and the master scheduler role will be assigned to another candidate. When you specify the DUMP(DISCONNECT) subparameter, the system performs the following functions:

- Directs all new dump requests to its local queue.
- Stops requesting dumps to be sent to remote systems.
- Stops receiving dumps from remote systems.
- Allows active requests from the common queue running in this system to finish. Also requests submitted by this host and running in the common group in other systems will be allowed to finish.
- The FORCE option, DUMP(DISCONNECT FORCE), allows requests from the common queue running in this system to finish. Requests submitted to the common queue from this system are failed locally on the submitting host but may continue to be processed elsewhere in the CDQ.
- The FORCE option, DUMP(DISCONNECT FORCE), on a master scheduler system will allow requests from the common queue for this function running in the group to finish. However, any request received from remote members in the group on this master scheduler system and are waiting to be processed will be failed.
- If the system was the master scheduler for this function, this role will be assigned to another candidate master scheduler system in the group.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: None.

For more information: Refer to [“Common dump queue” on page 466](#).

COMMONQUEUE (DUMP(MASTERSCHEDULERCANDIDATE)): Specifying whether the host is a master scheduler candidate for grouped function of dump

Explanation: DUMP(MASTERSCHEDULERCANDIDATE)) is an optional subparameter of the COMMONQUEUE parameter. This specifies whether the host is a master scheduler candidate for the common dump queue.

Subsequent SETSYS commands after a CONNECT can be issued to change the MASTERSCHEDULERCANDIDATE state of this member of the dump group. A specification of MSC(NO) will remove this member as a master scheduler candidate, and if the member was currently active as the master scheduler, it will be stopped. The system will reassign the master scheduler to another dump group member that was defined as a candidate, if one is available. New incoming work will be assigned to run in this new master scheduler member. Active and queued work will be allowed to complete. A

specification of MSC(YES) will allow this member to become the master scheduler if needed by the dump group. This stand-alone MSC specification will have no effect until the host is connected to the group.

If it is known at startup that a system is not a candidate for being the master scheduler, but the system needs to be a member of the dump group, the MSC(NO) and CONNECT(*basename*) subparameters should be specified together in the DUMP keyword.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: MASTERSCHEDULERCANDIDATE(Y), but will have no effect until a SETSYS CQUEUE(DUMP(CONNECT(*basename*))) command is issued.

For more information: Refer to [“Common dump queue” on page 466](#).

COMMONQUEUE (RECALL(CONNECT(*basename*))): Specifying the base name of the list structure for CRQ host connection

Explanation: RECALL(CONNECT(*basename*)) is an optional subparameter of the COMMONQUEUE parameter. This subparameter specifies the base name of the list structure that a host connects to for common recall queue (CRQ) activities. The list structure name must be defined in the active CFRM policy. For *basename*, substitute five characters. The five characters can be numeric, uppercase alphabetic, and national characters. The list structure name is SYSARC_*basename*_RCL. For example, if you specify PLEX1 for *basename*, the resulting list structure is SYSARC_PLEX1_RCL.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: None.

Note: A host must disassociate itself from one list structure before it attempts to connect to another structure. To disassociate, a host must disconnect from the list structure, and then you must define the new structure in the active CFRM policy. DFSMSHsm will attempt to connect to the new structure when it is defined in the active CFRM policy.

COMMONQUEUE (RECALL(DISCONNECT)): Specifying disconnection from the CRQ

Explanation: RECALL(DISCONNECT) is an optional subparameter of the COMMONQUEUE parameter which specifies that a host disconnect from the common recall queue (CRQ). When you specify the RECALL(DISCONNECT) subparameter, the host performs the following functions:

- Directs all new recall requests to the local recall queue
- Moves all recall requests that originated on that host from the CRQ to the local queue
- Completes any remote requests that were previously selected from the CRQ
- Stops selecting requests from the CRQ

The host disconnects from the CRQ when the following conditions are met:

- The host has finished processing all the remote requests that were previously selected from the CRQ
- All requests originating on that host have moved from the CRQ to the local queue
- Processing is complete for all requests that originate on that host that have been selected for remote processing

The disconnection must be complete before the host can connect to another CRQ.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: None.

Note: This command does not need to be issued before a DFSMSHsm shutdown. Disconnection from the CRQ is part of the normal shutdown process.

COMMONQUEUE RECOVER(CONNECT(basename)): Specifying the XCF functional group name suffix for remote full volume recover from dump processing

Explanation: RECOVER(CONNECT(basename)) is an optional subparameter of the COMMONQUEUE parameter.

This subparameter specifies the XCF functional group name suffix that this host uses for volume restores from dumps. This includes the FRRECOV COPYPOOL FROMDUMP, FRRECOV TOVOLUME FROMDUMP, and RECOVER * FROMDUMP volume restore commands. When specified, this host is allowed to send volume restores to the common recover queue for processing in the HSMplex with the same basename. For basename, substitute four characters. The 4 characters can be numeric, uppercase alphabetic, and national characters. The XCF group name is ARCVbasename. For example, if you specify REC1 for basename, the resulting XCF group name that this host becomes a member of is ARCVREC1.

This host becomes a candidate for the common recover queue master scheduler when the MSC keyword is not specified, or when specified as MSC with the CONNECT command or MSC(YES) on another separate command. If this master scheduler candidate system is selected during HSM startup or during recovery to be the Master Scheduler of this group, this system is considered to be the Master Scheduler for the group.

Subsequent SETSYS commands after a CONNECT can be issued to change the MASTERSCHEDULERCANDIDATE state of this member of the recover group. A specification of MSC(NO) remove this member as a master scheduler candidate. In addition, if the member was currently active as the master scheduler, it is stopped. The system reassigns the master scheduler to another recover group member that is defined as a candidate. New incoming work is assigned to run in this new Master Scheduler member. Active and queued work is allowed to complete. A specification of MSC(YES) allows this member to become the Master Scheduler if needed by the recover group. This stand-alone MSC specification has no effect when it is issued on a host that is not connected to the group.

If it is known at startup that a host is not a candidate for the master scheduler role, but the host needs to be a member of the recover group, then MSC(NO) and CONNECT(basename) subparameters can be specified together in the RECOVER keyword. Remote recover processing in the group is not performed when none of the group members are master scheduler candidates with the MSC(NO) specification.

Use the HOLD RECOVER command to prevent this system from starting additional recover processing tasks. See HOLD RECOVER command for details.

When the SETSYS COMMONQUEUE(RECOVER(CONNECT(basename))) is specified, the preference by the system preference is to use the available recover tasks as defined by MAXDUMPRECOVERTASKS on all the recover group members in the HSMplex. When this SETSYS is not specified, only local recover tasks, where the command was issued will be utilized. This reduced parallelism might lengthen the total time needed for the volume recovery to complete. In either case, the number of concurrent tasks is further limited by the stacking of the volumes on the dump tapes. For example, if all 100 volumes on a single tape are being recovered, then only a single recover task is used.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: If the RECOVER(CONNECT) subparameter is not specified, no XCF recover groups will be defined. Local processing of volume restores will be performed.

Note: A system must disconnect itself, and stop using XCF groups before it attempts to connect to a new XCF group.

The host will respond with message ARC1566I when it connects to the common recover group, and with message ARC1567I when it takes on the master scheduler responsibilities for it.

For more information: Refer to [“Common recover queue” on page 474](#).

COMMONQUEUE RECOVER(DISCONNECT): Specifying that a system disconnect itself from using the currently connected XCF functional group for recover processing

Explanation: RECOVER(DISCONNECT) is an optional subparameter of the COMMONQUEUE parameter that specifies that a host disconnect itself to stop using the currently connected common recover queue (CVQ) for processing. The following considerations apply to this disconnect process:

- All CVQ requests being processed on this host, and requests that originated from this host and are selected to be processed in the CVQ, are waited upon for completion before the host is disconnected from the CVQ. This host will no longer be selected to process new volume restore work from the CVQ.
- A HOLD RECOVER issued prior to the disconnect command prevents queued requests from being selected to be processed. After the disconnect has completed, a RELEASE RECOVER allows the queued requests to be processed locally.
- RECOVER after the disconnecting allows these requests to be processed locally.
- To verify the disconnection, use the QUERY ACTIVE command and look for message ARC1560I.
- Recover resources available on this host after the disconnection has completed are not known or available to the remaining CVQ members. All new recover requests received are placed on this host's local queue and processed locally.
- For shutdown procedures, a DISCONNECT should first be issued on each host being removed from the HSMplex to allow CVQ requests to complete first.
- The DISCONNECT FORCE option should be reserved for emergency situations. With FORCE specified, requests that originated from this host and are selected to be processed in the CVQ are not waited upon for completion before disconnecting. Messages are issued to indicate that a request has been failed but still may complete in the CVQ.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: None.

The host will respond with message ARC1567I when the host disconnects from the group, and with message ARC1568I if master scheduler responsibilities are lost.

Explanation: SETSYS CQUEUE(RECOVER) with no options specified will parse out with an error.

For more information: Refer to [“Common recover queue”](#) on page 474.

COMMONQUEUE RECOVER(MASTERSCHEDULERCANDIDATE): Specifying whether the host is a master scheduler candidate for grouped function of recover

Explanation: RECOVER(MASTERSCHEDULERCANDIDATE)) is an optional sub-parameter of the COMMONQUEUE parameter. This specifies whether the host is a master scheduler candidate for grouped function of recover.

Subsequent SETSYS commands after a CONNECT can be issued to change the MASTERSCHEDULERCANDIDATE state of this member of the recover group. A specification of MSC(NO) removes this member as a master scheduler candidate. In addition, if the member is currently active as the master scheduler, it is stopped. The system reassigns the master scheduler to another recover group member that is defined as a candidate, if one is available. New incoming work is assigned to run in this new master scheduler member. Active and queued work is allowed to complete. A specification of MSC(YES) allows this member to become the master scheduler if the recover group needs it.. This stand-alone MSC specification has no effect until the host is connected to the group. If it is known at startup that a system is not a candidate for the master scheduler role, but the system needs to be a member of the recover group, the MSC(NO) and CONNECT(basename) sub-parameters should be specified together in the RECOVER keyword.

If it is known at startup that a system is not a candidate for being the Master Scheduler but the system needs to be a member of the recover group, the MSC(NO) and CONNECT(basename) sub-parameters should be specified together in the RECOVER keyword.

SMS relationship: Not applicable.

SETSYS default: None.

DFSMSHsm default: MASTERSCHEDULERCANDIDATE(Y), but will have no effect until a SETSYS CQUEUE(RECOVER(CONNECT(groupname))) is issued.

The host will respond with message ARC15668I if the host lost its Master Scheduler responsibilities and with message ARC1569I if it took them on.

For more information: Refer to [“Common recover queue”](#) on page 474.

COMPACT: Specifying when compaction should be done

Explanation: **COMPACT** is an optional parameter set that you use to specify the type of compaction during migration or backup for all data sets.

The subparameters of the COMPACT parameter follow this discussion in alphabetical order. The following are optional subparameters:

- ALL | NONE
- DASDBACKUP
- DASDMIGRATE
- TAPEBACKUP
- TAPEMIGRATE

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets. This COMPACT parameter is only used for SMS data sets that are not compressed on L0; as for non-SMS data sets, it's always in effect since you cannot have a non-SMS dataset compressed on L0.

SETSYS default: If you specify the COMPACT parameter without specifying any subparameters, the SETSYS default is no compaction.

DFSMSHsm default: If you do not specify a subparameter with this parameter on any SETSYS command, the DFSMSHsm default is no compaction.

COMPACT ALL and NONE: Specifying whether to compact data sets

Explanation: **ALL | NONE** are mutually exclusive, optional subparameters of the COMPACT parameter specifying whether or not to compact a data set during migration or backup.

Subparameter	Explanation
ALL	DFSMSHsm can compact a data set during migration to DASD, migration to tape, backup to tape, and backup to DASD.
NONE	DFSMSHsm does not compact a data set during migration or backup.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify any subparameter of the COMPACT parameter on any SETSYS command, the DFSMSHsm default is NONE.

Note:

1. When you specify the ALL or NONE subparameter with the COMPACT parameter, DFSMSHsm ignores all other subparameters.
2. DFSMSHsm compacts cataloged and uncataloged data sets.
3. Since encrypted data sets do not compress, they will not be compacted during migration or backup.

COMPACT DASDBACKUP and NODASDBACKUP: Specifying whether to compact data sets during backup to DASD

Explanation: **DASDBACKUP** | **NODASDBACKUP** are mutually exclusive, optional subparameters of the COMPACT parameter specifying whether or not to compact a data set when DFSMSHsm backs it up to DASD.

Subparameter	Explanation
DASDBACKUP	DFSMSHsm can compact a data set when DFSMSHsm backs it up to DASD.
NODASDBACKUP	DFSMSHsm does not compact a data set when DFSMSHsm backs it up to DASD.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is NODASDBACKUP.

Note:

1. If you do not want a specific data set to be compacted during volume backup, use the data set backup exit (ARCBDEXT) to prevent compaction.
2. Since compressed format data sets are already compressed, they will not be compacted during backup to DASD.
3. Since encrypted data sets do not compress, they will not be compacted during backup to DASD..

COMPACT DASDMIGRATE and NODASDMIGRATE: Specifying whether to compact data sets during migration to DASD

Explanation: **DASDMIGRATE** | **NODASDMIGRATE** are mutually exclusive, optional subparameters of the COMPACT parameter specifying whether or not to compact a data set during migration to DASD.

Subparameter	Explanation
DASDMIGRATE	DFSMSHsm can compact a data set during migration to DASD.
NODASDMIGRATE	DFSMSHsm does not compact a data set during migration to DASD.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is NODASDMIGRATE.

Note:

1. If you do not want a specific data set compacted during volume migration to DASD, use the data set migration exit (ARCMDEXT) to prevent compaction.

2. If you specified the TAPEMIGRATION(DIRECT) parameter of the SETSYS command, DFSMSHsm ignores the DASDMIGRATE or NODASDMIGRATE subparameter if you specify one of them.
3. Although the MIGRATE and NOMIGRATE subparameters no longer exist, DFSMSHsm does not fail the command if you specify them. Instead, MIGRATE and NOMIGRATE have the same meaning as DASDMIGRATE and NODASDMIGRATE.
4. Since compressed format data sets are already compressed, they will not be compacted during migration to DASD.
5. Since encrypted data sets do not compress, they will not be compacted during migration to DASD.

COMPACT TAPEBACKUP and NOTAPEBACKUP: Specifying whether to compact data sets during backup to tape

Explanation: **TAPEBACKUP** | **NOTAPEBACKUP** are mutually exclusive, optional subparameters of the COMPACT parameter specifying whether or not to compact a data set when DFSMSHsm is backing it up to tape.

Subparameter	Explanation
TAPEBACKUP	DFSMSHsm can compact a data set when DFSMSHsm backs it up to tape.
NOTAPEBACKUP	DFSMSHsm does not compact a data set when DFSMSHsm backs it up to tape.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is NOTAPEBACKUP.

Note:

1. If you do not want a specific data set to be compacted during volume backup, use the data set backup exit (ARCBDEXT) to prevent compaction.
2. Since compressed format data sets are already compressed, they will not be compacted during backup to tape.
3. Since encrypted data sets do not compress, they will not be compacted during backup to tape.

COMPACT TAPEMIGRATE and NOTAPEMIGRATE: Specifying whether to compact data sets during migration to tape

Explanation: **TAPEMIGRATE** | **NOTAPEMIGRATE** are mutually exclusive, optional subparameters of the COMPACT parameter specifying whether or not to compact a data set when DFSMSHsm is migrating it to a tape migration level 2 volume.

Subparameter	Explanation
TAPEMIGRATE	DFSMSHsm can compact a data set when DFSMSHsm migrates it to a tape migration level 2 volume.
NOTAPEMIGRATE	DFSMSHsm does not compact a data set when DFSMSHsm migrates it to a tape migration level 2 volume.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMShsm default: If you do not specify either subparameter on any SETSYS command, the DFSMShsm default is NOTAPEMIGRATE.

Note:

1. If you do not want a specific data set to be compacted during volume migration, use the data set migration exit (ARCMDEXT) to prevent compaction.
2. Since compressed format data sets are already compressed, they will not be compacted during migration to tape.
3. Since encrypted data sets do not compress, they will not be compacted during migration to tape.

COMPACTPERCENT: Specifying the percentage of space saved for compaction

Explanation: **COMPACTPERCENT**(*pct*) is an optional parameter specifying the percentage of space saved if DFSMShsm compacts all data sets. For *pct*, substitute a decimal number from 0 to 99 to specify the least amount of space you want saved if DFSMShsm compacts a data set.

If you request compaction, DFSMShsm compacts a data set when it migrates or backs up the data set for the first time. DFSMShsm then compares the number of bytes written to the total bytes of the original data set and computes the percentage of bytes saved. If the percentage saved is not greater than or equal to *pct*, DFSMShsm does not compact the data set during subsequent migrations or backups. DFSMShsm does not check whether or not the data set was compacted during migration if DFSMShsm is currently backing up the data set. Similarly, DFSMShsm does not check whether or not the data set was compacted during backup if DFSMShsm is currently migrating the data set.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMShsm default: If you do not specify this parameter on any SETSYS command, the DFSMShsm default is 40.

Note:

1. DFSMShsm makes this comparison only for cataloged data sets.
2. DFSMShsm compacts a data set only if it can save at least one track on DASD or the data set migrated to an SDSP data set.
3. Compaction always occurs during backup processing if SETSYS COMPACTPERCENT(0) is specified and DFSMShsm is to compact the data set.

CONVERSION and NOCONVERSION: Specifying whether reblocking is to occur

Explanation: **CONVERSION** | **NOCONVERSION** are mutually exclusive, optional parameters specifying whether DFSMShsm reblocks eligible data sets during recall or recovery. This parameter is not applicable to system-reblockable data sets, whether they are SMS or non-SMS.

CONVERSION specifies that DFSMShsm can reblock the data sets during recall or recovery.

REBLOCKTOANY is an optional subparameter of the CONVERSION parameter specifying the type of reblocking during recall or recovery of data sets that use DFSMShsm as the data mover and also specifies reblocking to any DASD device type supported by DFSMShsm.

NOCONVERSION specifies that DFSMShsm cannot reblock the data sets even if you specified the data set reblock exit.

SMS relationship: Parameter applies in some instances to SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is NOCONVERSION.

Note:

1. DFSMSHsm reblocks only nonkeyed, physical sequential data sets with the following record organizations: fixed-blocked, fixed-blocked-standard, variable-blocked, or variable-blocked-spanned. If a data set specified LRECL=X, DFSMSHsm does not support reblocking.

DFSMSdss reblocks only physical sequential and partitioned data sets with the following record organizations: fixed-blocked, fixed-block-standard, and variable-blocked.
2. You must specify the CONVERSION parameter if you want to use the data set reblock exit (ARCCDEXT). You can prevent DFSMSHsm from reblocking user-selected data sets by using the data set reblock exit.

CSALIMITS and NOCSALIMITS: Specifying whether to use CSA limiting for MWE allocation

Explanation: **CSALIMITS** | **NOCSALIMITS** are mutually exclusive, optional parameters that you use to specify whether common service area (CSA) limiting of management work element (MWE) storage allocation is to be used during DFSMSHsm processing. The MWE describes a requested function to be performed by DFSMSHsm.

Note: DFSMSHsm acquires storage in extended CSA (ECSA) for MWE's. Since DFSMSHsm is using ECSA, more CSA storage is available for other applications

CSALIMITS specifies the values for CSA storage allocation used during DFSMSHsm processing.

The subparameters of the CSALIMITS parameter follow this discussion in alphabetical order. These optional subparameters are:

- ACTIVE
- INACTIVE
- MAXIMUM
- MWE

NOCSALIMITS specifies that CSA limiting should not be done during DFSMSHsm processing.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is CSALIMITS.

CSALIMITS (ACTIVE): Specifying the percentage of the maximum limit to be allocated to any type of MWE

Explanation: **ACTIVE(percent1)** is an optional subparameter of the CSALIMITS parameter that you use to specify the percentage of CSA storage that DFSMSHsm can allocate to all types of MWEs when DFSMSHsm is active. After the percentage is reached, only batch WAIT MWEs are added to the CSA queue. All other requests are failed until CSA storage usage drops below the limit or you change the limit. For *percent1*, specify a 1-to 3-character number indicating the percentage of CSA storage to be allocated to all types of MWEs. The number can range from 0 to 100.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this subparameter on any SETSYS command, the DFSMSHsm default is 90.

CSALIMITS (INACTIVE): Specifying the percentage of the maximum limit to be allocated to NOWAIT MWEs

Explanation: **INACTIVE(*percent2*)** is an optional subparameter of the CSALIMITS parameter specifying the percentage of CSA storage that DFSMSHsm can allocate to NOWAIT MWEs while DFSMSHsm is inactive. After the percentage is reached, all requests are failed until DFSMSHsm is started or until you change the limit. For *percent2*, specify a 1-to 3-character number indicating the percentage of CSA storage to be allocated to NOWAIT MWEs while DFSMSHsm is inactive. The number can range from 0 to 100.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this subparameter on any SETSYS command, the DFSMSHsm default is 30.

CSALIMITS (MAXIMUM): Specifying the maximum number of CSA bytes allocated to all types of MWEs

Explanation: **MAXIMUM(*Kbytes*)** is an optional subparameter of the CSALIMITS parameter specifying the maximum number of KBs of CSA storage to be allocated to MWEs. For *Kbytes*, specify a 2- to 5-character decimal number indicating the number of bytes of CSA storage to be allocated to MWEs. You can specify a value from 10 to 16383. If you specify 16384 or higher, no limiting of the number of bytes allocated in CSA is done. If the value for MAXIMUM is exceeded, DFSMSHsm fails all new requests until CSA usage drops below the maximum limit or you change the value for MAXIMUM.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this subparameter on any SETSYS command, the DFSMSHsm default is 100.

CSALIMITS (MWE): Specifying maximum number of NOWAIT MWEs per address space

Explanation: **MWE(*#mwes*)** is an optional subparameter of the CSALIMITS parameter specifying the maximum number of NOWAIT MWEs submitted by a single address space that is to be kept on the CSA queue until completed. NOWAIT MWEs in excess of the maximum number will be removed from the CSA queue as soon as they are copied into the DFSMSHsm address space. For *#mwes*, specify a 1-to 4-character decimal number indicating how many NOWAIT MWEs per address space are kept on the CSA queue until completed. To limit the number of NOWAIT MWEs kept on the CSA queue, specify a number between 0 and 8191. If you specify 0, all NOWAIT MWEs are removed from the CSA queue as soon as they are copied into the address space of DFSMSHsm. If you specify 8192 or higher, the number of MWEs to keep on the CSA queue per address space is limited only by the amount of CSA available, as specified in the MAXIMUM subparameter.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this subparameter on any SETSYS command, the DFSMSHsm default is 4.

Note: If DFSMSHsm is inactive, the NOWAIT MWEs will not be removed from the CSA queue. In that case, the number of NOWAIT MWEs submitted by all address spaces that will be added to the CSA queue is controlled by the INACTIVE subparameter.

DAYS: Specifying the minimum migration age of a data set

Explanation: **DAYS(days)** is an optional parameter specifying the number of contiguous days a data set must remain unreferenced before the data set is eligible for migration. DFSMSHsm uses this parameter to migrate data sets from primary volumes to migration level 1 volumes under the following conditions:

- The primary volume does not already have a minimum migration age.
 - For automatic migration, DFSMSHsm checks the minimum migration age you specified with the ADDVOL command. If you did not specify the MIGRATE(days) parameter of the ADDVOL command, or if you specified MIGRATE(0) with the ADDVOL command, or if you specified MIGRATE(1) with the ADDVOL command in a multiple-DFSMSHsm-host environment and did not specify USERDATASETSERIALIZATION, then DFSMSHsm uses the minimum migration age you specify with the SETSYS DAYS parameter.
 - For command migration, DFSMSHsm checks the minimum migration age you specified with the MIGRATE command. If you did not specify a value for days with the MIGRATE command, DFSMSHsm checks whether you specified the minimum migration age with the ADDVOL command. If you also did not specify the MIGRATE(days) parameter of the ADDVOL command, or you specified MIGRATE(0) with the ADDVOL command, or you specified MIGRATE(1) with the ADDVOL command in a multiple-DFSMSHsm-host environment and did not specify USERDATASETSERIALIZATION, then DFSMSHsm uses the minimum migration age you specify with the SETSYS DAYS parameter.
- For nonmanaged volumes, DFSMSHsm checks whether you used the MIGRATE command to specify the minimum migration age. If you did not, DFSMSHsm uses the age you specify with this parameter.

The value you specify for *days* depends on your environment. If USERDATASETSERIALIZATION has been specified, DAYS can be set between 0 and 999. If not, DAYS can be set between 1 and 999 in a single-DFSMSHsm-host environment, and between 2 and 999 in a multiple-DFSMSHsm-host environment.

SMS relationship: Parameter does not apply to SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is one day if DFSMSHsm is running in a single-DFSMSHsm-host environment or two days if DFSMSHsm is running in a multiple-DFSMSHsm-host environment.

Note: DFSMSHsm uses the DAYS value when you have not defined thresholds of occupancy for a volume. If you have defined thresholds of occupancy for the volume, the DAYS parameter is another way to set a lower limit for migration.

DEBUG and NODEBUG: Specifying debug mode

Explanation: DEBUG and NODEBUG are mutually exclusive, optional parameters specifying whether DFSMSHsm is to operate in debug mode.

DEBUG specifies that DFSMSHsm is to operate in debug mode. A command to process only one data set is not affected by the DEBUG parameter. You use DEBUG to monitor the effect of DFSMSHsm on your computing system. DFSMSHsm does all automatic and command volume space management, backup, dump, and command expiration of backup versions functions without moving or deleting any data sets. DEBUG specifies that DFSMSHsm is not to move any data. Because you can monitor the data sets and volumes that DFSMSHsm would have managed, you can decide which data sets you do not want DFSMSHsm to process.

Debug mode can be entered at any time during DFSMSHsm processing. During automatic volume space management, command volume space management, or backup functions, DFSMSHsm checks for debug mode before processing each data set. During automatic dump, command dump, or volume recovery functions, DFSMSHsm checks for debug mode before processing each volume. During the expiration of backup versions function, DFSMSHsm checks for debug mode at the start of each command. If a SETSYS DEBUG command is issued before the start of the next data set during space management or backup, or before the start of the next volume during dump or recovery, the next data set or volume to be processed will be processed in debug mode, and no data will be moved or deleted. Any processing accomplished before entering debug mode is not affected.

NODEBUG specifies that DFSMSHsm is not to operate in debug mode.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is NODEBUG.

Note:

1. Debug mode causes all ABACKUP and ARECOVER commands to be processed with the VERIFY option at the time the command is selected to be started.
2. Debug mode does not affect the BACKVOL CONTROLDATASETS command and the command will be processed as normal.
3. To use the debug mode, decide which volumes you want DFSMSHsm to manage. Then, run DFSMSHsm in debug mode. When you are satisfied that the volumes are being managed as you want, specify NODEBUG. DFSMSHsm will then process the data sets. You can use this gradual conversion procedure when you add more volumes to DFSMSHsm control.

DEFERMOUNT: Specifying defer mounting of tapes

The use of the SETSYS DEFERMOUNT parameter is considered to be nonfunctional and its use is ignored by DFSMSHsm.

DENSITY: Specifying the tape density

The use of the DENSITY parameter is considered to be nonfunctional and its use will result in an ARC1605I parse error.

DFHSMDATASETSERIALIZATION and USERDATASETSERIALIZATION: Specifying data set serialization

Explanation: **DFHSMDATASETSERIALIZATION** | **USERDATASETSERIALIZATION** are mutually exclusive, optional parameters that apply to volume migration and volume backup in a multiple-DFSMSHsm-host environment. These parameters specify whether data set serialization of system resources is provided for all data sets by system facilities.

DFHSMDATASETSERIALIZATION specifies that data set serialization of system resources during volume migration and volume backup in a multiple-image environment is not provided by system facilities, and that DFSMSHsm should implement its requirements for resource serialization.

USERDATASETSERIALIZATION specifies that data set serialization of system resources during volume migration and volume backup in a multiple-DFSMSHsm-host environment is provided by system facilities. Examples of such system facilities are Global Resource Serialization and JES3 data set reservation. In a single DFSMSHsm host, data set enqueue is provided. DFSMSHsm should optimize its requirements for resource serialization with those facilities.

Note:

1. The setting of this parameter affects the value of the DFSMSHsm integrity age. Refer to the “[Integrity age for space management](#)” on page 56 for more information on the relationship between integrity age and DFHSMDATASETSERIALIZATION | USERDATASETSERIALIZATION.
2. In DFSMSHsm Version 1 Release 5, the incremental backup function was restructured in order to improve the performance of that function. This improvement is only effective when USERDATASETSERIALIZATION is specified. Only use the SETSYS DFHSMDATASETSERIALIZATION command if it is required by your environment. Otherwise, it is recommended that you use the SETSYS USERDATASETSERIALIZATION command.
3. The fast subsequent migration function supports reconnection only in a USERDATASETSERIALIZATION environment.

4. Some data sets, such as multivolume physical sequential data sets, are processed only with SETSYS USERDATASETSERIALIZATION.

Attention

In a multiple-DFSMSHsm-host environment, do not specify USERDATASETSERIALIZATION unless you have a data set serialization facility installed and enabled on your system. Otherwise, serious data-integrity problems can occur.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is DFHSM DATASET SERIALIZATION.

DISASTERMODE: Specifying the tape operation mode for DFSMSHsm in disaster mode

Explanation: **DISASTERMODE** is an optional parameter that specifies whether DFSMSHsm establishes a tape substitution mode. This command plays an important part during an actual disaster recovery or when you are running a disaster recovery test.

While DFSMSHsm is running in disaster mode, recalls and recovers from tape volumes that are flagged as having disaster alternate volumes dynamically substitute the alternate tape for the original tape. For more information on the use of disaster alternate volumes, refer to [Chapter 9, “Making disaster backup copies of DFSMSHsm-owned tape volumes,”](#) on page 435.

If you specify Y for this parameter on the SETSYS command, DFSMSHsm checks the TTOC record of the original tape volume to see if a disaster alternate volume exists and if so, requests the use of it to return the data. If you specify N for this parameter on the SETSYS command or no disaster alternate volume exists, DFSMSHsm will request the original tape volume and makes no substitution.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you specify DISASTERMODE without specifying any subparameter, the default is Y.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is N.

DSBACKUP: Controlling the command data set backup environment

DSBACKUP is an optional parameter that allows you to control the command data set backup environment. You can balance the backup workload between DASD and tape tasks and you can tailor DFSMSHsm tape mounting and demounting for command data set backups. DFSMSHsm can run up to 64 concurrent command data set backup tasks.

If GENVSAMCOMPNames(YES) | GVCN(YES) is in effect, and *dsname* is a VSAM base cluster and *newdsname* is either migrated or uncataloged, DFSMSHsm will process the request, and default names will be assigned to the VSAM data and index components. If GENVSAMCOMPNames(NO)|GVCN(NO) is in effect, and *dsname* is a VSAM base cluster and *newdsname* is either migrated or uncataloged, the data set backup command will fail.

DASDSELECTIONSIZE (*maximum*, *standard*) helps you balance the workload between DASD and tape tasks for all WAIT-type requests that do not target tape. DASDSELECTIONSIZE is only applicable if both tape and ML1 DASD are allowed for command data set backups. *maximum* is the size, in kilobytes, of the largest data set, (a WAIT-type request) that is directed to ML1 DASD. *standard* is the largest size, in kilobytes, that is considered to be a small data set. DFSMSHsm directs small data sets (a WAIT-type request) to DASD if a tape task is not immediately available to process the request. DFSMSHsm directs

NOWAIT requests to tape, if tape tasks are allowed. The values that you can specify for *maximum* and *standard* range from 0 to 999999.

A data set is considered small if it is less than or equal to *standard*. A data set is considered large if it is greater than *maximum*. A data set with a value between *maximum* and *standard* is a medium data set.

To determine whether a data set uses tape or disk, refer to [Table 64 on page 1131](#).

Note: The default is DASDSELECTIONSIZE(3000,250), where 3000 is the *maximum* in kilobytes, and 250 is the *standard* in kilobytes. The categories in the table are based on these default values. If the you change the DASDSELECTIONSIZE(*maximum*,*standard*) settings, then these categories change.

Table 64. Determining data set task targets			
Default criteria	Large	Medium	Small
Data set size	3001 KB and up	251–3000 KB	0–250 KB
WAIT-type requests	Favor tape	First available	Favor disk
NOWAIT-type requests	Favor tape	Favor tape	Favor tape

The following list describes the task targets that are used in [Table 64 on page 1131](#).

This task target . . .	Means . . .
Favor disk	Small data sets that are processed by WAIT-type requests will select ML1 DASD as target. However, if you specify DASD(TASKS(0)), the data set will use tape.
Favor tape	Data set backups will be directed to tape unless you specify TAPE(TASKS(0)).
First available	Data sets will favor the first available task, regardless of disk or tape. If both tasks are available, it will favor the tape task.

Note:

1. If you want command data set backups to go to DASD only if targeted to do so, and all nontargeted command data set backups to go to tape, you can specify DASDSELECTIONSIZE(0 0). This forces all nontargeted command data set backups to go to tape.
2. If only one value is specified for DASDSELECTIONSIZE, it will be interpreted as the maximum size.

DASD(TASKS(nn)) specifies that you want to use ML1 DASD for command data set backups.

TAPE(TASKSnn) DEMOUNTDELAY(MAXIDLETASKS(drives) MINUTES(minutes)) specifies that you want to use tape for command data set backups.

You can also specify both DASD and tape for command data set backup tasks. TASKS(nn) specifies the maximum number of concurrent command data set backup tasks that DFSMSHsm directs to ML1 DASD or to tape. nn is a value between 0 and 64. If the number of DASD or tape tasks specified is 0, all command data set backup requests targeting the backup to DASD or tape fail. The sum of DASD(TASKS(nn)) and TAPE(TASKS(nn)) cannot exceed 64. If the sum is zero, any request fails.

DEMOUNTDELAY allows you to tailor the DFSMSHsm tape mounting and demounting for command data set backup. MINUTES(minutes) is the number of minutes that you want DFSMSHsm to wait before it deallocates the tape associated with continuously inactive (idle) command data set backup tasks. minutes is a value from 0 to 1440. A value of 1440 indicates that DFSMSHsm will not demount the tape until command data set backup tasks are held, a SWITCHTAPES event occurs, or DFSMSHsm shuts down, so this value extends across 24 hours.

If you specify DEMOUNTDELAY(MINUTES(0)), a tape that is mounted for command data set backup remains mounted long enough to support a continuous stream of WAIT-type backup requests to tape from a single job stream even if the queue of work momentarily becomes empty.

Note: The tape will remain mounted for up to 5 seconds if MINUTES(0) is specified.

A change to the DEMOUNTDELAY(MINUTES) value results in a new delay time being set. This new time is calculated from the current time of day. In other words, the new delay does not take into account any time the tape has been idle up to the issuance of this change.

MAXIDLETASKS(drives) is the maximum number of tape drives that DEMOUNTDELAY can accommodate. *drives* is a value from 0 to 64. You can specify a maximum of 64 drives for MAXIDLETASKS, regardless of the number of TAPE tasks specified. However, the effective number of tasks for MAXIDLETASKS is bound by the number of TAPE tasks. So, if you specify MAXIDLETASKS(64) and TAPE(TASKS(5)), the effective MAXIDLETASKS is 5. If at a later time you respecify TAPE(TASKS(7)), the effective MAXIDLETASKS becomes 7.

When you issue the QUERY SETSYS command, DFSMSHsm displays the number of MAXIDLETASKS specified. The QUERY ACTIVE command displays both the effective number of MAXIDLETASKS and the number of current idle tape tasks (that is, drives that are allocated but have no work).

When a command data set backup task writing to tape completes, DFSMSHsm does not deallocate tape units until all queued command data set backup requests have been processed. This enables all consecutive WAIT-type requests by the same batch job to be performed without going through a demount/mount sequence.

DEMOUNTDELAY notes:

1. DEMOUNTDELAY(MINUTES(0)) or MAXIDLETASKS(0) specifies that DFSMSHsm deallocates tapes when there are no requests on the queue that the tape task could select. The tape task can select any requests except those that are restricted to DASD.
2. If a tape drive is idle (waiting for additional backup requests) and is then used to satisfy a command data set backup request, DFSMSHsm resets the DEMOUNTDELAY time specified for that drive. In other words, the DEMOUNTDELAY is per instance, not an accumulation of idle time.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify the DASDSELECTIONSIZE parameter on any SETSYS command, the DFSMSHsm default for *maximum* is 3000 KB, and *standard* is 250 KB. If you do not specify the DASD(TASKS) or TAPE(TASKS) parameters on any SETSYS command, the DFSMSHsm default for *nn* is 2. If you do not specify DEMOUNTDELAY parameter on any SETSYS command, the DFSMSHsm default for MAXIDLETASKS is 0. If idle tape tasks exist and the MINUTES parameter was not specified, then the default for MINUTES is 60 minutes.

Note:

1. You can prevent DFSMSHsm from directing command data set backups to ML1 DASD by specifying DASD(TASKS(0)).
2. You can prevent DFSMSHsm from directing command data set backups to tape by specifying TAPE(TASKS(0)). You should do this if daily backup targets DASD in your installation.
3. You can prevent nontargeted command data set backups from going to ML1 DASD by specifying DASDSELECTIONSIZE(0).
4. A change in TAPE(TASKS(*nn*)) requires DFSMSHsm to start additional tape tasks or stop existing tape tasks (at the end of processing the current command data set).
5. If the sum of tape and DASD tasks exceed the high limit of 64, the command fails.
6. Use care in making a change in the DEMOUNTDELAY parameter. For instance, if a backup to tape task completes its work, the value for MAXIDLETASKS is greater than zero, and DEMOUNTDELAY(10) is in effect, DFSMSHsm sets an internal timer for 10 minutes.
 - If after 10 minutes, no additional work is received, the task begins to deallocate the drive and ends.
 - If during this 10 minute idle time the task is selected to perform a backup, the timer is canceled. Once the task is complete DFSMSHsm again sets a timer value of 10 minutes.

- If during this 10 minute delay, the time is decreased to 5 minutes by using DEMOUNTDELAY(MINUTES(5)), DFSMSHsm cancels the current timer and a new timer is established for 5 minutes (not taking into consideration any previous time).
 - If DEMOUNTDELAY(MINUTES(3)) is in effect and you issue an increase such as DEMOUNTDELAY(MINUTES(10)), DFSMSHsm cancels the current timer value and uses the new value. This resets the countdown to 10 minutes from the time the command is received.
7. If a parsing error occurs, DFSMSHsm fails the SETSYS command with the ARC1605I message. For nonparsing errors but contextual errors, DFSMSHsm fails the parameter in error but continues to process all other parameters in the SETSYS DSBACKUP command. DFSMSHsm treats the TAPE(TASKS) and DASD(TASKS) parameters as a single entity. For example, SETSYS DSBACKUP(DASDSELECTIONSIZE(5000 300) DASD(TASKS(45)) TAPE(TASKS(20))) results in DASDSELECTIONSIZE being set, but DASD(TASKS) and TAPE(TASKS) fail because the sum of both tasks exceeds 64.

DSSXMMODE: Specifying whether DFSMSdss will be loaded in its own address space using the cross memory interface, or in the DFSMSHsm address space, for backup, CDS backup, dump, migration, and full-volume and data set recovery

Explanation: DSSXMMODE is an optional parameter specifying whether DFSMSdss is loaded in its own address space, or in the DFSMSHsm address space, for backup, CDS backup, dump, migration, and full-volume and data set recovery functions.

This parameter can be issued from the ARCCMDxx PARMLIB member only and does not affect FRBACKUP and FRRECOV use of the DFSMSdss cross memory interface.

The following parameters are optional parameters of the DSSXMMODE parameter:

N

DFSMSdss will not be loaded into its own address space for the DFSMSHsm function being invoked. Instead, DFSMSdss will be loaded in the DFSMSHsm address space. This is the default.

Y

DFSMSdss will be loaded into a new address space for the DFSMSHsm function being invoked and will be retained for future invocations of the that function, until DFSMSHsm has been shutdown or restarted.

BACKUP

Controls where DFSMSdss for the DFSMSHsm backup function will be loaded; in its own address space (Y) or in the DFSMSHsm address space (N). The default is N (no).

CDSBACKUP

Controls where DFSMSdss for the DFSMSHsm CDS backup function will be loaded; in its own address space (Y) or in the DFSMSHsm address space (N). The default is N (no).

DUMP

Controls where DFSMSdss for the DFSMSHsm dump function will be loaded; in its own address space (Y) or in the DFSMSHsm address space (N). The default is N (no).

MIGRATION

Controls where DFSMSdss for the DFSMSHsm migration function will be loaded; in its own address space (Y) or in the DFSMSHsm address space (N). The default is N (no).

RECOVERY

Controls where DFSMSdss for the DFSMSHsm full-volume and data set recovery functions will be loaded; in their own address space (Y) or in the DFSMSHsm address space (N). The default is N (no).

Note:

1. When issuing the FRBACKUP or FRRECOV command to back up volumes to disk or to recover volumes and data sets from disk, DFSMSdss will always be loaded in its own address space. Note that FRBACKUP processing, automatic dump processing, and FRRECOV processing all invoke DFSMSdss

to dump from fast replication volumes to tape, or to restore from tape. DFSMSdss address space identifiers can be started optionally, based on the SETSYS DSSXMMODE command in the ARCCMDxx member of the SYS1.PARMLIB.

2. The DFSMSHsm recall function will always run with DFSMSdss loaded in the DFSMSHsm address space.
3. When transitioning or moving Db2 objects, DFSMSHsm will run with DFSMSdss loaded in the DFSMSHsm address space.
4. When migrating, dumping, or restoring data using direct to cloud connections, DFSMSdss will always be loaded in its own address space.

SMS relationship: None.

SETSYS default: If you specify the DSSXMMODE parameter without N (no), Y (yes), or a specific function, then the SETSYS default for this parameter and all unspecified functions is to run with DFSMSdss loaded in the DFSMSHsm address space.

DFSMSHsm default: If you do not specify DSSXMMODE on the SETSYS command during startup, the default is to load DFSMSdss in the DFSMSHsm address space.

DUMPIO: Specifying the DFSMSdss DASD I/O buffering technique

Explanation: **DUMPIO(*n,m*)** is an optional parameter identifying which DFSMSdss DASD I/O buffering technique to use and the number of tracks to read for each EXCP.

The *n* indicates the DFSMSdss DASD I/O buffering technique for reading tracks. This value applies to DFSMSdss physical volume dump, when using the BACKVOL command with the DUMP subparameter.

The *m* indicates the value used for DFSMSdss logical dump. If *m* is not specified, the value will be the same as *n*.

The values to be used for *n* and *m* and their meanings are:

Value

Meaning

- | | |
|----------|-------------------------------------|
| 1 | DFSMSdss reads 1 track at a time |
| 2 | DFSMSdss reads 2 tracks at a time |
| 3 | DFSMSdss reads 5 tracks at a time |
| 4 | DFSMSdss reads 1 cylinder at a time |

Note:

1. Exercise caution when using the DUMPIO parameter because it can have a severe effect on virtual and real storage requirements in a z/OS system. The higher the level of optimization used, the more real storage required for I/O buffers, which are page-fixed. This could result in a limit to the number of concurrent DFSMSHsm tasks, could require that DFSMSHsm address space be expanded, or could cause overall system degradation. Refer to [z/OS DFSMSHsm Implementation and Customization Guide](#) for information on how to calculate storage requirements.
2. DFSMSdss uses three I/O buffers for each active dump task. The size of each of these buffers is determined by the DASD device type and by how many tracks or cylinders are specified in the DUMPIO parameter. For example, if you specify SETSYS DUMPIO(4) and SETSYS MAXDUMPTASKS(15), DFSMSdss gets buffers for 3 cylinders of data for each dump task that is running. The total storage required, if you are using a 3380 DASD device, is 33MB of 31-bit storage. This means that the operating system region size must be larger than the default of 32MB, or an ABEND878 occurs.
3. One other performance consideration is that *each* page of buffer storage will be page-fixed, requiring time and machine cycles.

4. DUMPIO does not apply during aggregate backup and recovery, use the SETSYS ABARSOPTIMIZE parameter instead.
5. DUMPIO applies to migration and full volume dump to DIRECT cloud connections. It is ignored during other cloud operations.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you specify DUMPIO without a value for *m*, then *m* defaults to *n*.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default for *n* is one and the default for *m* is four.

DUPLEX: Allowing DFSMSHsm to create two tapes concurrently

Explanation: **DUPLEX** allows DFSMSHsm to create two tapes concurrently, the original tape intended to be kept onsite and the alternate tape taken offsite or written to a remote tape library.

Subparameter	Explanation
BACKUP(Y N)	<p>Duplex alternates will or will not be made for backup volumes. If this subparameter is not specified, no change will be made to the current setting for backup. Y specifies that duplexing will occur for backup tape volumes. N specifies that duplexing will not occur for backup tape volumes.</p> <p>If BACKUP(Y) is specified, then the ERRORALTERNATE subparameter can also be specified:</p> <p>BACKUP(Y ERRORALTERNATE(CONTINUE)) Specifies that if an error occurs on the alternate volume, then the processing will continue on the original tape. This is the default if neither CONTINUE nor MARKFULL are specified.</p> <p>Upon completion of the original tape, either a tapecopy or a recycle is immediately attempted, according to the following subparameter:</p> <p>BACKUP(Y ERRORALTERNATE(CONTINUE(TAPECOPY))) A TCN record for tapecopy is created and written to OCDS. The TCN record indicates that TAPECOPY should be used to create the duplex tape. This is the default if neither TAPECOPY nor RECYCLE are specified.</p> <p>BACKUP(Y ERRORALTERNATE(CONTINUE(RECYCLE))) A TCN record for recycle is created and written to OCDS. The TCN record indicates that the original volume should be recycled in order to create new original and duplex tapes.</p>

Subparameter	Explanation
MIGRATION(Y N)	<p>Duplex alternates will or will not be made for migration volumes. If this subparameter is not specified, no change will be made to the current setting for migration. Y specifies that duplexing will occur for migration tape volumes. N specifies that duplexing will not occur for migration tape volumes.</p> <p>If MIGRATION(Y) is specified, then the ERRORALTERNATE subparameter can also be specified:</p> <p>MIGRATION(Y ERRORALTERNATE(CONTINUE)) Specifies that if an error occurs on the alternate volume, then the processing will continue on the original tape. This is the default if neither CONTINUE nor MARKFULL are specified.</p> <p>Upon completion of the original tape, either a tapecopy or a recycle is immediately attempted, according to the following subparameter:</p> <p>MIGRATION(Y ERRORALTERNATE(CONTINUE(TAPECOPY))) A TCN record for tapecopy is created and written to OCDS. The TCN record indicates that TAPECOPY should be used to create the duplex tape. This is the default if neither TAPECOPY nor RECYCLE are specified.</p> <p>MIGRATION(Y ERRORALTERNATE(CONTINUE(RECYCLE))) A TCN record for recycle is created and written to OCDS. The TCN record indicates that the original volume should be recycled in order to create new original and duplex tapes.</p> <p>MIGRATION(Y ERRORALTERNATE(MARKFULL)) Specifies that both the original and alternate tapes will be marked full if an error occurs on the alternate tape and the failing data set will be retried on a new set of tapes.</p>

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: When either BACKUP or MIGRATION is specified with no subparameter, the default is Y.

DFSMSHsm default: If you do not specify DUPLEX and either of its subparameters, DUPLEXING does not occur.

Note: The SETSYS DUPLEX command is not saved across DFSMSHsm startups, therefore, it is recommended that you specify the SETSYS DUPLEX command in your DFSMSHsm startup member so that duplexing will be your normal environment. Although the SETSYS DUPLEX values can be changed while DFSMSHsm is up and running, the values will not take effect until after any currently running function has completed.

EMERGENCY and NOEMERGENCY: Specifying no data set movement

Explanation: **EMERGENCY** | **NOEMERGENCY** are mutually exclusive, optional parameters specifying whether emergency mode is in effect. Emergency mode stops all DFSMSHsm automatic and command data set processing in both SMS and non-SMS environments and in both primary and secondary address spaces, including functions like aggregate backup, aggregate recovery, audit, migration, backup, recall, recovery and recycle.

EMERGENCY specifies that emergency mode is in effect, and no new DFSMSHsm data set processing can be performed. DFSMSHsm allows any operation in process to finish the currently selected data set and it allows any currently running ABACKUP and ARECOVER commands to continue until completion. However, DFSMSHsm also allows commands that correct or set up options to run. For example, you can issue FIXCDS, QUERY, or SETSYS commands while running in EMERGENCY mode.

NOEMERGENCY specifies that emergency mode is not in effect.

Note:

1. If DFSMSHsm does not resume processing for all functions after you specify SETSYS EMERGENCY and then SETSYS NOEMERGENCY, you may need to issue a RELEASE command, assuming a HOLD command was issued previously.
2. You can start DFSMSHsm in emergency mode by specifying EMERG=YES on the START command.
3. Emergency mode does not prevent a host from placing recall requests in the common recall queue (CRQ), where those requests are then processed by other active hosts. To prevent a host from placing requests in the CRQ, specify HOLD COMMONQUEUE(RECALL(PLACEMENT)).
4. Emergency mode does not prevent a host from placing dump or recover requests onto a common queue, where those requests are then processed by other active hosts. It does prevent a host from accepting any new requests for processing from a common queue.
5. A SETSYS xxEMERGENCY command in the ARCCMDxx startup member would override any EMERG=xxx parameter.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is NOEMERGENCY.

ERASEONSCRATCH and NOERASEONSCRATCH: Specifying to check for erasure of DFSMSHsm-owned data sets

Explanation: **ERASEONSCRATCH** | **NOERASEONSCRATCH** are mutually exclusive, optional parameters specifying whether DFSMSHsm should check for erasure of DFSMSHsm-owned data sets that have migrated or been backed up to DFSMSHsm-owned volumes.

ERASEONSCRATCH specifies that DFSMSHsm asks RACF for the erase status of the user's data set when backup versions and migration copies are scratched from DFSMSHsm-owned DASD volumes. The data set is deleted, and if RACF indicates erase-on-scratch, the DASD residual data is overwritten by data management.

NOERASEONSCRATCH specifies that DFSMSHsm does not ask RACF for the erase status of the user's data set when backup versions and migration copies are scratched from DFSMSHsm-owned DASD volumes. The data set is deleted but the DASD residual data is not overwritten by data management.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is NOERASEONSCRATCH.

Note:

1. DFSMSHsm always allows data management to perform erase-on-scratch functions for user-owned data sets.
2. ERASEONSCRATCH and NOERASEONSCRATCH affect only backup versions and migration copies of data sets.

EVENTDRIVENMIGRATIONQUIESCE: Specifying quiesce window(s) for Interval and On-demand Migration

Explanation: **EVENTDRIVENMIGRATIONQUIESCE**(*hhmm1-hhmm2/hhmm3-hhmm4*) is an optional parameter specifying the time window(s) of when Interval and On-demand migration should quiesce. The times defined:

- *hhmm1* is the quiesce start time of the first window.
- *hhmm2* is the quiesce end time of the first window.
- *hhmm3* is the quiesce start time of the second window.
- *hhmm4* is the quiesce start time of the second window.

These times represent the window(s) that Interval and On-demand Migration should quiesce, or halt execution. If Interval or On-demand Migration is started outside of the specified window(s) it will be allowed to run until completion.

Note: Time is based on the 24-hour clock and is expressed as a four-digit decimal number in the format *hhmm*. The hour is represented by *hh* and the minutes by *mm*. For example, 1:15 p.m. is 1315 and midnight is 2400. The maximum value you can specify for *hh* is 24, the maximum value you can specify for *mm* is 59, and the maximum value you can specify for *hhmm* is 2400.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None. If specified, at least the first window must contain time values.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the default is no Event Driven Migration Quiesce (for example, Interval or On-demand Migration will run as normal throughout the day).

Note:

1. To reset the time windows, specify all *hhmm*(s) in the first window only as 0000 (for example, 'EVENTDRIVENMIGRATIONQUIESCE(0000 0000)'). Specifying any other combination of 0000 will result in a parse error and neither of the time windows being reset simultaneously. Individual windows cannot be reset while the other is left unchanged. If you desire to keep one window, you must reset as described earlier, and reissue the command with the desired time window.
2. A single time window may span midnight, but not both. This can be specified by the end time < start time (for example, 'EVENTDRIVENMIGRATIONQUIESCE(1315 04340)').
3. Time windows cannot overlap (for example, 'EVENTDRIVENMIGRATIONQUIESCE(1315 0430 0400 0800)').
4. Start and end times within a window, and within a set of windows, cannot be equal to the exception of specifying 0000 (as noted in the first note) in the first time window.

Note: It is imperative that the issuer of this command understand that quiescing Interval and On-demand Migration may result in DASD being filled to capacity because DFSMSHsm is not processing inactive data.

EXITOFF: Specifying inactive installation exits

Explanation: **EXITOFF**(*modname,modname, ...*) is an optional parameter specifying installation exits inactive in DFSMSHsm primary address space. EXITOFF now applies to ABARS exits. For *modname*, substitute the module name of the installation exit you want to be inactive. See [z/OS DFSMS Installation Exits](#) for more information about installation exits.

Valid installation exit module names and their meanings are as follows:

Module Name	Abbreviation	Meaning
ARCADEXT	AD	Data set deletion exit
ARCBDEXT	BD	Data set backup exit

Module Name	Abbreviation	Meaning
ARCBEXT	BE	Aggregate Backup error exit
ARCCBEXT	CB	Control data set backup exit
ARCCDEXT	CD	Data set reblock exit
ARCCREXT	CR	Aggregate Recovery conflict resolution exit
ARCEDEXT	ED	ABARS expiration date setting exit
ARCINEXT	IN	Initialization exit
ARCMDEXT	MD	Data set migration exit
ARCMMEXT	MM	Second level migration data set exit
ARCMVEXT	MV	Space management volume exit
ARCM2EXT	M2	Aggregate Backup ML2 bypass exit
ARCRDEXT	RD	Data set recall exit
ARCRPEXT	RP	Return-priority exit
ARCSAEXT	SA	Space management and backup exit
ARCSDEXT	SD	Shutdown exit
ARCSKEXT	SK	Aggregate Recovery bypass exit
ARCTDEXT	TD	Tape data set exit
ARCTEEXT	TE	Tape ejected exit
ARCTVEXT	TV	Tape volume exit

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMShsm default: If you do not specify an installation exit *modname* on any SETSYS command, the DFSMShsm default is that installation exits will remain in their current state from a previous SETSYS command.

Note:

1. You can delete installation exits while DFSMShsm is running. If you want to delete installation exits, you must use the EXITOFF parameter to specify them.
2. All existing installation exits are supported on the EXITOFF parameter.
3. The *modnames* are not positional. Installation exits not specified remain in their current state.
4. If the EXITS parameter and either the EXITOFF or the EXITON parameter both specify the same installation exit on a SETSYS command, the status specified with the EXITOFF or EXITON parameter has precedence over the status specified with the EXITS parameter.
5. If the same installation exit is specified in both the EXITOFF and the EXITON parameters on a SETSYS command, DFSMShsm leaves that installation exit inactive.
6. The recall exit (ARCRDEXT) and data set deletion exit (ARCADEXT) will not be invoked for SMS-managed data sets.

EXITON: Specifying active installation exits

Explanation: **EXITON(modname,modname, ...)** is an optional parameter specifying active installation exits in DFSMShsm primary address space. EXITON now applies to ABARS exits. For *modname*, substitute the module name of the installation exit you want to be active.

The ARCINEXT exit is passed parameters having 31-bit addresses. This exit may require modification to accept and use the 31-bit addresses. See *z/OS DFSMS Installation Exits* for more information about installation exits.

Valid installation exit module names and their meanings are as follows:

Module Name	Abbreviation	Meaning
ARCADEXT	AD	Data set deletion exit
ARCBDEXT	BD	Data set backup exit
ARCBEEEXT	BE	Aggregate Backup error exit
ARCCBEXT	CB	Control data set backup exit
ARCCDEXT	CD	Data set reblock exit
ARCCREXT	CR	Aggregate Recovery conflict resolution exit
ARCEDEXT	ED	ABARS expiration date setting exit
ARCINEXT	IN	Initialization exit
ARCMDEXT	MD	Data set migration exit
ARCMMEXT	MM	Second level migration data set exit
ARCMVEXT	MV	Space management volume exit
ARCM2EXT	M2	Aggregate Backup ML2 bypass exit
ARCRDEXT	RD	Data set recall exit
ARCRPEXT	RP	Return-priority exit
ARCSAEXT	SA	Space management and backup exit
ARCSDEXT	SD	Shutdown exit
ARCSKEXT	SK	Aggregate Recovery bypass exit
ARCTDEXT	TD	Tape data set exit
ARCTEEXT	TE	Tape ejected exit
ARCTVEXT	TV	Tape volume exit

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify an installation exit *modname* on any SETSYS command, the DFSMSHsm default is that no installation exits are active.

Note:

1. You can dynamically load installation exits while DFSMSHsm is running. If you want to load installation exits, you must specify the exits to be loaded with the EXITON parameter.
2. All existing installation exits are supported on the EXITON parameter.
3. The *modnames* are not positional. Installation exits not specified remain in their current state.
4. If the EXITS parameter and either the EXITOFF or the EXITON parameter both specify the same installation exits on the same SETSYS command, the status specified with the EXITOFF or EXITON parameter has precedence over the status specified with the EXITS parameter.
5. If the same installation exit is specified in both the EXITOFF and the EXITON parameters on a SETSYS command, DFSMSHsm leaves that installation exit inactive.
6. If an installation exit is specified with EXITON and that exit is already active, DFSMSHsm does not load a new copy of the exit.

7. The recall exit (ARCRDTEXT) and data set deletion exit (ARCADEXT) are not invoked for SMS-managed data sets.

EXITS: Specifying the installation exits to be taken

The use of the SETSYS EXITS parameter has been replaced by the SETSYS EXITON and SETSYS EXITOFF parameters. The EXITS parameter is still supported but is no longer documented.

EXPIREDDATASETS: Specifying whether to scratch data sets with expired expiration dates

Explanation: **EXPIREDDATASETS(SCRATCH | NOSCRATCH)** is an optional parameter specifying whether or not DFSMSHsm should scratch data sets having explicit expiration dates when the date has passed (data set has expired). SMS-managed data sets not having explicit expiration dates are expired when they have met their management class expiration criteria.

Note: Explicit expiration dates can be set by JCL, utility parameters, Data Class Expiration parameters, or Data Class Retention parameters. The explicit expiration date for a data set can be found in the data set's VTOC entry.

SCRATCH specifies that DFSMSHsm scratch data sets that have an expired expiration date when it performs space management and migration cleanup.

NOSCRATCH specifies that DFSMSHsm ignore the expiration date. DFSMSHsm processes the data set as if the expiration date has not been reached.

SMS relationship: Parameter applies in some instances to SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is not to scratch the expired data sets.

EXTENDEDTTOC: Specifying whether to use extended tape table of contents (TTOC)

Explanation: **EXTENDEDTTOC(Y|N)** is an optional parameter specifying whether DFSMSHsm should use extended tape table of contents (TTOCs) in order to better utilize tapes with higher capacity volumes in your installation. Extended TTOCs allow DFSMSHsm to write slightly more than one million data sets (potentially) to a migration tape or backup tape.

To use extended TTOCs in your installation, you must define your offline control data set(OCDS) with a maximum record size of 6144 bytes, as described in *z/OS DFSMSHsm Implementation and Customization Guide*. Then enable the support by entering the SETSYS command with optional parameter EXTENDEDTTOC(Y) or its shortened form, EXTTC(Y).

Note: If there is an attempt to specify SETSYS EXTENDEDTTOC(Y) and the OCDS has not been defined with a record size of 6144, DFSMSHsm will issue an error message and force the value of EXTENDEDTTOC to "N".

To return to using non-extended TTOCs, enter the SETSYS command with EXTENDEDTTOC(N) or its shortened form, EXTTC(N).

Note: Users should not specify the SETSYS EXTENDEDTTOC(Y) command on any one host in an HSMplex until the OCDS has been redefined with a record length of 6144 bytes and all hosts in an HSMplex are prepared to issue the SETSYS EXTENDEDTTOC(Y) command.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is not to use extended TTOCs.

FASTREPLICATION: Specifying fast replication methods

Explanation

FASTREPLICATION is an optional parameter that specifies which copy method to use for data set level recovery and other fast replication options.

The following parameters are optional parameters of the FASTREPLICATION parameter:

DATASETRECOVERY

DSR

Fast replication should be used for data set level recovery.

NONE

Fast replication is not used. Only traditional copy methods are to be used. This is the default.

PREFERRED

Fast replication should be used as the copy method for data-set-level recovery whenever possible. For situations in which fast replication cannot be used to recover a data set, recovery is done through traditional copy methods.

REQUIRED

Fast replication must always be used. If fast replication cannot be used, the recovery request fails.

FCRELATION

FCREL

Specifies whether an extent-level or a full-volume-level FlashCopy relationship is to be established for fast replication backup and recovery of volumes.

EXTENT

Extent-level FlashCopy relationships are established during backup and recovery. Free space is not flashed. This is the default.

FULL

Full-volume FlashCopy relationships are established during volume backup and recovery. The entire volume including free space is copied. Therefore, the background copy might take longer to complete.

MESSAGEDATASET

MSGDS

Specifies that messages for each fast replication command (or auto dump of a copy pool) should be directed to a unique fast replication message data set.

NO

A fast replication message data set is not used.

YES

Messages are directed to a fast replication message data set.

HLQ(*hlq*)

Overrides the default high level qualifier of HSMMSG that will be used for a fast replication message data set. The *hlq* value can be a maximum six characters in length, and must meet all other requirements for data set name high level qualifiers.

VOLUMEPAIRMESSAGES

VPM

Controlling FlashCopy volume pairing debug message ARC1809I.

YES

FlashCopy volume pairing debug message ARC1809I is enabled.

NO

FlashCopy volume pairing debug message ARC1809I is disabled.

SMS relationship

Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default

If you specify:

- DATASETRECOVERY without a parameter, then the SETSYS default is to use traditional copy methods. Fast replication is not used (NONE).
- FCRELATION without a parameter, then the SETSYS default is to establish extent-level FlashCopy relationships (EXTENT).
- MESSAGEDATASET without a parameter, then the SETSYS default is to direct messages to a fast replication message data set (YES).

DFSMSHsm default

If you do not specify:

- DATASETRECOVERY on a SETSYS command or FASTREPLICATION on a FRRECOV DSNAMES, the default is to use traditional copy methods. Fast replication is not used (NONE).
- FCRELATION on the SETSYS command, the default is to establish extent-level FlashCopy relationships (EXTENT).
- MESSAGEDATASET on the SETSYS command, the default is to *not* direct messages to a fast replication message data set (NO).
- VOLUMEPAIRMESSAGES on the SETSYS command, the default is to disable ARC1809I messages (NO).

Usage notes:

1. The FASTREPLICATION value specified on the FRRECOV DSNAMES command overrides the SETSYS value.
2. The SETSYS FASTREPLICATION command is applicable for recovery from DASD only. If the recovery is from tape, the DATASETRECOVERY parameter is ignored.
3. DATASETRECOVERY(NONE) cannot be used in combination with FlashCopy targets that are Metro Mirror primary volumes. That is, when the ALLOWPPRCP parameter is specified with: YES, PRESERVEMIRRORNO, PRESERVEMIRRORPREFERRED, or PRESERVEMIRRORREQUIRED.
4. Data set recovery from DASD is not permitted if NOCOPY is used to create the backup copy. The recovery cannot be performed until a dump copy has been created.
5. The combination of the FROMDASD and FASTREPLICATION(DATASETRECOVERY(REQUIRED)) restricts data set recovery to being performed only when fast replication can be used.
6. Specifying FCRELATION does not affect data set recovery.
7. Specifying FCRELATION(EXTENT) to establish extent-level FlashCopy relationships is ignored during FRBACKUP and FRRECOV operations when the copy pool is defined to use a function that requires full-volume FlashCopy (such as fast reverse restore or incremental).

FREQUENCY: Specifying the number of days between backup versions of a data set

Explanation

FREQUENCY is an optional parameter specifying the number of days that must have elapsed since the last time DFSMSHsm backed up a data set before DFSMSHsm will back up the data set again.

The following parameter is an optional parameter of the FREQUENCY parameter:

days

A decimal number from 0 to 999. A value of 0 allows DFSMSHsm to back up data sets every time volume backup runs unless you use the FREQUENCY parameter of the ALTERDS or BACKVOL command to change the frequency.

SMS relationship

Parameter does not apply to SMS-managed DASD volumes or data sets.

SETSYS default

None.

DFSMSHsm default

If you do not specify FREQUENCY parameter on any SETSYS command, the DFSMSHsm default is zero.

Usage notes:

1. The FREQUENCY parameter only applies to data sets that have been backed up.

INCREMENTALBACKUP: Specifying whether to back up only changed data sets during incremental volume backup

Explanation: **INCREMENTALBACKUP** is an optional parameter specifying whether DFSMSHsm should back up only changed data sets during incremental volume backup.

Subparameter	Explanation
CHANGEDONLY	<p>INCREMENTALBACKUP(CHANGEDONLY) specifies that DFSMSHsm backs up the following data sets only if their change bit is on in the data set VTOC entry:</p> <ul style="list-style-type: none"> • Non-VSAM data sets • VSAM data sets cataloged in an integrated catalog facility catalog.
ORIGINAL	<p>INCREMENTALBACKUP(ORIGINAL) specifies that DFSMSHsm creates an initial backup version (if one does not exist) for all non-VSAM and integrated catalog facility VSAM data sets on a primary volume when incremental backup takes place, regardless of the setting of the change bit in the data set VTOC entry.</p>

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you do not specify either subparameter on any SETSYS command, the SETSYS default is ORIGINAL.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is ORIGINAL.

Note:

1. VSAM data sets not cataloged in an integrated catalog facility Catalog are backed up during incremental backup by comparing the date of the last update from the catalog record to the date of the last backup from the BCDS.
2. It is suggested that you consider running the INCREMENTALBACKUP command with the ORIGINAL parameter once a month, once a quarter, or when some of the following scenarios are suspected because a backup version might not always exist for all data sets that do not have the change flag on. The scenarios are:

- DFSMSdss or a similar product backed up the volume and reset the change flags before DFSMSHsm was able to back up the data sets.
 - Users may have deleted the backup versions that DFSMSHsm created and, the data sets had not been changed again.
3. The INCREMENTALBACKUP parameter affects only data sets residing on a primary volume when incremental backup is performed. Even if you have specified ORIGINAL, no backup copy will be created for a migrated data set for which the change bit in the data set VTOC entry is off. These data sets have to be backed up individually using the BACKDS command.

INPUTTAPEALLOCATION: Specifying whether to wait for the input tape to be allocated

Explanation: **INPUTTAPEALLOCATION(WAIT | NOWAIT)** is an optional parameter specifying whether DFSMSHsm should wait until the tape unit is allocated for the input tape for recall, recovery, or restore.

Subparameter	Explanation
WAIT	<p>INPUTTAPEALLOCATION(WAIT) specifies that DFSMSHsm waits until the input tape unit is allocated. All DFSMSHsm functions stop until this allocation request is satisfied.</p> <p>When using the WAIT option, caution must be exercised to limit the number of DFSMSHsm tasks that require tape. If more tasks run than there are available tape devices, a deadlock condition will occur because of the exclusive enqueue that MVS allocation will put on the task input/output table (SYSZTIOT) while awaiting the tape device. This condition can be resolved only if another job (not DFSMSHsm) releases a tape device, or if you cancel DFSMSHsm.</p>
NOWAIT	<p>INPUTTAPEALLOCATION(NOWAIT) specifies that DFSMSHsm does not wait until the input tape unit is allocated. Instead, DFSMSHsm reissues the request every 10 seconds up to six times. If the input tape unit is still not allocated after seven tries, DFSMSHsm asks the operator whether DFSMSHsm should cancel the request or repeat the allocation sequence. With the NOWAIT option, dynamic allocation issues message IEF238D. Once answered, other DFSMSHsm tasks can proceed with allocations.</p>

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is NOWAIT.

INTERVALMIGRATION and NOINTERVALMIGRATION: Specifying whether interval migration is performed

Explanation: INTERVALMIGRATION and NOINTERVALMIGRATION are mutually exclusive, optional parameters specifying whether DFSMSHsm should perform interval migration. The purpose of interval migration is to prevent the DFSMSHsm-managed volumes from running out of space during the day. DFSMSHsm normally does a space check every hour. Therefore, if you request interval migration, DFSMSHsm uses this hourly space check to determine on which volumes interval migration is done.

INTERVALMIGRATION specifies that interval migration is performed.

NOINTERVALMIGRATION specifies that interval migration is not performed.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is INTERVALMIGRATION.

Note:

1. If you want the operator to decide whether interval migration should be allowed after DFSMSHsm decides it is necessary, specify the INTERVALMIGRATION parameter and the REQUEST parameter. Interval migration does not occur during the DFSMSHsm startup process.
2. With Tape Mount Management (TMM), interval migration can still be performed on SMS volumes in a storage group having the attribute of AUTO MIGRATE=I, regardless of the setting for the SETSYS INTERVALMIGRATION parameter or the SETSYS NOINTERVALMIGRATION parameter.
3. Only primary space management will be performed on SMS volumes in a storage group having the attribute of AM=P. When this attribute is specified, interval migration is not performed even if the SETSYS INTERVALMIGRATION parameter is specified.

ISSUEARC1902: Enable/disable the issue of ARC1902I message to the console

Explanation:ISSUEARC1902(YES|NO DELAYTIME(secs)) is an optional parameter that controls whether DFSMSHsm will issue message ARC1902I when an auto backup function is still running after its specified quiesce/end time. If enabled, ARC1902I will be displayed on the operator console after the user specified amount of time, in seconds, after the quiesce/end time of an autobackup function. The ARC1902I message will be moved from the operator console if the affected automatic function completer normally.

The Sub-parameters of the ISSUEARC1902 parameterSet follow this discussion.

Subparameter	Explanation
Y N	Enable/disable the DFSMSHsm to issuing ARC1902I message to the console.
DELAYTIME(SECS)	Amount of time (seconds) after the quiesce/end time need to display ARC1902I message on the console when autobackup is still running.

SMS relationship:Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default:The following table describes the SETSYS ISSUEARC1902 default.

If.....	then.....
If you specify ISSUEARC1902.	Issue of ARC1902I message is enabled with default delaytime 7200 secs
If you specify DELAYTIME without a value.	SETSYS default is 7200 secs (equivalent to 2 hrs).
If you specify DELAYTIME with a value greater than 50400 secs (equivalent to 14 hrs.)	SETSYS default is 50400 secs (equivalent to 14 hours.)

DFSMSHsm default: The following table describes the SETSYS ISSUEARC1902 DFSMSHsm default.

If.....	then.....
If you specify ISSUEARC1902.	Issue of ARC1902I message is enabled with default delaytime 7200 secs

If.....	then.....
If you specify DELAYTIME without a value.	DFSMSHsm default is 7200 secs (equivalent to 2 hrs).
If you specify DELAYTIME with a value greater than 50400 secs (equivalent to 14 hrs.)	N/A

JES2 and JES3: Specifying the job entry subsystem

Explanation: JES2 and JES3 are mutually exclusive, optional parameters specifying the job entry subsystem that is used with DFSMSHsm.

If you want to use JES3, you must specify the JES3 parameter before you specify the first ADDVOL command. If you do not, DFSMSHsm defaults to JES2. When DFSMSHsm is started in an operating system that has JES3 and when you do not specify JES3 during DFSMSHsm initialization, an error message is written when DFSMSHsm receives the first request from the JES3 converter (interpreter). If you specify JES3, but the operating system uses JES2, DFSMSHsm is not notified of the error. DFSMSHsm, however, uses the rules that govern pool configuration for JES3 and one or both of the following situations can occur:

- Some ADDVOL, SETSYS, and DEFINE commands fail if they are issued when it is unacceptable in a JES3 system.
- Volumes eligible for recall in a JES2 system might not qualify for the DFSMSHsm general pool and, in some cases, are not available for recall in the JES3 system.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is JES2.

JOURNAL and NOJOURNAL: Specifying whether to write updated control data set records in the journal data set

Explanation: JOURNAL and NOJOURNAL are mutually exclusive, optional parameters specifying whether MCDS data set records, BCDS data set records, and OCDS data set records are written in the journal data set when DFSMSHsm updates the control data sets. If the MCDS, BCDS, or OCDS is lost or damaged, you can use the journal data set with a backup copy of the control data set to recreate the affected control data set.

JOURNAL specifies that DFSMSHsm write the BCDS, MCDS, and OCDS data set records in the journal data set when DFSMSHsm updates them.

Subparameter	Explanation
SPEED	<p>JOURNAL(SPEED) specifies that the module making the change should wait only until the journaling request has been added to the journaling queue. When you use SPEED, DFSMSHsm does not wait long, because of the following reasons:</p> <ul style="list-style-type: none"> • The journal task has the highest dispatching priority among all DFSMSHsm tasks. • The journal entries are not blocked. • In each DFSMSHsm host, the journal entries are written in the order they are created. <p>The task probably does not lose more than one entry if the system fails, unless you allocate the journal data set on a volume that is used often.</p>
RECOVERY	<p>JOURNAL(RECOVERY) specifies that the module making the change should wait until the journal entry has been written in the journal data set. DFSMSHsm writes each record as it receives the record. Therefore, not more than one entry can be lost if DFSMSHsm abnormally ends. After DFSMSHsm has updated the affected control data set and has written each journal data set entry, DFSMSHsm continues processing.</p>

NOJOURNAL specifies that DFSMSHsm is not to write the updated BCDS, MCDS, and OCDS records in the journal data set.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you specify JOURNAL without a subparameter, the SETSYS default is RECOVERY.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is JOURNAL(RECOVERY).

Note:

1. The SPEED option queues the journal updates to be written as soon as possible, but the requesting task does not wait for the journal entry to be written. This can cause ABEND878, ABEND80A, or ABEND106-C during heavy journaling activity, if the logging task cannot keep up with the journaling load. This can happen if the ARCILOG task gets tied up during I/O to the DASD data set. If DFSMSHsm should be abnormally terminated while running this option, all journal updates not yet written will be lost.
2. If JOURNAL(RECOVERY) is specified and DFSMSHsm senses it is waiting too long for the journal entry to be written, DFSMSHsm stops its wait and leaves the request on the journaling queue. This temporary return to SPEED mode helps prevent lockouts caused by resource contention.
3. DFSMSHsm does not write updated entries from the BCDS, MCDS, and OCDS unless you include a JCL DD statement for the journal data set in the DFSMSHsm startup procedure.
4. DFSMSHsm nulls the journal data set every time DFSMSHsm successfully backs up the control data sets.
5. When defined as a large format data set, the DFSMSHsm journal can exceed 65,535 tracks per volume. For information about defining the DFSMSHsm journal, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

MAXABARSADDRESSSPACE: Specifying the maximum number of concurrent secondary address spaces supported

Explanation: **MAXABARSADDRESSSPACE(nn)** is an optional parameter specifying the maximum number of concurrent aggregate backup and recovery address spaces that DFSMSHsm supports. For *nn*,

substitute a decimal number from 1 to 64 to represent the number of secondary address spaces that can run concurrently for aggregate backup and recovery.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is one.

MAXBACKUPTASKS: Specifying the maximum number of concurrent volume backup tasks

Explanation: **MAXBACKUPTASKS(*nn*)** is an optional parameter specifying the maximum number of volume backup tasks DFSMSHsm can concurrently process. For *nn*, substitute a decimal number from 1 to 15 to represent the number of volume backup tasks to be run concurrently.

To decide how many concurrent volume backup tasks DFSMSHsm should run, consider how many tape units are available if you use tape backup volumes and if you are utilizing duplex tape. Also, consider the system work load and how many backup volumes you have available when volume backup is running. Because each volume backup task chooses a unique daily backup volume, it is impractical to specify more volume backup tasks than available daily backup volumes, because a volume backup task waits until a daily backup volume is available.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is two tasks.

MAXCOPYPOOLTASKS: Specifying the maximum number of concurrent DFSMSdss invocations for each fast replication command

Explanation: **MAXCOPYPOOLTASKS** is an optional parameter that you use to define the maximum number of DFSMSdss invocations for each fast replication command that DFSMSHsm can concurrently process. **FRBACKUP tasks:** The maximum number of concurrent fast replication backup volume tasks is the product of the FRBACKUP and DSS parameters. **FRRECOV tasks:** The maximum number of concurrent fast replication recover volume tasks is the product of the FRRECOV and DSS parameters.

Note: When the FRBACKUP COPYPOOL command uses FlashCopy consistency groups, DFSMSHsm processes all the copy pool volumes in one DFSMSdss invocation, regardless of the SETSYS MAXCOPYPOOLTASKS(FRBACKUP(*nn*) DSS(*nnn*)) setting.

Subparameter	Explanation
FRBACKUP	MAXCOPYPOOLTASKS (FRBACKUP(<i>nn</i>)) is an optional parameter that you specify to define the maximum number of DFSMSdss invocations for each FRBACKUP command that DFSMSHsm can concurrently process. For <i>nn</i> , substitute a decimal number from 1 to 64 to represent the number of DFSMSdss invocations for each FRBACKUP command that runs concurrently.
FRRECOV	MAXCOPYPOOLTASKS (FRRECOV(<i>nn</i>)) is an optional parameter that you specify to define the maximum number of DFSMSdss invocations for each FRRECOV command that DFSMSHsm can concurrently process. For <i>nn</i> , substitute a decimal number from 1 to 64 to represent the number of DFSMSdss invocations for each FRRECOV command that runs concurrently.

Subparameter	Explanation
DSS	MAXCOPYPOOLTASKS (DSS(<i>nnn</i>)) is an optional parameter that you specify to define the maximum number of volume pairs that DFSMSdss passes to each DFSMSdss invocation for backup and recover functions. For <i>nnn</i> , substitute a decimal number from 1 to 254 to represent the number volume pairs that DFSMSdss passes to each DFSMSdss invocation.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMShsm default: If you do not specify this parameter on any SETSYS command, the DFSMShsm default for the FRBACKUP parameter is 15 tasks, the FRRECOV parameter is 15 tasks, and the DSS parameter is 24 tasks.

Attention

The default values recommended for these parameters are based on performance tests that determined the values that would provide the optimal throughput. You should realize that increasing the value of the DSS(*nnn*) parameter above the recommended default may have a negative impact on throughput.

MAXDSRECOVERTASKS: Specifying the maximum number of data set recovery tasks from DASD and tape

Explanation: **MAXDSRECOVERTASKS(*nn*)** is an optional parameter specifying the maximum number of data set recovery tasks DFSMShsm can concurrently process for both DASD and tape. This parameter allows an installation to determine the appropriate tasking level to recover individual data sets, based on the amount of disk storage and the number of tape drive resources available. The actual number of effective tasks may be limited if the number of backup resources is less than the total tasks specified. For *nn*, substitute a decimal number from 1 to 64 to represent the number of data set recovery tasks to be run concurrently.

The value you specify for MAXDSRECOVERTASKS applies to data set recovery tasks for both DASD and tape. To limit tape requests to a subset of this value, use the keyword MAXDSTAPERECOVERTASKS instead. The maximum number of tasks from DASD is the value of MAXDSRECOVERTASKS minus the current number of tape tasks specified by MAXDSTAPERECOVERTASKS.

If there are no active tape tasks, the number of disk tasks can reach the value specified by MAXDSRECOVERTASKS. To prevent tape tasks, specify SETSYS MAXDSTAPERECOVERTASKS(0).

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMShsm default: If you do not specify this parameter on any SETSYS command, the DFSMShsm default is two tasks.

MAXDSTAPERECOVERTASKS: Specifying the maximum number of data set recovery tasks from tape

Explanation: **MAXDSTAPERECOVERTASKS(*nn*)** is an optional parameter specifying the maximum number of data set recovery tasks DFSMShsm can concurrently process from tape. This parameter allows you to determine the appropriate tasking level to recover individual data sets, based on the number of tape drives available. The actual number of effective tasks might be limited if the number of backup tape cartridges is fewer than the total tasks specified.

For *nn*, specify a decimal number from 0 to 64 to represent the number of data set recovery tasks to be run concurrently. The value for *nn* cannot exceed the value you specify for the MAXDSRECOVERTASKS parameter.

To prevent any DFSMSHsm data set recoveries from tape, specify SETSYS MAXDSTAPERECOVERTASKS(0).

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is the value of MAXDSRECOVERTASKS.

MAXDUMPRECOVERTASKS: Specifying the maximum number of concurrent volume recovery from dump tasks

Explanation: MAXDUMPRECOVERTASKS(*nn*) is an optional parameter specifying the maximum number of volume recovery from dump tasks DFSMSHsm can concurrently process. This parameter allows you to set the appropriate tasking level to recover individual volumes, based on the number of tape drives available. The actual number of effective tasks will be limited by the number of tape drives available and dump stacking value.

For *nn*, specify a decimal number from 0 to 64 to represent the number of volume recovery from dump tasks to be processed concurrently.

Note:

1. Special care must be taken when considering dump classes and their dump stacking values. A higher dump stacking value requires fewer tapes; however, the higher the number of volumes dumped onto a single tape, the greater the likelihood that a single tape will be needed for multiple recovery tasks. If the specified number of concurrent tasks is greater than the number of tape drives in use, the maximum number of concurrent tasks will be limited.
2. For CVQ processing consider using SETSYS MAXDUMPRECOVERTASKS(0) from the master scheduler, instead of a HOLD RECOVER or FRRECOV, when you want the master scheduler to assign CVQ work, but not use the master scheduler resources for CVQ fast replication or for volume recovery processing. To end this state, use a SETSYS MAXDUMPRECOVERTASKS command with a non zero value.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed volumes.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the default is 1.

MAXDUMPTASKS: Specifying the maximum number of concurrent volume dump tasks

Explanation: MAXDUMPTASKS(*nn*) is an optional parameter specifying the maximum number of volume dump tasks DFSMSHsm can concurrently process. For *nn*, substitute a decimal number from 0 to 64 to represent the number of dump tasks to be run concurrently.

Because multiple classes can be requested for each input volume, MAXDUMPTASKS does not explicitly limit the number of output copies. To decide how many concurrent volume dump tasks DFSMSHsm should run, consider how many tape units are available. Multiple dump classes for an input volume require a separate tape drive for a tape in each class. To determine the required number of tape drives for the volume dump process, multiply the number of classes per volume by the number of dump tasks (*nn*). This number should not exceed the number of tape drives expected to be available during the volume dump process.

MAXDUMPTASKS(0) allows hosts to submit work but the dump tasks available on those hosts are not selected for processing. In a common dump queue, dump tasks available from other hosts in the common

queue are used. To end this state, use a SETSYS MAXDUMPTASKS command with a non zero value, and a RELEASE DUMP command.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is two tasks.

Note:

1. When specifying more than a few tasks, ensure that DFSMSdss is loaded in its own address space by specifying SETSYS DSSXMMODE(DUMP(Y)) in the ARCCMDxx PARMLIB member. Failure to do so may result in 878 abends and related errors.

MAXEXTENTS: Specifying the number of extents to cause extent reduction

Explanation: **MAXEXTENTS(*extents*)** is an optional parameter specifying the number of extents that, if equaled or exceeded, causes DFSMSHsm to reallocate all non-VSAM data sets. For *extents*, substitute a decimal number from 0 to 16 specifying the number of extents to allow before DFSMSHsm does extent reduction. If you specify 0, DFSMSHsm does not do extent reduction.

When volume migration occurs for volumes containing a non-VSAM data set that cannot migrate either because the data set was referred to recently or has reached or exceeded its maximum extents, DFSMSHsm migrates and then immediately recalls the data set.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you specify a value larger than 16, the SETSYS default for *extents* is 16.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 0.

Note:

1. The MAXEXTENTS parameter does not apply to VSAM data sets or to data set organizations not supported by DFSMSHsm.
2. You use this parameter for active, non-VSAM data sets. This parameter only applies to those non-VSAM data sets that are architecturally limited to 16 extents. It does not apply to data sets that are allocated to more than one volume, striped data sets, extended format data sets, PDSE or HFS data sets. Data sets that can be allocated to multiple volumes via the DATACLASS, but are only allocated on one volume, are eligible for EXTENT REDUCTION.
3. DFSMSHsm will not do extent reduction unless the data set is more than one day old in a single-DFSMSHsm-host environment or two days old in a multiple-DFSMSHsm-host environment.
4. Using a low MAXEXTENTS value might cause extent reduction to occur more frequently than you want. If you specify a MAXEXTENTS value, try using a value of 5 or more.

MAXINTERVALTASKS: Specifying the maximum number of automatic interval migration or on-demand migration tasks that can run concurrently

Explanation: **MAXINTERVALTASKS(*nn*)** is an optional parameter specifying the maximum number of automatic interval or event-driven on-demand migration tasks that can run concurrently. It applies to the processing of primary volumes and SMS-managed volumes.

For *nn*, substitute a decimal number from 0 to 15, specifying the maximum number of concurrent tasks. A zero value allows you to restrict the interval migration or on-demand migration task from running on any host that has MAXINTERVALTASKS set to zero. Once MAXINTERVALTASKS is set to a nonzero value, interval migration or on-demand migration processing will resume on the host.

Note: MAXEVENTDRIVENMIGTASKS or MAXEDMT alias can be used instead of MAXINTERVALTASKS parameter.

Tape units must be available to support the maximum number of migration tasks. In a direct-to-tape environment, the number of tape units available must be at least equal to *nn*. The number of SDSP data sets available should be greater than *nn*. In addition, in a duplex tape environment, the number of tape units must be at least 2 times *nn*.

In an environment where data sets can be routed to either DASD or tape, a tape unit is required for each migration task that encounters a data set that is migrated directly to tape. Make sure that you have *nn* tape units available. Once DFSMSHsm has selected a tape unit, that unit will not be released until the automatic volume space management function has completed, or until recall needs the tape. In a duplex tape environment, DFSMSHsm selects two tape units for each migration task and these tape units will not be released until the automatic volume space management function is complete.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS defaults: None.

DFSMSHsm defaults: If you do not specify this parameter on any SETSYS command, the default value for the maximum number of concurrent interval migration or on-demand migration tasks (*nn*) is equal to the value defined for MAXMIGRATIONTASKS. The default value for MAXMIGRATIONTASKS is 2.

MAXMIGRATIONTASKS: Specifying the maximum number of automatic volume space management tasks that can run concurrently

Explanation: MAXMIGRATIONTASKS(*nn*) is an optional parameter specifying the maximum number of automatic volume space management (primary space management, interval migration and on-demand migration) tasks that can run concurrently. It applies to the processing of primary volumes and SMS-managed volumes. You can specify the maximum number of concurrent tasks (*nn*) from 1 to 15.

MAXMIGRATIONTASKS applies only to automatic functions. You can run only one command migration task at a time. Individual commands for data set or volume migrations do not run concurrently.

Tape units must be available to support the maximum number of migration tasks. In a direct-to-tape environment, the number of tape units available must be at least equal to *nn*. The number of SDSP data sets available should be greater than *nn*. In addition, in a duplex tape environment, the number of tape units must be at least 2 times *nn*.

In the environment where data sets can be routed to either DASD or tape, a tape unit is required for each migration task that encounters a data set to be migrated directly to tape. Make sure that you have *nn* tape units available. Once DFSMSHsm has selected a tape unit, it will not be released until the automatic volume space management function has completed. In a duplex tape environment, DFSMSHsm selects two tape units for each migration task and these tape units will not be released until the automatic volume space management function is complete. This does not apply to DASD volumes.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the default value for the maximum number of concurrent migration tasks (*nn*) is 2.

MAXRECALLTASKS: Specifying the maximum number of active recall tasks

Explanation: MAXRECALLTASKS(*nn*) is an optional parameter specifying the number of active recall tasks DFSMSHsm can concurrently process. For *nn*, substitute a decimal number from 1 to 15 for the number of concurrent recall tasks DFSMSHsm can process.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you specify a value larger than 15, the SETSYS default for *nn* is 15.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default for *nn* is 15.

Note: The MAXRECALLTASKS parameter is utilized in the processing of deletes initiated by secondary space management.

MAXRECYCLETASKS: Specifying the maximum number of active recycle tasks

Explanation: **MAXRECYCLETASKS(*nn*)** is an optional parameter specifying the maximum number of tape processing tasks initiated by a single RECYCLE request. For *nn*, substitute a decimal number from 1 to 15 for the number of tasks to initiate. The number of tape processing tasks can be changed dynamically, even while DFSMSHsm is processing a RECYCLE request. This allows you to free tape drives for uses other than recycle processing, should the need arise. Each task that moves data requires two tape drives, one for input and one for output. In a duplex tape environment, each task moving data requires three tape drives, one for input and two for output.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default for *nn* is 2.

Note: Only one RECYCLE command is processed at a time when you use the VOLUME parameter of the specific RECYCLE command. An MVS image can process only one RECYCLE command at a time. If you issue a generic RECYCLE command, and multiple volumes are eligible, then MAXRECYCLETASKS is used.

MAXSSMTASKS: Specifying the maximum number of concurrent automatic secondary space management tasks

MAXSSMTASKS(CLEANUP(*nn*) | TAPEMOVEMENT(*mm*) | EMPTYCONTAINERDELETION(*kk*)) is used to define the maximum number of secondary space management tasks that can be run concurrently.

Subparameter	Explanation
CLEANUP(<i>nn</i>)	MAXSSMTASKS(CLEANUP(<i>nn</i>)) specifies the maximum number of secondary space management migration cleanup tasks that can run concurrently. For <i>nn</i> , substitute a decimal number from 0 to 15 for the maximum number of concurrent migration cleanup tasks. A value of 0 indicates that there are no cleanup tasks, and therefore the secondary space management window includes no cleanup processing.
TAPEMOVEMENT(<i>mm</i>)	MAXSSMTASKS(TAPEMOVEMENT(<i>mm</i>)) specifies the maximum number of automatic secondary space management tape migration tasks that can be run concurrently. This parameter applies to the migration of data sets from level 1 volumes to level 2 tape volumes. If you are duplexing migration tapes, each TAPEMOVEMENT task you define will require two tape drives. For <i>mm</i> , substitute a decimal number from 0 to 15 for the maximum number of concurrent tape migration tasks. A value of 0 indicates that there are no tape movement tasks. In that case, DFSMSHsm performs no ML1-to-ML2 data movement. This setting is useful when no tape drives are available.

Subparameter	Explanation
EMPTYCONTAINERDELETION(<i>kk</i>)	MAXSSMTASKS(EMPTYCONTAINERDELETION(<i>kk</i>)) specifies the number of Cloud Storage containers processing tasks. Only 0 or 1 values are acceptable. A value of 0 indicates that Cloud Storage containers deletion task will not be submitted. Alias: ECD

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify the CLEANUP parameter on any SETSYS command, the DFSMSHsm default for *nn* is 2. If you do not specify the TAPEMOVEMENT parameter on any SETSYS command, the DFSMSHsm default for *mm* is 1. If you do not specify the EMPTYCONTAINERDELETION parameter on any SETSYS command, the DFSMSHsm default for *kk* is 1.

MIGDENSITY: Specifying the tape density when allocating scratch tapes for migration

The use of the MIGDENSITY parameter is considered to be nonfunctional, and its use results in an ARC1605I parse error.

MIGRATEPREFIX: Specifying the prefix for the migration copy name

Explanation: **MIGRATEPREFIX(*prefix*)** is an optional parameter specifying the prefix (high-level qualifier) of the data set name that DFSMSHsm generates when it migrates the data set. *prefix* must be defined as 1 to 7 alphanumeric characters; the first character must be alphabetic only.



Attention: Changing this parameter after the DFSMSHsm environment is set could result in failures during the recall and recovery of data sets from tape and SDSPs.

The migration copy name has the following format:

```
prefix.HMIG.Tssmmhh.user1.user2.Xydd
```

If patch-enabled, the following dataset name will be generated:

```
prefix.HMIG.Tcccchh.user1.user2.Xydd
```

where:

- *prefix* is replaced with the prefix you specify with this command.
- HMIG indicates that this is a migrated data set.
- Tssmmhh is the time when DFSMSHsm migrated the data set; *ss* is the second, *mm* is the minute, and *hh* is the hour. If a duplicate exists after this name is generated, DFSMSHsm changes the first character of the time stamp.
- Tcccchh is the time when DFSMSHsm migrated the data set; *cccc* is the time in hundredths of seconds from the beginning of the hour converted to four alphabetic characters and *hh* is hours. For more information, see “Names of migrated data sets” on page 73.
- *user1* and *user2* are replaced with the first two qualifiers of the data set name. The data set name can be 44 characters long.

- *Xydd* is the date when DFSMSHsm migrated the data set. DFSMSHsm replaces the X with a letter that represents the decade. A–J have the following meaning:

```
A-1,  B-2,  C-3,  D-4,  E-5
F-6,  G-7,  H-8,  I-9,  J-0
```

For example, you specify MIGRATEPREFIX(HSMmig). Your data set name is TERRY.CLIST.TEXT. DFSMSHsm migrates the data set on 15 July 1990 at 5:33 p.m. The migration copy name is:

```
HSMmig.HMIG.T003317.TERRY.CLIST.I0198
```

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is the UID you specified in the DFSMSHsm startup procedure. This UID is used as the first qualifier for data set names generated by DFSMSHsm during the BACKUP or MIGRATION functions. Changing this UID after the DFSMSHsm environment is set could result in failures during the recall and recovery of data sets from tape and SDSPs. [Chapter 2, “Space management of SMS-managed storage,” on page 23](#) explains the UID and the DFSMSHsm startup procedure.

MIGRATIONAUTOCLOUD: Allow the host to execute Cloud migration

MIGRATIONAUTOCLOUD ({ALL | CLOUDONLY | NOCLOUD} SELECTIONTHRESHOLD(*nnn*)) defines the possible target tier during the MIGRATE STORAGEGROUP command and automatic migration and the method of CEC thread serialization.

Optional subparameters of the SETSYS MIGRATIONAUTOCLOUD command:

ALL

Allows data set migration to ML1, ML2, transition, and migration to Cloud storage.

CLOUDONLY

Allows data set migration to Cloud storage only.

NOCLOUD

Allows data set migration to ML1, ML2, and transition only. NOCLOUD is the default.

Alias: MAC

SELECTIONTHRESHOLD(*nnn*)

Specifies the expected maximum percentage of data sets on a volume eligible to be migrated to the cloud before DFSMSHsm reserves a data movement thread in the disk storage controller unique to this volume. This parameter only applies when the ALL optional subparameter is specified. It does not apply when the NOCLOUD or CLOUDONLY options are specified.

A finite number of data movement threads is available in the disk controller to move data sets to cloud object storage. This threshold can be tuned to optimize when DFSMSHsm reserves a thread during volume selection based on the number of data sets expected to migrate to cloud object storage from a volume.

Each time a volume is processed, DFSMSHsm records the percentage of data sets that were migrated to cloud object storage. This percentage is used during the subsequent processing of the volume to determine whether a storage controller thread for cloud data movement should be reserved. If the percentage of data sets eligible for cloud migrations on the volume is less than this SETSYS value, then a thread is not reserved during volume selection. If the percentage is greater than the threshold, then a thread is reserved.

Reserving a disk controller data movement thread ensures that the DFSMSHsm volume task can select data sets eligible to migrate to the cloud and immediately process them. If a thread is not reserved, then data sets eligible to migrate to the cloud might have to wait if there are not enough available threads for all of the cloud migration activity. Also, if a DFSMSHsm volume task has reserved a thread

but is not currently using it, then that thread is not available to be used by other DFSMSHsm migration tasks.

Valid values are 0 to 100.

- 0 indicates that if any data set on a volume has migrated to the cloud, then a disk controller data movement thread should always be reserved. Use this value only if it is certain that all migrations will target cloud storage.
- 100 indicates that a data movement thread should never be reserved. This value should only be used if it is certain that migrations will never target cloud storage. In this case, the NOCLOUD parameter should be specified instead.

The default value is 40. This default value will likely need to be tuned based on the activity in your environment.

Using the SETSYS MIGRATIONAUTOCLOUD command without the SELECTIONTHRESHOLD parameter does not change this value.

Alias: ST

Table 65. Summary of DFSMSHsm default values	
DFSMSHsm default value	Result
MIGRATEPSMEARLY(NO)	The actual data set age is used during primary space management.
MIGRATIONAUTOCLOUD(NOCLOUD)	Only data sets that would target ML1 or ML2, or transition will be processed. Any eligible data sets that would target cloud storage will not be migrated by this host during the automatic migration.
MIGRATIONAUTOCLOUD(SELECTIONTHRESHOLD(40))	If the percentage of data sets migrated from a volume to the cloud is not greater than 40, this volume is selected without checking for "free" CEC threads. Otherwise, the volume is selected if a "FREE CEC" thread can be found to process the migrations.

MIGRATIONCLEANUPDAYS: Specifying the number of days to keep MCDS data set records and statistics records

Explanation: **MIGRATIONCLEANUPDAYS(recalldays statdays reconnectdays clouddays)** is an optional parameter specifying the number of days that DFSMSHsm keeps MCDS records for recalled data sets or keeps daily and volume statistics records before it deletes them during migration cleanup. The REPORT command uses the daily and volume statistics records. The IDCAMS DCOLLECT function gathers them when the CAPPLANDATA keyword is specified.

For *recalldays*, substitute a decimal number from 0 to 999 for the number of days since the date of recall that DFSMSHsm keeps MCDS data set records for recalled data sets.

For *statdays*, substitute a decimal number from 1 to 999 for the number of days DFSMSHsm keeps the daily and volume statistics records.

For *reconnectdays*, substitute a decimal number from 0 to 999 for the number of days that DFSMSHsm uses to calculate how long to keep the MCDS data set records for recalled data sets that are candidates for reconnection.

For *clouddays*, substitute a decimal number from 0 to 999 for the number of days that DFSMSHsm should keep old cloud migration copies around after a modified version of the same data set has been remigrated to the cloud.

This algorithm shows the calculation. The predicted remigration date is the date of recall plus the number of days of nonuse when the data set last migrated. For example:

- Date last referenced at time of migration: 99.100

- Date of migration: 99.110
- Date of recall: 99.125

This equals a predicted remigration date of 99.135. The 10 days the data set went unreferenced prior to the migration is added to the recall date to calculate the predicted remigration date. To this predicted remigration date, add the value specified for the *reconnectdays* parameter (in our example, we are using a value of 5). The MCDS data set record is retained through 99.140.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm defaults for *recalldays* is 10, for *statdays* is 30, for *reconnectdays* is 3, and for *clouddays* is 0.

Note:

1. The MCDS record for a reconnectable data set is not deleted until both the *recalldays* and *reconnectdays* criteria are met.
2. The *recalldays* value specifies to DFSMSHsm how long to keep MCDS data set records for recalled data sets that were not compacted or were compacted and met or exceeded the current value of SETSYS COMPACTPERCENT. For data sets that were compacted but did not meet the current value of SETSYS COMPACTPERCENT, the MCDS data set records are retained for 90 days.
3. Specifying a value other than 0, for *clouddays*, will cause DFSMSHsm to create 'Retained' MCA records describing old cloud migrated data sets, which will be kept for the number of days specified, once a newer version of the data set is remigrated to the cloud. Specifying a value of 0 will cause DFSMSHsm to not keep 'Retained' MCA records and the old cloud migrated data sets they describe. As a side effect of specifying 0, after having a non-zero value for a time, all 'Retained MCA' records and the old cloud migration copies they describe will be deleted on the next run of Secondary Space Management.

MIGRATIONLEVEL1DAYS: Specifying the inactive age for data sets migrating from level 1 volumes

Explanation: **MIGRATIONLEVEL1DAYS(days)** is an optional parameter specifying the number of contiguous days a data set must remain unreferenced before the data set is eligible for migration from a level 1 volume to a level 2 volume. For *days*, substitute a decimal number from 0 to 999. This value includes the time the unreferenced data set was on a primary volume. For example, if you specified MIGRATIONLEVEL1DAYS(15) and the minimum migration age is 5, the unreferenced data set remains on the primary volume for 5 days and on the migration level 1 volume for 10 days before DFSMSHsm migrates it to a migration level 2 volume.

SMS relationship: Parameter does not apply to SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter with any SETSYS command, the DFSMSHsm default is 60 calendar days.

Note: MIGRATIONLEVEL1DAYS applies only to migration from level 1 volumes to level 2 volumes.

MIGRATEPSMEARLY: Activate the PSM earlier checking option

MIGRATEPSMEARLY (YES | NO) activates the earlier checking option provided by primary space management (PSM). One is added to the age check, up until midnight.

Required subparameters of the SETSYS MIGRATEPSMEARLY command:

YES

One is added to the age check, up until midnight.

NO

The actual data set age is always used.

Alias: MPE

The default is NO.

MIGRATIONSUBTASKS: Enabling multiple migration subtasks

Explanation: **MIGRATIONSUBTASKS(NO | YES ADDITIONALSUBTASKS(nn))** is an optional parameter that allows DFSMSHsm to run multiple subtasks concurrently. Each subtask manages the migration for a single data set, and these run under migration tasks for primary space management, on-demand migration, and interval migration on level 0 volumes that migrate data sets to ML1 or ML2 volumes. Note that, when MIGRATIONSUBTASKS(YES) is specified, this will apply to *all* migration tasks running in DFSMSHsm.

The ADDITIONALSUBTASKS(nn) subparameter allows you to dynamically change the number of additional subtasks that the system can use, running under each migration task. These additional subtasks add to the number of subtasks that the system already uses when the MIGRATIONSUBTASKS parameter is specified. Note that the actual number of total subtasks used can vary. In general, the total migration subtasks used will be lower if a large value is specified with the maximum migration tasks (MAXMIGRATIONTASKS) parameter. Conversely, the total migration subtasks used will be higher, up to 15, if a smaller value is specified with the maximum migration tasks (MAXMIGRATIONTASKS) parameter.

Enablement occurs at DFSMSHsm startup, with SETSYS MIGRATIONSUBTASKS(YES) specified (or defaulted to YES if SETSYS MIGRATIONSUBTASKS is specified without the YES/NO keyword). Disablement occurs at DFSMSHsm startup with SETSYS MIGRATIONSUBTASKS(NO) specified, or if the command is not specified at all. Additional subtasks can be added or removed dynamically with the ADDITIONALSUBTASKS(nn) subparameter, with no DFSMSHsm startup necessary.

You may find that adding additional subtasks is advantageous, based on different possible factors in your system environment. If tape units limit the maximum number of migration tasks, for instance, then you may find that specifying additional migration subtasks increases the amount of concurrency. As another example, you may find that if many data sets being migrated are small, then additional migration subtasks will provide a improvement in overall throughput.

You must weigh such gains, however, against the possibility that specifying additional subtasks may increase the amount of below the line storage used by DFSMSHsm.

Note that when specifying MIGRATIONSUBTASKS(YES), USERDATASETSERIALIZATION must also be specified. Specifically, SETSYS USERDATASETSERIALIZATION must be specified in the ARCCMDxx parmlib member, prior to the SETSYS MIGRATIONSUBTASKS(YES) command.

Note, also, that when MIGRATIONSUBTASKS(YES) is specified, concurrent cell pool activity may increase in cell pool 5. It is therefore recommended to review the value specified by the CELLS parameter in the DFHSM startup procedure for CELL POOL 5. See "Specifying the size of cell pools" and "Adjusting the size of cell pools" in the *z/OS DFSMSHsm Implementation and Customization Guide* for more information.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is to not use migration subtasks. When MIGRATIONSUBTASKS is specified without the YES or NO keyword, the default is YES. When the ADDITIONALSUBTASKS option is not specified, DFSMSHsm will not use any additional subtasks.

MIGUNITNAME: Specifying the type of unit for allocating a scratch tape during migration

Explanation: **MIGUNITNAME(unitname)** is an optional parameter identifying the type of unit that should be specified the first time DFSMSHsm requests to allocate a scratch tape during migration. You can request several unit names for allocating the scratch tape. For *unitname* specify either 3480, 3480X, 3490, 3590-1, or an esoteric name which you specify in the USERUNITTABLE parameter of the SETSYS

command. If an esoteric tape unit name is used, the SETSYS USERUNITTABLE command must be issued prior to the SETSYS MIGUNITNAME command.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 3590-1.

Note:

1. You cannot specify an esoteric unit name that represents DASD; it must represent tape.
2. DFSMSHsm does not use MIGUNITNAME when it requests that a scratch volume be allocated while continuing from another volume. Instead, DFSMSHsm uses the same unit the volume was allocated on.
3. If you specify MIGDENSITY and MIGUNITNAME, the density must match the density capabilities for that type of unit. If you specify MIGUNITNAME and do not specify MIGDENSITY, DFSMSHsm uses the highest density for the specified unit.
4. If you specify an esoteric unit name with the MIGUNITNAME parameter, you must also have identified the esoteric unit name to DFSMSHsm with the USERUNITTABLE parameter.
5. You can specify the USERUNITTABLE parameter when you specify the MIGUNITNAME parameter or you could have specified USERUNITTABLE with a previous SETSYS command during this startup.

ML1OVERFLOW: Manage ML1 OVERFLOW volumes

Explanation: **ML1OVERFLOW** is an optional parameter specifying:

- Minimum data set size for ML1 OVERFLOW volume preference
- Threshold for ML1 OVERFLOW volume capacity for automatic secondary space management migration from ML1 OVERFLOW to ML2 volumes

ML1OVERFLOW has the following subparameters:

Subparameter	Explanation
DATASSETSIZE(<i>dssize</i>)	<p>Specifies the minimum size, in K bytes, of the data set for which an ML1 OVERFLOW volume is preferred for migration or backup. This includes invoking inline backup, the HBACKDS or BACKDS commands, or the ARCHBACK macro.</p> <p>SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.</p> <p>SETSYS default: 2000000 K bytes.</p> <p>DFSMSHsm default: If you do not specify this parameter on any SETSYS command, DFSMSHsm will use the value of 2000000 K bytes.</p> <p>If the calculated size of a data set being backed up or migrated is equal to or greater than <i>dssize</i>, DFSMSHsm prefers the OVERFLOW ML1 volume with the least amount of free space that still has enough remaining space for the data set.</p> <p>If the calculated size of the data set is less than the minimum size specified in <i>dssize</i>, DFSMSHsm prefers the NOOVERFLOW ML1 volume with the maximum amount of free space and least number of users.</p>

Subparameter	Explanation
THRESHOLD(<i>threshold</i>)	<p>Specifies the percent of occupied space in the ML1 OVERFLOW volume before migration of data sets to ML2 volumes during automatic secondary space management. DFSMSHsm calculates the percentage of occupied space for each ML1 OVERFLOW volume and then compares the average value with the specified threshold. If the average of the ML1 OVERFLOW volume pool has an occupancy that is equal to or greater than the threshold, then secondary space management is started and moves data from ML1 to ML2.</p> <p>SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.</p> <p>SETSYS default: 80%.</p> <p>DFSMSHsm default: If you do not specify this parameter on any SETSYS command, DFSMSHsm will use the value of 80%.</p>

ML2PARTIALSNOTASSOCIATEDGOAL: Specifying the maximum number of allowed partial ML2 tapes

Explanation: **ML2PARTIALSNOTASSOCIATEDGOAL(*nnn* | NOLIMIT)** is an optional parameter specifying the maximum number of partial ML2 tapes not associated with any migration or recycle task to remain after the generic RECYCLE command is invoked to recycle ML2 tape volumes. A partial tape is one that is not full and not empty. For *nnn*, substitute a decimal number from 0 to 999.

If the number of unassociated ML2 partial tapes is less than or equal to *nnn* when recycle is invoked, no partial tapes are recycled. To recycle all unassociated ML2 partial tapes, use the value of zero (0) for *nnn*. To prevent recycling of all ML2 partial tapes, specify the NOLIMIT subparameter.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, DFSMSHsm will use the value of 10.

Note:

1. This parameter is effective only during recycle processing
2. The allowed number of partial tape volumes applies to your complex of hosts which share the MCDS and OCDS and which define those volumes to DFSMSHsm.

ML2RECYCLEPERCENT: Specifying the maximum percentage of valid data on a migration tape

Explanation: **ML2RECYCLEPERCENT(*pct*)** is an optional parameter specifying the percent-valid criterion DFSMSHsm uses to recycle tape migration level 2 volumes, if you do not specify the PERCENTVALID parameter of the RECYCLE command. For *pct*, substitute a decimal number from 0 to 100. When DFSMSHsm invalidates a data set entry for a full volume and the percentage of valid data on the migration level 2 tape volume becomes less than or equal to *pct*, DFSMSHsm writes the informational message, ARC0365I, in the command activity log.

For a discussion of how DFSMSHsm calculates percentages based on valid blocks, refer to [“Consolidating valid data on tape volumes”](#) on page 537.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, DFSMSHsm will use the value specified with the SETSYS RECYCLEPERCENT parameter.

Note: DFSMSHsm ignores the ML2RECYCLEPERCENT parameter of this command and the PERCENTVALID parameter of the RECYCLE command when you use the RECYCLE command to recycle specific tape migration level 2 volumes.

MONITOR: Specifying which informational messages to print at the system console

Explanation: **MONITOR** is an optional parameter that specifies which informational messages DFSMSHsm is to print at the system console, and the values DFSMSHsm is to use for monitoring space in the journal and control data sets.

The subparameters of the MONITOR parameter are listed in alphabetical order and can be found at the following locations:

Subparameter	See
BACKUPCONTROLDATASET	“MONITOR (BACKUPCONTROLDATASET): Monitoring the space used in the BCDS” on page 1163
JOURNAL	“MONITOR (JOURNAL): Monitoring the space used in the journal data set” on page 1163
MIGRATIONCONTROLDATASET	“MONITOR (MIGRATIONCONTROLDATASET): Monitoring the space used in the MCDS” on page 1163
OFFLINECONTROLDATASET	“MONITOR (OFFLINECONTROLDATASET): Monitoring the space used in the OCDS” on page 1164
SPACE	“MONITOR (SPACE and NOSPACE): Printing volume space-use messages” on page 1164
STARTUP	“MONITOR (STARTUP and NOSTARTUP): Printing DFSMSHsm startup informational messages” on page 1164
VOLUME	“MONITOR (VOLUME and NOVOLUME): Print data set messages during volume processing” on page 1165

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 80%. No messages are printed to the console about space used during volume processing. No messages are printed to the console from DFSMSHsm commands read from the ARCCMDxx parmlib member during DFSMSHsm initialization.

Note: If you specify MONITOR, the specified level of activity logging controls which messages are written to the system console. For example, if you specify ACTLOGMSGLVL(EXCEPTIONONLY), DFSMSHsm does not write the successful execution messages to the system console.

MONITOR (BACKUPCONTROLDATASET): Monitoring the space used in the BCDS

Explanation: **BACKUPCONTROLDATASET(thresh)** is an optional subparameter of the MONITOR parameter that specifies when DFSMSHsm issues warning messages about the amount of space used in the BCDS.

For *thresh*, substitute a decimal number from 0 to 100 that specifies a percentage value. When the occupied space in the BCDS exceeds this percentage value, DFSMSHsm warns the system operator that the data set is getting full.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify a threshold on any SETSYS command, the DFSMSHsm default is 80%.

MONITOR (JOURNAL): Monitoring the space used in the journal data set

Explanation: **JOURNAL(thresh)** is an optional subparameter of the MONITOR parameter that specifies when DFSMSHsm issues warning messages about the amount of space used in the journal data set.

For *thresh*, substitute a decimal number from 0 to 100 that specifies a percentage value. When the occupied space in the journal data set exceeds this percentage value, DFSMSHsm warns the system operator that the data set is getting full.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this subparameter on any SETSYS command, the DFSMSHsm default is 80%.

Note:

When defined as a large format data set, the DFSMSHsm journal can exceed 65,535 tracks per volume. For information about defining the DFSMSHsm journal, see [z/OS DFSMSHsm Implementation and Customization Guide](#).

MONITOR (MIGRATIONCONTROLDATASET): Monitoring the space used in the MCDS

Explanation: **MIGRATIONCONTROLDATASET(thresh)** is an optional subparameter of the MONITOR parameter that specifies when DFSMSHsm issues warning messages about the amount of space used in the MCDS.

For *thresh*, substitute a decimal number from 0 to 100 that specifies a percentage value. When the occupied space in the MCDS exceeds this percentage value, DFSMSHsm warns the system operator that the data set is getting full.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify a threshold on any SETSYS command, the DFSMSHsm default is 80%.

MONITOR (OFFLINECONTROLDATASET): Monitoring the space used in the OCDS

Explanation: **OFFLINECONTROLDATASET(thresh)** is an optional subparameter of the MONITOR parameter that specifies when DFSMSHsm issues warning messages about the amount of space used in the OCDS.

For *thresh*, substitute a decimal number from 0 to 100 that specifies a percentage value. When the occupied space in the OCDS exceeds this percentage value, DFSMSHsm warns the system operator that the data set is getting full.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify a threshold on any SETSYS command, the DFSMSHsm default is 80%.

MONITOR (SPACE and NOSPACE): Printing volume space-use messages

Explanation

SPACE and NOSPACE are mutually exclusive, optional parameters of the MONITOR parameter that specify whether to print volume space-use messages when DFSMSHsm performs a space check for interval migration, on-demand migration, or primary space management. The three messages printed are: ARC0400I, ARC0401I, and ARC0402I.

SPACE

Volume space-use messages are printed to the system console, the DFSMSHsm log, and the appropriate activity log.

NOSPACE

Volume space-use messages printed to the DFSMSHsm log and the appropriate activity log only.

SMS relationship

The SPACE and NOSPACE parameters have the same meaning when applied to SMS-managed or non-SMS-managed volumes or data sets.

SETSYS default

None.

DFSMSHsm default

If you do not specify the SPACE or NOSPACE parameter on any SETSYS MONITOR command, the DFSMSHsm default is NOSPACE.

MONITOR (STARTUP and NOSTARTUP): Printing DFSMSHsm startup informational messages

Explanation: **STARTUP | NOSTARTUP** are mutually exclusive, optional subparameters of the MONITOR parameter that specify whether or not to print DFSMSHsm startup informational messages at the system console. The messages result from DFSMSHsm commands that are read from the PARMLIB member during DFSMSHsm initialization. You could also specify the STARTUP option on the EXEC PARM options of the DFSMSHsm startup procedure to direct the startup informational messages to the system console.

Subparameters	Explanation
STARTUP	MONITOR(STARTUP) specifies that DFSMSHsm startup informational messages print at the system console and in the DFSMSHsm log.
NOSTARTUP	MONITOR(NOSTARTUP) specifies that DFSMSHsm startup informational messages print only in the DFSMSHsm log, with the exception of messages from the PATCH command, which is printed at the system console.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is NOSTARTUP.

MONITOR (VOLUME and NOVOLUME): Print data set messages during volume processing

Explanation: **VOLUME** | **NOVOLUME** are mutually exclusive, optional subparameters of the MONITOR parameter that specify whether or not to print data set (ARC0734I) messages at the system console.

Subparameter	Explanation
VOLUME	MONITOR(VOLUME) specifies that messages print at the system console, DFSMSHsm log, and appropriate activity log. DFSMSHsm prints messages about data sets on volumes processed by space management, backup, and recovery.
NOVOLUME	MONITOR(NOVOLUME) specifies that messages relating to the data sets on the volumes processed by space management, backup, and recovery print only in the DFSMSHsm log and the appropriate activity log.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is NOVOLUME.

MOUNTWAITTIME: Specifying the time DFSMSHsm waits for a tape mount and open

Explanation: **MOUNTWAITTIME(*minutes*)** is an optional parameter specifying the time, in minutes, that DFSMSHsm waits for the tape volume to be mounted and opened. For *minutes*, substitute a decimal number from 1 to 120 to represent how long DFSMSHsm waits for a tape to be mounted and opened. If the first time period expires, DFSMSHsm sends message ARC0310A to the operator asking whether to mount the tape volume. If the operator answers Y to message ARC0310A, DFSMSHsm resets the timer to *minutes*. If the input volume has not been mounted and opened when the second time period expires, DFSMSHsm automatically ends the task. If the output volume has not been mounted and opened when the second time period expires, DFSMSHsm marks this volume as unavailable and selects another tape volume.

If you have not mounted an output tape dump volume when the second time period expires, DFSMSHsm automatically ends the dump task.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 15 minutes.

Note:

1. This parameter does not apply to tape mounts performed during aggregate backup and recovery processing.
2. DFSMSHsm sets a mount wait timer for both manual and automatic tape libraries. Appropriate consideration should be given based on which kind of mount DFSMSHsm will be involved with. Unique considerations may exist when using a virtual tape system (VTS).

OBJECTNAMES: Specifying the compaction-control qualifier of the object data set

Explanation: **OBJECTNAMES**(*name1,name2,...*) is an optional parameter specifying the compaction-control qualifier of all data sets that should be compacted with the compaction table for object data sets. For generation data group data sets, this compaction-control qualifier is the next-to-the-last qualifier of the data set name. For all other data sets, this compaction-control qualifier is the last qualifier of the data set name.

For *name1,name2,...*, specify the compaction-control qualifiers of the data set names of those data sets you want to compact with the compaction table for data sets that contain object code. For example, you could specify LINKLIB as a compaction-control qualifier.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is that the general compaction table is used to compact the object data sets.

ONDEMANDMIGRATION: Specifying whether on-demand migration should be performed

Explanation: **ONDEMANDMIGRATION** is an optional parameter specifying whether on-demand migration should be performed on SMS-managed volumes that have exceeded the high threshold and that belong to a storage group defined using AM=Y (Auto Migrate, yes). This parameter can be abbreviated as ODM.

The following parameters are optional parameters of the ONDEMANDMIGRATION parameter:

N

(no) On-demand migration is not performed.

Y

(yes) On-demand migration is performed on eligible SMS-managed volumes.

For more information about on-demand migration, see [“Using on-demand migration” on page 27](#).

SMS relationship: The ONDEMANDMIGRATION parameter applies to SMS-managed volumes only. It is ignored for non-SMS-managed volumes. For non-SMS managed volumes, interval migration is performed instead (if enabled).

SETSYS default: If you specify the ONDEMANDMIGRATION parameter without N (no) or Y (yes), then the SETSYS default for this parameter is N (on-demand migration is not performed).

DFSMSHsm default: If you do not specify ONDEMANDMIGRATION on the SETSYS command during startup, the default is to perform interval migration.

Notes:

1. On-demand migration is performed on a volume when an SMS ENF72 signal is received (indicating the volume has exceeded the high threshold).
2. If you want to perform on-demand migration on TMM storage groups, you must convert AM=I (Auto Migrate, interval) TMM storage groups to AM=Y.
3. On-demand migration is paused when interval migration or primary space management are started on the same DFSMSHsm host performing on-demand migration. On-demand migration will start again after interval migration or primary space management has ended, if there are eligible volumes to be processed.

ODMNOTIFICATIONLIMIT: Specifying an on-demand migration number of concurrently queued volumes notification limit

Explanation: **ODMNOTIFICATIONLIMIT** is an optional parameter specifying a limit before notification of the number of volumes concurrently queued for on-demand migration is issued. When the notification limit is exceeded, highlighted message ARC1901E is issued to the operator's console. When the number of volumes eligible for on-demand migration falls below the notification limit, the message is removed from the operator's console.

The following parameter is an optional parameter of the ODMNOTIFICATIONLIMIT parameter:

limit

Number of volumes (1 to 99999) that must be concurrently queued for on-demand migration before notification is issued.

SMS relationship: The ODMNOTIFICATIONLIMIT parameter applies to SMS-managed volumes being processed by on-demand migration only.

SETSYS default: None.

DFSMSHsm default: If you do not specify ODMNOTIFICATIONLIMIT on the SETSYS command during startup, the default limit is 100.

OPTIMUMDASDBLOCKING and NOOPTIMUMDASDBLOCKING: Specifying whether DFSMSHsm should use its defined optimum block size

Explanation: **OPTIMUMDASDBLOCKING** | **NOOPTIMUMDASDBLOCKING** are mutually exclusive, optional parameters indicating whether DFSMSHsm should use its defined optimum block size when outputting or recovering to DFSMSHsm-owned DASD. Optimum blocking refers to DFSMSHsm writing its output or input in blocks that are multiples of 2 KB and is determined by the device's track size for data being written to DFSMSHsm-owned DASD. For example, all models of 3390 DASD have the same track length and an optimum block size of 18 KB (1 KB equals 1024 bytes). This permits better use of space on DFSMSHsm-owned DASD devices.

OPTIMUMDASDBLOCKING indicates to DFSMSHsm to use its defined optimum block size when writing to DFSMSHsm-owned DASD.

NOOPTIMUMDASDBLOCKING indicates to DFSMSHsm to not use its defined optimum block size when writing to DFSMSHsm-owned DASD. DFSMSHsm writes to DFSMSHsm-owned DASD in 2 KB blocks.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify the NOOPTIMUMDASDBLOCKING parameter on any SETSYS command, DFSMSHsm uses its defined optimum block size when writing to DFSMSHsm-owned DASD as the default.

Note:

1. OPTIMUMDASDBLOCKING does not pertain to small-data-set-packing data sets, dump VTOC copy data sets, or backup VTOC copy data sets.
2. You can change this parameter while DFSMSHsm is running. The change takes effect when the next data set destined for DFSMSHsm-owned DASD is opened for output.

OUTPUTTAPEALLOCATION: Specifying if to wait for output tape to be allocated

Explanation: **OUTPUTTAPEALLOCATION (WAIT | NOWAIT)** is an optional parameter specifying whether DFSMSHsm should wait until the tape unit is allocated for the output tape for migration or backup.

Subparameter	Explanation
WAIT	<p>DFSMSHsm waits until the output tape unit is allocated. Any DFSMSHsm function requiring dynamic allocation services will be unable to proceed until this tape unit is allocated.</p> <p>When using the WAIT option, caution must be exercised to limit the number of DFSMSHsm tasks that require tape. If more tasks run than there are available tape devices, a deadlock condition occurs because of the exclusive enqueue that MVS allocation puts on the task input/output table (SYSZTIOT) while awaiting the tape device. This condition can be resolved only if another job (not DFSMSHsm) releases a tape device, or if you cancel DFSMSHsm.</p>
NOWAIT	<p>DFSMSHsm does not wait until the output tape unit is allocated. Instead, DFSMSHsm reissues the request every 10 seconds up to six times. If the output tape unit still is not allocated after seven tries, DFSMSHsm asks the operator whether DFSMSHsm should cancel the request or repeat the allocation sequence. With the NOWAIT option, dynamic allocation issues message IEF238D. Once answered, other DFSMSHsm tasks can proceed with allocations.</p>

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is NOWAIT.

PARTIALTAPE: Specifying whether to reuse partially full tapes

Explanation: **PARTIALTAPE** is an optional parameter specifying that you want DFSMSHsm to mark a cartridge-type single-file formatted tape volume full when the volume has been demounted.

MARKFULL and REUSE are mutually exclusive parameters of the PARTIALTAPE parameter. One of these parameters must be specified if the PARTIALTAPE parameter is specified. If not, an error message is issued.

Subparameter	Explanation
MARKFULL	<p>DFSMSHsm automatically marks a single-file-formatted tape volume full (except after data set migration or data set backup) independent of the block count when it demounts the tape. These volumes can be marked as full even though the volume is only partially used. DFSMSHsm excludes data set migration or backup to avoid marking a tape volume full after a command migration or after a command backup of a single data set to migration level 2.</p> <p>When running in a PARTIALTAPE(MARKFULL) environment, a generic TAPECOPY command will select, copy, and mark full nonduplexed original tapes that are <i>not</i> marked as full, even if an alternate tape volume serial number is already recorded in the DFSMSHsm inventory (TTOC record). In this case the older alternate is removed from DFSMSHsm's inventory but <i>not</i> explicitly communicated to tape management products as with the ARCTVEXT exit method. DFSMSHsm creates the new alternate because the partial original may have been extended after the old alternate was created by an explicit TAPECOPY. The older alternate is not communicated to tape management products so that the customer can retain alternates in addition to the one DFSMSHsm maintains. DFSMSHsm issues an ARC0436I message to the console and identifies the old alternate, the original, and the new alternate. Customers can update their tape management product if they do not want to retain the old alternates.</p>
REUSE	<p>DFSMSHsm uses the normal process of marking a single-file formatted tape full only when the percent full (specified with the SETSYS TAPEUTILIZATION parameter) is reached.</p>

If you specify the global PARTIALTAPE(MARKFULL | REUSE) parameter, the parameter applies to both migration and backup. If you specify a specific function, the parameter applies only to that function. If you specify a global value and a specific function in a single command, the command will fail.

The PARTIALTAPE(MIGRATION(...)) parameter applies to tape volume migration only. The PARTIALTAPE(BACKUP(...)) parameter applies to tape volume backup only.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: The default is REUSE, meaning that DFSMSHsm marks the tape volume full only when the volume reaches the maximum block count specified with the SETSYS TAPEUTILIZATION parameter.

Note: If you are using a VTS subsystem for migration or backup output, the MARKFULL parameter can improve performance by reducing the need to remount yesterday's partial volume in order to extend it today. Additionally, this usage can somewhat reduce the reclamation process of the VTS subsystem because the physical tape containing the initial partial tape, that was extended, was not invalidated on a true tape.

PDA: Specifying whether the problem determination aid is initiated

Explanation: PDA(NONE | ON | OFF) is an optional subparameter that specifies whether the problem determination aid (PDA) is enabled. The PDA facility gathers diagnostic information and records the information in a storage buffer and in DASD data sets.

You can specify PDA tracing in the DFSMSHsm startup procedure. Specifying the PDA=YES parameter allows the PDA facility to collect trace data prior to the processing of the SETSYS PDA(ON) command.

Subparameter	Explanation
NONE	The PDA facility does not start at DFSMSHsm startup. No space for the storage buffer is requested, and the trace output data sets are not opened. Note: This keyword is effective only at DFSMSHsm startup. If issued after DFSMSHsm startup, NONE has the same results as issuing the OFF keyword.
ON	Storage is requested for the trace buffer and the trace output data sets open. If the trace data sets are not allocated, the problem determination aid function continues to trace in internal storage.
OFF	The trace facilities are suspended but the primary module is left active. The trace buffer storage area remains allocated and the output data sets remain open, but the bit to allow tracing is turned off.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you do not specify either the ON or OFF keyword, the default is ON.

DFSMSHsm default: If you do not specify either the ON or OFF keywords, the default is ON.

PLEXNAME: Specifying a name for an HSMplex

Explanation: **PLEXNAME(hsmplex_suffix)** is an optional parameter that allows you to name multiple HSMplexes within a sysplex. This parameter is used during startup of DFSMSHsm and must be specified in an ARCCMDxx member. This parameter must be specified if there are multiple HSMplexes in a sysplex because this HSMplex name is used to distinguish between the separate HSMplexes.

The prefix of ARC is used by DFSMSHsm and you can specify a maximum of five characters as the suffix for the HSMplex name. The default suffix used by DFSMSHsm is PLEX0.

Note:

1. If an HSMplex name other than the default is specified on one host in an HSMplex, then that name must be specified on all other DFSMSHsm hosts in the HSMplex.
2. If data is to be transferred to cloud storage, the HSMplex name should contain only characters from the alphabetic, numeric, and national character sets (A-Z, a-z, 0-9, and #, \$, and @).
3. If utilizing the cloud storage, it is recommended to specify this parameter with an HSMplex name other than the default to ensure the cloud containers created are isolated from visibility to other HSMplexes in your environment that utilize the same cloud. When a cloud container is created, DFSMSHsm uses the HSMplex name as the 2nd qualifier for the cloud container name.
4. If the HSMplex name is changed, the next migration or dump to cloud that occurs will cause a new container to be created utilizing the new HSMplex name as its 2nd qualifier. After the HSMplex name is changed, the LIST CLOUD(cloud_network_connection_name), AUDIT MEDIACONTROLS(CLOUD(cloud_network_connection_name)) and empty container deletion functions will only utilize the current HSMplex name during their processing. These functions do not have visibility to containers created using an HSMplex name other than the current HSMplex name.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: By default, SETSYS uses PLEXNAME(PLEX0), which causes DFSMSHsm to use an HSMplex name of ARCPLEX0.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the default HSMplex name is ARCPLEX0.

PRIMARYSPMGMTSTART: Specifying the start and stop times for automatic primary space management

Explanation: PRIMARYSPMGMTSTART (*hhmm1* *hhmm2*) is an optional parameter specifying the times when automatic primary space management should start. Use this parameter instead of AUTOMIGRATIONSTART to set the start and stop times for automatic primary space management.

Using PRIMARYSPMGMTSTART(*hhmm1* *hhmm2*), you can specify a planned start time and a planned end time. If DFSMSHsm does not start automatic primary space management between the start and stop times, automatic primary space management is not performed for that day.

For *hhmm1*, substitute the planned start time for automatic primary space management. If you specify *hhmm1* as zero, DFSMSHsm does not start automatic primary space management.

For *hhmm2*, substitute the planned stop time when DFSMSHsm no longer starts automatic primary space management on a new volume. If *hhmm2* is set to zero, this is the same as not including the *hhmm2* parameter.

The *hhmm* format expresses time as a four-digit decimal number that is based on the 24-hour clock. *hh* specifies the hour and *mm* specifies the minutes. For example, you can specify 1:15 p.m. as 1315, and you can specify midnight as 2400. The maximum value you can specify for *hh* is 24, the maximum value you can specify for *mm* is 59, and the maximum value you can specify for *hhmm* is 2400.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If only one time is specified in PRIMARYSPMGMTSTART, DFSMSHsm starts the automatic primary space management functions between the time specified and midnight. Automatic primary space management runs until completion.

DFSMSHsm default: If PRIMARYSPMGMTSTART is not specified, the default is no automatic primary space management (unless AUTOMIGRATIONSTART was specified).

Note:

1. If you do not specify a nonzero value for *hhmm1* during start up of DFSMSHsm, DFSMSHsm does not perform automatic primary space management.
2. The type of automatic primary space management DFSMSHsm does on a particular volume (data set deletion, data set retirement, or migration) depends on the parameter you have specified with the ADDVOL command.
3. If you want the operator to decide whether to start automatic primary space management, specify the REQUEST parameter with the PRIMARYSPMGMTSTART parameter.
4. As with automatic backup, DFSMSHsm serializes access to data sets and volumes for a short period of time. Therefore, select a time for automatic primary space management when the computing system is not very active and there is not much interactive user activity (usually at night).
5. Automatic primary space management usually starts at the planned start time. Assume, however, that DFSMSHsm is not running at the planned start time or the operating system is lost while automatic primary space management is running. When DFSMSHsm restarts, it checks the start (*hhmm1*) and stop (*hhmm2*) times. These start and stop times form a window. If DFSMSHsm restarts during the window, DFSMSHsm also restarts automatic primary space management.
6. PRIMARYSPMGMTSTART fails if *hhmm1* and *hhmm2* are equal.
7. In a test environment, DFSMSHsm can run automatic primary space management multiple times a day. However, this procedure requires certain DFSMSHsm supported patches. Refer to the [z/OS DFSMSHsm Implementation and Customization Guide](#) for more information about running primary space management multiple times a day. If automatic primary space management runs to completion and you want to start automatic primary space management again, use the following procedure:
 - Issue the SETSYS PRIMARYSPMGMTSTART command to define a new start window.
 - Specify a start window that has a planned start time (*hhmm1*) that is after the time that automatic primary space management last ended.

Automatic primary space management starts immediately if the current time is between the start and stop times you specified. If the current time is not between the start and stop times you specified, automatic primary space management starts automatically when the planned start time is reached.

8. If PRIMARYSPMGMTSTART is issued dynamically with a zero value for *hhmm1* to convert to no longer running the function and automatic primary space management is currently running, it runs to completion with the originally established planned stop time. This is true regardless of the *hhmm2* value specified in the newly issued command. DFSMSHsm does not start automatic primary space management again unless another command is issued establishing a new start time. A QUERY SETSYS will show zero for the planned start time and the originally established stop time.

PROFILEBACKUP and NOPROFILEBACKUP: Specifying whether to create backup RACF discrete profiles for cataloged data sets

Explanation: **PROFILEBACKUP** | **NOPROFILEBACKUP** are mutually exclusive, optional parameters that you use to specify whether DFSMSHsm creates a backup RACF discrete profile for all cataloged RACF-indicated data sets during backup.

PROFILEBACKUP specifies that DFSMSHsm creates a backup RACF discrete profile for the cataloged RACF-indicated data set when it backs up the data set. This backup profile applies to all backup versions of the cataloged RACF-indicated data set.

NOPROFILEBACKUP specifies that DFSMSHsm does not create a backup RACF discrete profile when it backs up a cataloged RACF-indicated data set.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the default is **PROFILEBACKUP**.

Note:

1. If a RACF naming conversion table (NCT) is used, the recover function is not able to re-establish the discrete profile protection.
2. Failure to use **PROFILEBACKUP** allows unauthorized users to have access to data sets. Refer to [Chapter 18, “RACF considerations,” on page 617](#) for more information about backing up RACF discrete profiles for cataloged data sets.

PROMOTE: Specifying whether a host is eligible for promotion

Explanation: **PROMOTE** is an optional parameter that allows users to specify which host in an HSMplex may be promoted and which should not be. Currently, one host in an HSMplex is designated as the Primary host and all other hosts are considered secondary. You should specify the **PROMOTE** parameter on only those hosts that are actually eligible for promotion.

Subparameter	Explanation
PRIMARYHOST (YES NO)	Allows you to designate if a host is eligible to take over the unique functions from a failed primary host. PRIMARYHOST(YES) specifies that this specific host in an HSMplex is eligible to take over the unique functions from a failed primary host. If PRIMARYHOST(YES) is issued from a primary host, it is ignored. PRIMARYHOST(NO) specifies that this specific host in an HSMplex is disabled from being eligible for promotion. For example, this can be done if a host is assigned specific work that would conflict with promotion or the host resides on a slow z/OS image.

Subparameter	Explanation
SSM(YES NO)	Allows you to designate a host's eligibility to take over Secondary Space Management (SSM) functions from a failed host. Secondary Space Management generally runs on a single host. This can be the primary host or a specific secondary host. SSM hosts are not eligible for promotion for other SSM hosts because each SSM host may have different or identical SSM windows and cycles defined. SSM(YES) specifies that this specific non-SSM host in an HSMplex is eligible to take over Secondary Space Management functions from a failed host. If SSM(YES) is issued from a SSM host, the message ARC1521I is issued. SSM(NO) specifies that this specific host in an HSMplex is disabled from being eligible for promotion.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, no hosts in an HSMplex are eligible to take over the Primary or Secondary Space Management functions of a failed host.

Note: The SETSYS PROMOTE parameter only makes the host eligible to be promoted for Primary or SSM hosts that become disabled after the command was issued. If you want to make the host take over for a host that is already disabled, you can issue the SETSYS NOEMERGENCY command on this host, which will cause the host to go and look for hosts that are already disabled and in need of being promoted.

RACFIND and NORACFIND: Specifying whether to put RACF-indication on migration copies and backup versions

Explanation: **RACFIND** | **NORACFIND** are mutually exclusive, optional parameters that you use to specify whether DFSMSHsm puts RACF-indication on all migration copies and backup versions.

RACFIND specifies that DFSMSHsm puts RACF-indication on migration copies and backup versions.

NORACFIND specifies that migration copies and backup versions are not RACF-indicated, including those data sets that are password-protected. Specify this parameter only if generic profile checking is activated and generic profiles are defined for migration and backup qualifiers in all DFSMSHsm hosts.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is **RACFIND**.

RECALL: Specifying the target volume to receive recalled data sets

Explanation: **RECALL** is an optional parameter specifying where DFSMSHsm puts the non-SMS-managed recalled data sets.

Subparameter	Explanation
ANYSTORAGEVOLUME	DFSMSHsm recalls a data set to an online primary volume with the use attribute of storage and the primary volume attribute of automatic recall. You specify the primary volume attribute with the ADDVOL command.

Subparameter	Explanation
PRIVATEVOLUME	DFSMSHsm recalls a data set to any online primary volume with the use attribute of public, storage, or private and the primary volume attribute of automatic recall. You specify the primary volume attribute with the ADDVOL command.

LIKE | UNLIKE are mutually exclusive, optional subparameters of the ANYSTORAGEVOLUME or PRIVATEVOLUME subparameters.

Subparameter	Explanation
LIKE	During undirected recall, only primary volumes whose primary volume attributes match the attributes of the primary volume that the data set migrated from are considered as possible target volumes for the recall.
UNLIKE	During undirected recall, volumes whose currently specified primary volume attributes do not match those of the source primary volume are eligible for selection. DFSMSHsm selects the volume for recall from one of three groups of volumes. The following are groups of volumes and how they are ordered: <ul style="list-style-type: none"> • Volumes with like recall attributes • Volumes with unlike recall attributes in automatic backup and backup device category • Volumes with unlike recall attributes in automatic migration

The volumes in each group are ordered according to most-available free space. For more information on defining pools of volumes, refer to [“Defining pools of volumes”](#) on page 105.

SMS relationship: Parameter does not apply to SMS-managed DASD volumes or data sets.

SETSYS default: If you specify RECALL without a subparameter, the SETSYS default is ANYSTORAGEVOLUME. If you do not specify the LIKE or UNLIKE subparameter with the ANYSTORAGEVOLUME or PRIVATEVOLUME subparameter, the SETSYS default is LIKE.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default for data sets is RECALL(ANYSSTORAGEVOLUME).

Note:

1. The LIKE/UNLIKE subparameter does not apply if a data set is SMS-managed when migrated and later is recalled as a non-SMS-managed data set.
2. For BDAM data sets, DFSMSHsm selects only devices that are the same device type as the one the data sets migrated from. The DAOPTION of the RECALL and RECOVER commands can be used to specify how BDAM data sets can be moved to a target volume using relative track or relative block accessing. In addition, target volumes with track sizes different from the last level 0 volume from which the BDAM data set came can be selected.
3. In a JES3 environment, specify the same subparameter for DFSMSHsm hosts that set up jobs and for DFSMSHsm hosts that run the jobs.

RECYCLEINPUTDEALLOCFREQUENCY: Allowing periodic deallocation of input unit during recycle processing

Explanation: **RECYCLEINPUTDEALLOCFREQUENCY(BACKUP(*bfreq*) MIGRATION(*mfreq*))** is an optional parameter that you use to periodically deallocate an input unit during recycle processing. This dynamically changeable parameter prevents DFSMSHsm from possibly keeping an input unit allocated for hours.

Tape Library Relationship: In a tape environment where contention for tape drives may be a consideration, the following table shows BACKUP(*bfreq*) or MIGRATION(*mfreq*) values that are recommended when you recycle single-file-format cartridges:

Type of Environment	Value	Resulting Allocation
A single IBM 3494/3495 Tape Library Dataserver environment having all compatible drives	0	The input unit is allocated for the duration of recycle processing.
An SMS-managed tape environment that has multiple IBM 3494/3495 machines where inputs can come from different ATLS	0	A unique allocation is allowed for each input connected set.
A BTLS-managed environment that has multiple IBM 3494/3495 machines where inputs can come from different ATLS	1	RECYCLE allocates tape drives for each connected set in the correct library.
A multiple non-SMS-managed tape library where not all backup or migration tapes are in a single library (for example, an STK® ACS cluster where you want to prevent pass-through processing)	1	RECYCLE allocates tape drives for each connected set in the correct library.
An environment that has incompatible tapes even though they may appear the same, including when one has a mix of real and emulated devices, such as 3490s and emulators of 3490s	1	RECYCLE allocates tape drives for each connected set.
A tape environment with manual tape operators, but no incompatible mix of real and emulated devices that appear as the same device	5 or 10	New allocations are allowed every 5 to 10 input connected sets.

Subparameter	Explanation
BACKUP(<i>bfreq</i>)	During recycle processing of backup volumes, deallocation of an input unit will occur after the specified number of input connected sets representing single-file-format cartridges have been processed. For <i>bfreq</i> , substitute a decimal number from 1 to 255 for the number of connected sets to process before deallocation of the input unit. Specifying 0 will retain the input unit until recycle processing has completed.
MIGRATION(<i>mfreq</i>)	During recycle processing of migration volumes, deallocation of an input unit will occur after the specified number of input connected sets representing single-file-format cartridges have been processed. For <i>mfreq</i> , substitute a decimal number from 1 to 255 for the number of connected sets to process before deallocation of the input unit. Specifying 0 will retain the input unit until recycle processing has completed.

Note:

- For example, specifying SETSYS RECYCLEINPUTDEALLOCFREQUENCY(BACKUP(20)) will deallocate the input unit every 20 input backup connected sets. Using SETSYS RECYCLEINPUTDEALLOCFREQUENCY(MIGRATION(1)) will deallocate the input unit after each input migration connected set has been processed.
- The input unit will always be deallocated between connected sets when the tape volume is not a single-file-format cartridge.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter (or either subparameter) on any SETSYS command, the DFSMSHsm default is that recycle processing will keep the input unit allocated until the processing of that type of volume (backup or migration) is complete.

RECYCLEOUTPUT: Limiting the selection and allocation of an output volume during recycle processing

Explanation: **RECYCLEOUTPUT(BACKUP MIGRATION)** is an optional parameter that you use to limit the selection and allocation of an output volume during recycle processing.

Tape Library Relationship: A UNITNAME can be used for filtering in the ACS routines. If you specify the unittype for this command, it will be sent to the ACS routines in SMS for the RECYCLEOUTPUT function.

Subparameter	Explanation
BACKUP(<i>unittype</i>)	During recycle processing of a tape backup volume, only tape spill backup volumes that can be mounted and written on the specified type of unit are used for output. In addition, when the tape spill backup volume is allocated for output during recycle processing, the volume is allocated using the unit name specified with the BACKUP subparameter of the RECYCLEOUTPUT parameter, overriding the unit name specified on the ADDVOL command for the tape spill backup volume.
MIGRATION(<i>unittype</i>)	During recycle processing of a tape level 2 migration volume, only tape level 2 migration volumes that can be mounted and written on the specified type of unit will be used for output. In addition, when the tape level 2 migration volume is allocated for output during recycle processing, the volume is allocated using the unit name specified with the MIGRATION subparameter of the RECYCLEOUTPUT parameter, overriding the unit name specified on the ADDVOL command for the tape level 2 migration volume.

For *unittype*, specify a generic or esoteric unit name. The following generic unit names are acceptable: 3480, 3480X, 3490, and 3590-1. If an esoteric unit name is specified, it must have been previously defined to DFSMSHsm using the USERUNITTABLE parameter of the SETSYS command.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If *unittype* is not specified with the BACKUP parameter, there is no restriction on the type of tape spill backup volume that can be selected for output during the recycle processing of a tape backup volume. If *unittype* is not specified here and a scratch tape is required, the value specified for SETSYS UNITNAME is used when an output tape is allocated for a backup volume.

If *unittype* is not specified with the MIGRATION parameter, there is no restriction on the type of tape level 2 migration volume that can be selected for output during the recycle processing of a tape level 2 migration volume. If *unittype* is not specified here and a scratch tape is required, the value specified for SETSYS MIGUNITNAME is used when an output tape is allocated for a migration volume.

If the RECYCLEOUTPUT parameter is specified without any subparameters, no changes are made to the current DFSMSHsm values.

DFSMSHsm default: None.

RECYCLEPERCENT: Specifying the maximum percentage of valid data on a backup tape

Explanation: **RECYCLEPERCENT(*pct*)** is an optional parameter specifying the percent-valid criteria DFSMSHsm uses to recycle tape backup volumes if you do not specify the PERCENTVALID parameter of the RECYCLE command. For *pct*, substitute a decimal number from 0 to 100.

When DFSMSHsm invalidates a data set entry for a full backup volume and the percentage of valid data on the tape backup volume becomes less than or equal to *pct*, DFSMSHsm writes the informational message, ARC0365I, in the command activity log. You then use the RECYCLE command to recycle all eligible tape backup volumes.

For a discussion of how DFSMSHsm calculates percentages based on valid blocks, refer to [Chapter 13, “Managing DFSMSHsm media,”](#) on page 527.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 20%.

Note:

1. DFSMSHsm ignores the RECYCLEPERCENT parameter of this command and the PERCENTVALID parameter of the RECYCLE command when you use the RECYCLE command to recycle specific backup volumes.
2. DFSMSHsm also uses this value for migration level 2 tape volumes when the ML2RECYCLEPERCENT parameter of the SETSYS command has not been specified.

RECYCLETAKEAWAYRETRY: Specifying whether to automatically generate a new RECYCLE command for a tape

Explanation: **RECYCLETAKEAWAYRETRY (NO | YES MAXRETRYATTEMPTS(*nn*) DELAY(*ssss*))** is an optional parameter specifying whether DFSMSHsm should automatically generate a new RECYCLE command for a tape when the original recycle must terminate due to the takeaway process, or when the tape is in use by the another DFSMSHsm task.

Note that each tape volume has its own retry count and can be retried up to *nn* times (12 by default). This count will be reset for this tape only when all data sets from this tape are recycled.

If YES is specified, then the following subparameters can also be specified:

MAXRETRYATTEMPTS(*nn*)

sets the maximum number of recycle retry attempts. The value can be 1-99. The default is 12.

DELAY(*ssss*)

sets the delay interval in seconds between recycle attempts. The value can be 1-9999. The default is 300.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do *not* specify this parameter on any SETSYS command, the DFSMSHsm default is to *not* automatically generate a new RECYCLE command for a tape. When RECYCLETAKEAWAYRETRY is specified without the YES or NO keyword, the default is NO. If either the MAXRETRYATTEMPTS or DELAY subparameters are *not* specified, the defaults are 12 and 300, respectively.

RECYCLETAPEALLOCATION: Specifying if to wait for recycle tape to be allocated

Explanation: **RECYCLETAPEALLOCATION (WAIT | NOWAIT)** is an optional parameter specifying whether DFSMSHsm should wait until the tape unit is allocated for the source and target tape volumes.

Subparameter	Explanation
WAIT	<p>DFSMSHsm waits until the recycle tape unit is allocated. All DFSMSHsm functions requesting allocations stop until this allocation request is satisfied.</p> <p>When using the WAIT option, you must exercise caution to limit the number of DFSMSHsm tasks that require tape. If more tasks run than there are available tape devices, a deadlock condition occurs because of the exclusive enqueue that MVS allocation will put on the task input/output table (SYSZTIOT) while awaiting the tape device. This condition can be resolved only if another job (not DFSMSHsm) releases a tape device, or if you cancel DFSMSHsm.</p>
NOWAIT	<p>DFSMSHsm does not wait until the recycle tape unit is allocated. Instead, DFSMSHsm reissues the request every 10 seconds up to six times. If the recycle tape unit is still not allocated after seven tries, DFSMSHsm asks the operator whether to have DFSMSHsm cancel the request or repeat the allocation sequence. With the NOWAIT option, dynamic allocation issues message IEF238D. Once answered, other DFSMSHsm tasks can proceed with allocations.</p>

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is NOWAIT.

RECYCLETAPERESIDENCE: Specifying which IBM VTS tape volumes should be included in RECYCLE processing

Explanation: **RECYCLETAPERESIDENCE (BOTH | INCACHE | NOTINCACHE)** setting indicates which eligible IBM VTS tape volumes will be included in RECYCLE processing. IBM VTS tape volumes that are eligible for recycle and reside either in cache, not in cache, or both in cache and not in cache as specified via the RECYCLETAPERESIDENCE parameter will be included during RECYCLE processing.

SETSYS default: BOTH

DFSMSHsm default: If the **SELECT(TAPERESIDENCE(BOTH | INCACHE | NOTINCACHE))** parameters are specified on the RECYCLE command, they will take precedence over the SETSYS RECYCLETAPERESIDENCE setting.

REMOVECOMPACTNAMES: Removing the compaction-control qualifier for source or object data sets

Explanation: **REMOVECOMPACTNAMES(name1,name2,...)** is an optional parameter specifying the compaction-control qualifier of all data sets to be removed from the SOURCENAMES or OBJECTNAMES table. For *name1,name2,...*, specify the compaction-control qualifiers of those data sets you no longer want to have compacted by the source or object compaction table.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, there is no DFSMSHsm default.

REQUEST and NOREQUEST: Specifying whether the operator's permission is required

Explanation

REQUEST and NOREQUEST are mutually exclusive, optional parameters specifying whether the operator's permission is required before DFSMSHsm starts any of the following actions:

- automatic primary space management
- automatic secondary space management
- interval migration
- automatic backup
- automatic dump

Note: Operator permission is never required for on-demand migration, even if SETSYS REQUEST is specified.

REQUEST

Operator's permission is required before DFSMSHsm starts any of the actions listed above.

NOREQUEST

Operator's permission is not required before DFSMSHsm starts any of the actions listed above.

SMS relationship

Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default

None.

DFSMSHsm default

If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is NOREQUEST.

SCRATCHFREQUENCY: Specifying how long to retain list data sets

Explanation: **SCRATCHFREQUENCY(days)** is an optional parameter specifying the number of days DFSMSHsm should keep list data sets. For *days*, substitute a decimal number from 0 to 9999 specifying how long DFSMSHsm should keep an unused list data set. If you specify 0, DFSMSHsm scratches list data sets during the next space management of the volume. If the last qualifier in a data set name is .LIST, .OUTLIST, or .LINKLIST, the data set is a list data set. DFSMSHsm scratches list data sets during volume space management after the specified number of days have passed since anyone used the data sets.

SMS relationship: Parameter does not apply to SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 9999 days.

SECONDARYSPMGMTDIRECTIVE: Specifying Secondary Space Management controls pertaining to Cloud and non-Cloud data

Explanation: **SECONDARYSPMGMTDIRECTIVE** is an optional parameter that is used to direct an instance of DFSMSHsm Secondary Space Management to do either one of the following exclusively:

ALL

Secondary Space Management might process data regardless whether it is in the Cloud or not.

CLOUDONLY

Secondary Space Management might only process data in the Cloud and run Empty Container Deletion.

NOCLOUD

Secondary Space Management might not process any data that resides in the Cloud and might not run Empty Container Deletion. All other tasks that Secondary Space Management might run are permissible to run (for example, ML1/ML2 Data Set Expiration, ML1 -> ML2 migration, statistics cleanup, and so on).

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: ALL.

DFSMSHsm default: If you do not specify this parameter, DFSMSHsm defaults to SECONDARYSPMGMTDIRECTIVE(ALL).

Note: Within a given DFSMSHsm SYSPLEX or MASH environment, multiple instances of Secondary Space Management (SSM) might be running. If two, or more, of these instances overlap in run time with mutually exclusive directives declared (for example, SSMDIR(CLOUDONLY) and SSMDIR(NOCLOUD)), the second instance is blocked from executing. Also, message ARC0552I is received indicating that MIGRATION CLEANUP is not performed on that processing unit. Therefore, it is important to make sure SSM instances with mutually exclusive directives do not overlap in run time. More specifically, it is important to ensure that SSM with the CLOUDONLY directive is started at a different time than another SSM with the ALL or NOCLOUD directives, and vice versa.

SECONDARYSPMGMTSTART: Specifying the start and stop time for automatic secondary space management

Explanation: **SECONDARYSPMGMTSTART(hhmm1 hhmm2)** is an optional parameter you use to specify when automatic secondary space management functions are to start and stop. If DFSMSHsm does not start automatic secondary space management between the specified start and stop times, automatic secondary space management functions are not done for that day.

For *hhmm1*, substitute the planned start time for automatic secondary space management. If you specify it as 0000, secondary space management will not start.

For *hhmm2*, substitute the planned stop time for automatic secondary space management. If *hhmm2* is set to zero, this is the same as not including the *hhmm2* parameter.

Note: Time is based on the 24-hour clock and is expressed as a four-digit decimal number in the format *hhmm*. The hour is represented by *hh* and the minutes by *mm*. For example, 1:15 p.m. is 1315 and midnight is 2400. The maximum value you can specify for *hh* is 24, the maximum value you can specify for *mm* is 59, and the maximum value you can specify for *hhmm* is 2400.

The SETSYS SECONDARYSPMGMTSTART command will fail if the start and stop times are the same.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If only one time is specified in SECONDARYSPMGMTSTART, DFSMSHsm starts the automatic secondary space management functions between the time specified and midnight. Automatic secondary space management runs until completion.

DFSMSHsm default: If you do not specify this parameter, no automatic secondary space management functions are done. ML1 volumes will eventually run out of space.

Note:

1. Any outstanding MWEs for data set deletion that were built by secondary space management up to the stop time, continue to be processed. After the stop time, secondary space management

generates no new MWEs. When many MWEs are waiting for processing, their processing may continue well after secondary space management itself has stopped processing. However, recalls and user-requested deletes have a higher priority than those deletes that are performed from secondary space management.

2. If SECONDARYSPMGMTSTART is issued dynamically with a zero value for *hhmm1* to convert to no longer running the function and automatic secondary space management is currently running, it runs to completion with the originally established planned stop time. This is true regardless of the *hhmm2* value specified in the newly issued command. DFSMSHsm does not start automatic secondary space management again unless another command is issued establishing a new start time. A QUERY SETSYS will show zero for the planned start time and the originally established stop time.

SELECTVOLUME: Specifying whether DFSMSHsm should select a scratch tape volume or a DFSMSHsm tape volume

Explanation: **SELECTVOLUME** is an optional parameter specifying whether DFSMSHsm chooses a tape volume owned by DFSMSHsm or a scratch tape volume when an end-of-volume (EOV) condition occurs while DFSMSHsm is writing on a tape volume. **SELECTVOLUME(SCRATCH | SPECIFIC)** applies to the initial selection for dump tape volumes as well as to the EOV condition.

If you specify the global **SELECTVOLUME(SCRATCH | SPECIFIC)** parameter, the parameter applies to all functions (migration, backup, and dump). If you specify a specific volume type with the **SELECTVOLUME(MIGRATION | BACKUP | DUMP)** parameter, the parameter applies only to that type of function. If you specify a global value and a specific volume type in a single command, the command fails.

Subparameter	Explanation
SCRATCH	A PRIVAT mount request is always sent to the operator when an EOV condition occurs on a tape backup volume, tape migration level 2 volume, or dump tape volume. The operator should mount a private scratch tape volume.
SPECIFIC	DFSMSHsm chooses another tape volume when an EOV condition occurs on a tape backup volume, tape migration level 2 volume, or dump tape volume. DFSMSHsm chooses a tape volume you have already defined to DFSMSHsm with the ADDVOL command. If the EOV condition occurs on a tape backup volume, DFSMSHsm chooses another tape backup volume. If the EOV condition occurs on a tape migration level 2 volume, DFSMSHsm chooses another tape migration level 2 volume. If the EOV condition occurs on a dump tape volume, DFSMSHsm chooses another dump tape volume. DFSMSHsm first attempts to select a dump tape volume that is assigned to the same class as is currently being dumped to. If a dump tape volume with the same class cannot be found, then DFSMSHsm looks for a dump tape volume that is not assigned to any class. When DFSMSHsm chooses the tape volume, the operator receives a message specifying which tape to mount. If DFSMSHsm cannot find an acceptable tape volume, the operator receives a PRIVAT mount request.

DFSMSHsm distinguishes between devices that have the same logical device type but which have different physical devices. There are four types of 3490's:

- Emulated on a 128 track device
- Emulated on a 256 track device
- Emulated on a virtual tape server (VTS) subsystem as a virtual tape volume
- All others

There are three types of 3590's:

- Native 128 track devices

- Native 256 track devices
- Native 384 track devices

DFSMSHsm examines the technology of the restricting output unit name and prefers selecting a partially filled cartridge having that recording format, at initial selection for backup to tape or migration to tape. If no cartridge matching the recording technology is found, then a nonspecific mount is requested.

Note: If a specific volume is specified for output during initial or subsequent (end-of-volume) processing the following requirements apply:

- If a library device is allocated:
 - The tape volume must be in the library for it to be considered eligible for selection. If the tape volume is outside the library, it is marked unavailable in the DFSMSHsm inventory.
 - If a data class is assigned to the DFSMSHsm tape data set name and the ACS COMPACTION attribute is specified (Y or N), the following considerations apply:

A partial tape is eligible for initial tape selection when the compaction status is consistent with the ACS data class COMPACTION attribute, and its device type is consistent with the allocated device type.

If a tape volume is empty, the existing selection rules related to device type eligibility still apply.
 - If a data class is not assigned or a data class is assigned to the DFSMSHsm tape data set name and the ACS COMPACTION attribute is *not* specified (blank), the following considerations apply:

A partial tape is eligible for initial tape selection when its device type is consistent with the allocated device type (the compaction status assigned to a partially full tape does not influence initial tape selection).

If a tape volume is empty, the existing selection rules related to device type eligibility still apply.
- If a non library device is allocated, the tape volume must *not* be in a library for it to be considered eligible for selection. If the tape volume is in a library, it is marked unavailable in the DFSMSHsm inventory.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is SCRATCH.

Note:

1. Use the SCRATCH subparameter if you have the cartridge loader of the 3480 Magnetic Tape Subsystem. The cartridge loader provides the operator with the capability to preload multiple tapes used as migration, backup, or dump tapes. The system then writes migration data sets, backup data sets, or dump copies to these tapes without operator intervention.
2. The volume verification exit can reinitialize a tape volume if the tape remains under control of DFSMSHsm. Therefore, to use the volume verification exit effectively, specify SPECIFIC and TAPEDELETION(HSMTAPE).
3. This parameter has no effect when aggregate recovery is recovering data sets to ML2 tape volumes. Aggregate recovery will always request the mount of a non specific (scratch) volume.
4. Repeating the SELECTVOLUME parameter for specific volume types in a single SETSYS command causes conflict. The last entered SELECTVOLUME parameter overrides all other SELECTVOLUME parameters in this single command and is the only parameter processed.

For example, if you enter the following command, only the SELECTVOLUME(DUMP(SCRATCH)) parameter is processed. The SELECTVOLUME(MIGRATION(SPECIFIC)) and the SELECTVOLUME(BACKUP(SCRATCH)) parameters are not processed and their DFSMSHsm defaults are applied.


```
SETSYS SELECTVOLUME(MIGRATION(SPECIFIC)) -
SELECTVOLUME(BACKUP(SCRATCH)) -
SELECTVOLUME(DUMP(SCRATCH))
```

To prevent this from happening, enter the command as follows. In this way, all of the parameters are processed.

```
SETSYS SELECTVOLUME(MIGRATION(SPECIFIC)) -
BACKUP(SCRATCH) -
DUMP(SCRATCH))
```

Another way to enter this command is to repeat the SETSYS command three times with different subparameters for the SELECTVOLUME parameter. Issuing the SELECTVOLUME command in this way allows you to group parameters together in separate sections of your PARMLIB member.

```
SETSYS SELECTVOLUME(MIGRATION(SPECIFIC))
SETSYS SELECTVOLUME(BACKUP(SCRATCH))
SETSYS SELECTVOLUME(DUMP(SCRATCH))
```

SKIPABPRIMARY and NOSKIPABPRIMARY: Specifying whether to skip automatic backup of primary volumes

Explanation: **SKIPABPRIMARY** | **NOSKIPABPRIMARY** are mutually exclusive, optional parameters specifying whether or not to back up data sets on DFSMSHsm-managed volumes with:

- The primary volume attribute of automatic backup, specified with the PRIMARY(AUTOBACKUP) parameter of the ADDVOL command
- The storage group attribute for automatic backup, specified with the AUTOBACKUP=YES for the storage group.

SKIPABPRIMARY specifies that during automatic backup, DFSMSHsm skips the backup of DFSMSHsm-managed volumes with:

- The primary volume attribute of automatic backup
- The storage group attribute of AUTOBACKUP=YES

Therefore, all users must back up their data sets on the DFSMSHsm-managed volumes. DFSMSHsm, however, backs up data sets while the data sets are on migration volumes that migrated from a volume with the primary volume attribute of automatic backup or from a storage group with the attribute of AUTOBACKUP=YES.

NOSKIPABPRIMARY specifies that during automatic backup, DFSMSHsm backs up primary volumes with the automatic backup attribute and storage groups with the attribute of AUTOBACKUP=YES.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is **NOSKIPABPRIMARY**.

SMALLDATASETPACKING and NOSMALLDATASETPACKING: Specifying whether DFSMSHsm should use small data set packing

Explanation: **SMALLDATASETPACKING** | **NOSMALLDATASETPACKING** are mutually exclusive, optional parameters specifying whether or not to pack small data sets into predefined VSAM key-sequenced small-data-set-packing data sets on migration level 1 volumes.

SMALLDATASETPACKING specifies that when a small data set migrates, it is eligible to be packed into a predefined VSAM key-sequenced small-data-set-packing data set on a migration level 1 volume.

For *tracks*, substitute a decimal number from 1 to 18 specifying that DFSMSHsm consider any data set using that number of tracks or fewer as a candidate for migration to a small-data-set-packing data set. DFSMSHsm converts the *tracks* value to kilobytes with an assumed track size of device type 3380.

KB(*kilobytes*) specifies the source size of a data set to be considered as a candidate for migration to a small-data-set-packing data set. For *kilobytes*, substitute a decimal number from 1 to 800.

The source size of a DFSMSHsm data set is calculated by multiplying the number of bytes per track for the device type times the number of tracks used by the data set. This calculated value is compared to the value specified in the SMALLDATASETPACKING parameter. If the calculated value is equal to or less than the value specified in the SMALLDATASETPACKING parameter, the data set is a candidate to be migrated to an SDSP. A candidate data set is eligible for migration to an SDSP data set if its estimated size, after compaction, is less than or equal to 400 kilobytes or to the value you specify, whichever is smaller. The number of bytes per track for each DFSMSHsm device type is as follows:

- 3380 is 47,476 bytes per track
- 3390 is 56,664 bytes per track

The compaction size of a DFSMSHsm data set is determined in one of two ways:

1. If compaction during migration is not active, the source size equals the migration size.
2. If compaction for migration is active, an internal default of 50% is used in establishing the estimated compaction size for data sets that are candidates for an SDSP. Previous compaction history is not considered for determining SDSP candidacy.

NOSMALLDATASETPACKING specifies that when small data sets migrate, they are not packed into predefined VSAM key-sequenced small-data-set-packing data sets on a level 1 volume, even if you specified the MIGRATION(SMALLDATASETPACKING) parameter of the ADDVOL command for the volume DFSMSHsm selects.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify *tracks* or KB(*kilobytes*) with the SMALLDATASETPACKING parameter on any SETSYS command, the DFSMSHsm default is 120 KB.

If you do not specify either SMALLDATASETPACKING or NOSMALLDATASETPACKING, the DFSMSHsm default is NOSMALLDATASETPACKING.

Note:

1. This parameter applies only to data sets that migrate from primary volumes to migration level 1 volumes.
2. If both *tracks* and *kilobytes* are specified, only the *kilobyte* value is honored.
3. Small-data-set-packing data sets have the following advantages:
 - The volume table of contents of the migration level 1 volumes are not filled with DSCBs for small data sets.
 - Small data sets become records in the small-data-set-packing data sets, so they do not have to start on track boundaries, giving you a better use of space on the migration level 1 volumes.
 - You get reduced volume fragmentation on the migration level 1 volumes.
4. Specify both primary and secondary extents for the small-data-set-packing data sets. If you want to use small data set packing, you should specify the SMALLDATASETPACKING parameter of the ADDVOL command for at least one migration level 1 volume per migration task. You do not have to specify SMALLDATASETPACKING for all migration level 1 volumes. If a data set is eligible for small data set packing, DFSMSHsm selects a volume that has been added to DFSMSHsm with the SMALLDATASETPACKING parameter.
5. The names of small-data-set-packing data sets have the following format:

uid.SMALLDS.Vvolser

DFSMSHsm replaces *uid* with the UID you specified in the DFSMSHsm startup procedure.

6. Encrypted files will not use the SDSP data sets at targets for migration to ML1 DASD.

Related reading

- For more information about UID and the DFSMSHsm startup procedure, see [Chapter 17, “Operating procedures for DFSMSHsm,”](#) on page 593.
- For more information about small-data-set-packing data sets and how to create them, see [“Managing SDSP data sets”](#) on page 555.

SMF and NOSMF: Specifying whether DFSMSHsm writes SMF records

Explanation: **SMF(*smfid*)** | **NOSMF** are mutually exclusive, optional parameters specifying whether DFSMSHsm writes SMF records that contain DFSMSHsm statistics.

SMF(*smfid*) specifies that DFSMSHsm write SMF records in the SYS1.MANX or SYS1.MANY system data sets. For *smfid*, substitute a record identification. Use SMF user codes for the record identification in the range of 128 to 255.

If you specify *smfid*, DFSMSHsm writes records with SMF identifications of *smfid* and *smfid*+1. Records with an identification of *smfid* contain daily statistics (DSR) and volume statistics (VSR). Records with an identification of *smfid*+1 contain function statistics (FSR) and ABARS function statistics (WWFSR).

Note: When a DFSMSHsm function executes, only selected fields within the FSR record are set. The fields set depend upon the function being performed and the method used to request the function. The fields that are set are required by the DFSMSHsm REPORT command.

NOSMF specifies that no SMF records are to be written.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is NOSMF.

Additional information on this subject is available in the [z/OS DFSMSHsm Implementation and Customization Guide](#).

SOURCENAMES: Specifying the compaction-control qualifier of the source data set

Explanation: **SOURCENAMES(*name1,name2,...*)** is an optional parameter specifying the compaction-control qualifier of data sets that should be compacted with the table for source data sets. For *name1,name2,...*, specify the compaction-control qualifiers of the data set names of those data sets you want to compact with the table for data sets that contain source code for programming languages. For example, you could specify BAS and ASM as qualifiers. For generation data group data sets, this compaction-control qualifier is the next-to-last qualifier of the data set name. For all other data sets, this compaction-control qualifier is the last qualifier of the data set name.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is that the general compaction table can compact the source data sets.

SPILL and NOSPILL: Specifying whether DFSMSHsm does spill or cleanup processing on full DASD daily backup volumes

Explanation: **SPILL** | **NOSPILL** are mutually exclusive, optional parameters specifying whether DFSMSHsm can do spill or cleanup processing on full-DASD daily backup volumes during backup volume selection and limiting the selection and allocation of an output volume during spill processing.

SPILL specifies that DFSMSHsm spills full DASD daily backup volumes when a DASD daily backup volume is needed and none is available. Spill processing moves all but the latest backup version of each data set from the daily backup volumes to the spill backup volumes.

Subparameter	Explanation
ANY	Any type of spill backup volume can be used for output during spill processing.
DASD	Only DASD spill backup volumes are used for output during spill processing. TAPE specifies that only tape spill backup volumes be used for output during spill processing. In addition, when a tape spill backup volume is allocated during spill or recycle processing, the volume is allocated using the unit name specified on the ADDVOL command for the volume.
TAPE(<i>unittype</i>)	Only tape spill backup volumes that can be mounted and written on the specified <i>unittype</i> are used during spill processing. In addition, when the tape spill backup volume is allocated during spill processing, the volume is allocated using the unit name specified with the TAPE subparameter of the SPILL parameter, overriding the unit name specified on the ADDVOL command for the tape spill backup volume. For <i>unittype</i> , specify a generic or esoteric unit name. The following generic unit names are acceptable: 3480, 3480X, 3490, and 3590-1. If an esoteric unit name is specified, it must have been previously defined to DFSMSHsm using the USERUNITTABLE parameter of the SETSYS command.

NOSPILL specifies that DFSMSHsm never spills full DASD daily backup volumes. Instead, DFSMSHsm cleans up full DASD daily backup volumes when one is needed and none is available. During cleanup processing, DFSMSHsm scratches the following data sets from the full unallocated DASD daily backup volume that DFSMSHsm has not cleaned up in the longest period of time:

- Unneeded VTOC copy data sets
- Unneeded VCAT copy data sets
- Backup versions without backup-version (MCC) records.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: If you specify SPILL without any value, any previous spill output specification is retained.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is SPILL(ANY).

Note: During spill, DFSMSHsm does not keep the association of a primary volume and a particular backup device category. You specify this association with the PRIMARY(BACKUPDEVICECATEGORY) parameter of the ADDVOL command. For example, if you specify ADDVOL PRIMARY(BACKUPDEVICECATEGORY(DASD)), the backup version can be moved to a tape spill backup volume; however, the latest backup version of each data set on the primary volume remains on a DASD daily backup volume.

STORAGEGROUPAFFINITY: Define the set of host names that have affinity to process a storage group cloud migration

STORAGEGROUPAFFINITY is an optional parameter that specifies a set of hosts that have affinity to process a storage group. Up to 15 host names can be specified.

Required subparameters of the SETSYS STORAGEGROUPAFFINITY command:

A set of hosts that have affinity to process a storage group.

Up to 15 host names can be specified. If a set of hosts name is empty, the list of previously-defined host names is removed.

Alias: SGA

SWAP and NOSWAP: Specifying whether the DFSMSHsm address space can be swapped

Explanation: **SWAP** | **NOSWAP** are mutually exclusive, optional parameters specifying that the DFSMSHsm address space can be swapped.

SWAP specifies that the DFSMSHsm address space can be swapped by the MVS system resource manager. NOSWAP specifies that the DFSMSHsm address space cannot be swapped by the MVS system resource manager.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either parameter on any SETSYS command, the DFSMSHsm default is whatever your computing system would normally do with the DFSMSHsm address space.

Note:

1. DFSMSHsm cannot be swapped while secondary address spaces are processing. While ABARS secondary address spaces are active, DFSMSHsm always sets itself to NOSWAP mode. Once all secondary address spaces have completed processing, DFSMSHsm will reset to the original SETSYS SWAP | NOSWAP setting.
2. To support cross-address space communication, DFSMSHsm cannot be swapped when it is in nonlocal mode. If SETSYS SWAP is issued in the startup PARMLIB member, DFSMSHsm is changed to NOSWAP mode temporarily and message ARC0062I is issued. If SETSYS SWAP is issued while DFSMSHsm is in nonlocal mode, DFSMSHsm is not swapped and message ARC0103I is issued.

SYSOUT: Directing DFSMSHsm output

Explanation: **SYSOUT(class copies forms)** is an optional parameter specifying the class, number of copies, and any special forms used to print output from DFSMSHsm. For *class*, substitute an alphanumeric character for the class that DFSMSHsm uses when it prints output. For *copies*, substitute a decimal number from 1 to 99 for the number of copies to print of DFSMSHsm output. For *forms*, substitute up to 4 alphanumeric characters for the special forms to use when printing DFSMSHsm output.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm defaults are class A, one copy, and no special form.

SYS1DUMP and NOSYS1DUMP: Specifying where DFSMSHsm dumps should be written

Explanation: **SYS1DUMP** | **NOSYS1DUMP** are mutually exclusive, optional parameters specifying whether DFSMSHsm dumps resulting from an abnormal end or error condition in primary or secondary address spaces be written in a system dump data set.

SYS1DUMP specifies that DFSMSHsm dumps be written in a system dump data set. This format is required if using the Interactive Problem Control System (IPCS). When you specify the SETSYS SYS1DUMP command and you have specified the SUPPRESSALL keyword for SVCDUMP in the SYS1.PARMLIB member ADYSETxx, the dump analysis elimination (DAE) function suppresses duplicate SVCDUMPS.

NOSYS1DUMP specifies that when DFSMSHsm abends, the dump produced is directed to the dump related DD statements specified in the DFSMSHsm startup procedure. These DD statements are: SYSABEND, SYSUDUMP, and SYSMDUMP.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is SYS1DUMP.

TAPEDATASETORDER: Specifying order for processing tape

Explanation: **TAPEDATASETORDER** is an optional parameter that specifies the order in which you want DFSMSHsm to recall or recover data sets from a single tape.

PRIORITY | **FBID** are mutually exclusive sub-parameters of the TAPEDATASETORDER parameter. If you specify TAPEDATASETORDER, you must also specify either PRIORITY or FBID, otherwise DFSMSHsm does not make any change to the TAPEDATASETORDER state and issues an error message.

Subparameter	Explanation
PRIORITY	DFSMSHsm processes data sets in the priority order. Data sets within the same priority are processed in the FIFO order.
FBID	DFSMSHsm processes the data sets in the ascending file block id order.

If you specify TAPEDATASETORDER(PRIORITY | FBID) parameter, the parameter applies to both recall and recover processing. If you specify a specific functional parameter, TAPEDATASETORDER applies only to that function. If you specify a general option and a specific function in a single command, the command fails. If you specify the RECALL(PRIORITY | FBID) parameter, the parameter applies to recall processing only. If you specify the RECOVER(PRIORITY | FBID) parameter, the parameter applies to recover processing only.

SETSYS default: None.

DFSMSHsm default: If you do not enter the TAPEDATASETORDER command, the DFSMSHsm default is PRIORITY. DFSMSHsm recalls and recovers data sets from a single tape in priority order. Recall requests with the same priority are processed in FIFO order.

Note:

1. The TAPEDATASETORDER command provides the installation with additional flexibility to handle circumstances in which ascending FBID order offers a significant reduction in tape repositioning, such as when a large number of data sets are being recalled or recovered from a single tape. Specifying FBID might increase throughput, but might cause lower priority requests to be processed prior to a higher priority requests.

2. If a host connected to the Common Recall Queue issues the SETSYS command to change the TAPEDATASETORDER, all hosts connected to the Common Recall Queue adopt the same processing order.
3. If you enter the SETSYS command to change the TAPEDATASETORDER while there are requests residing in the Common Recall Queue, it might take some time before the new processing order takes effect.
4. If you enter the ALTERPRI command to re-prioritize requests residing on the Recall or Recover Queue to HIGH, ALTERPRI takes precedence over FBID.

TAPEDELETION: Specifying what to do with tapes that no longer contain valid data

Explanation: **TAPEDELETION** is an optional parameter specifying whether recycled tape volumes be deleted from DFSMSHsm control and become scratch tapes, or are to remain under DFSMSHsm control as unassigned, available tape backup volumes, migration level 2 volumes, or dump volumes.

Subparameter	Explanation
HSMTAPE	The recycled volumes remain under DFSMSHsm control and become either unassigned backup volumes or available migration level 2 volumes, based on the previous definition of the volume. The tape volume is now available to be selected as a daily or spill backup volume, or as a target volume for data set migration, volume migration, or recycling of tape migration level 2 or dump volumes.
SCRATCHTAPE	Recycled tape volumes are deleted from DFSMSHsm control and become scratch tapes after DFSMSHsm recycles them. The tape volumes are not selected as tape backup or migration level 2 volumes, but an operator can mount them in response to a mount PRIVAT request for a tape backup or migration level 2 volume.

If you specify the global TAPEDELETION(SCRATCHTAPE | HSMTAPE) parameter, the parameter applies to all functions (migration, backup, and dump). If you specify a specific volume type, the parameter applies only to that type of function. If you specify a global value and a specific volume type in a single command, the command fails.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is SCRATCHTAPE.

Note:

1. Although the HSMBACKUPTAPE parameter has been changed to HSMTAPE, if you specify HSMBACKUPTAPE, it has the same meaning as HSMTAPE. The command will not fail.
2. If you choose SCRATCHTAPE and the tape security method that DFSMSHsm used is an expiration date, the DFSMSrmm VLPOOL EXPDTCHECK(N) option should be enough for DFSMSrmm to allow reuse with operator intervention and to allow relabelling to reinitialize the tape before DFSMSHsm or any other user tries to reuse the tape volume.
3. If you choose SCRATCHTAPE and the tape security method that DFSMSHsm used is a password, DFSMSrmm does not support password protection, so you would need to initialize the tapes to avoid problems. You can use the SECCLS parmlib options to specify the requirements to ERASE volumes DFSMSHsm uses before they return to scratch. You can automate this, see Chapter 16, “Initializing and Erasing Volumes” in [z/OS DFSMSrmm Implementation and Customization Guide](#).

4. If you do not use a tape management system, you choose SCRATCHTAPE, and the tape security method that DFSMSHsm used is password or expiration date, you must use the IEHINITT utility to reinitialize the tape before DFSMSHsm or any other user tries to reuse the tape volume.
5. If you choose HSM TAPE, you do not need to reinitialize the tapes that remain under DFSMSHsm control before you reuse them as tape backup or migration level 2 volumes.
6. The disposition of tapes automatically made available for reuse depends on the TAPEDELETION parameter of the SETSYS command. TAPEDELETION(HSM TAPE) is equivalent to issuing a DELVOL UNASSIGN for the tapes. TAPEDELETION(SCRATCHTAPE) is equivalent to issuing a DELVOL PURGE.

TAPEHARDWARECOMPACT and NOTAPEHARDWARECOMPACT: Specifying the data format used when writing to a 3480X tape device

Explanation: TAPEHARDWARECOMPACT | NOTAPEHARDWARECOMPACT are mutually exclusive, optional parameters that specify whether the Improved Data Recording Capability (IDRC) should be used when a 3480X is used as the output device.

IDRC is a form of data compaction that can improve storage capacity for most data sets. The amount of improvement depends upon the data contained within the data set.

TAPEHARDWARECOMPACT specifies that DFSMSHsm *will* request that IDRC be used on eligible 3480 tape cartridges. NOTAPEHARDWARECOMPACT specifies that DFSMSHsm *will not* use IDRC.

The compaction status assigned to a 3480 tape volume occurs at the time the volume is first used for output. This status is determined by the tape unit being used for output and the current specification of SETSYS TAPEHARDWARECOMPACT. DFSMSHsm can be directed to use the 3480X device, but not the IDRC feature by specifying the SETSYS NOTAPEHARDWARECOMPACT command. The following table summarizes how the compaction status is assigned to an empty 3480 tape volume:

SETSYS Parameter	Output Unit (3480)	Output Unit (3480X)	Output Unit (3490)	Output Unit (3590-1)
TAPEHWC	No compaction	Compaction	Compaction	Compaction
NOTAPEHWC	No compaction	No compaction	Compaction	Compaction

When output is restricted to a specific unit, the DFSMSHsm unit name may change. An empty 3480 cartridge can become a 3480X or 3490 cartridge containing compacted data. An empty 3480 or 3480X cartridge can become a 3490 cartridge containing compacted data.

When output is not restricted to a specific unit, DFSMSHsm selects the first available volume. There is no preference between a 3480, 3480X, or 3490 unit. The compaction status assigned to an empty tape volume is based on the following:

- The tape unit associated with the tape volume
- Whether or not 3480X or 3490 tape units are installed on the system
- The SETSYS command parameters TAPEHARDWARECOMPACT or NOTAPEHARDWARECOMPACT

The TAPEHARDWARECOMPACT parameter used with the following unit-restriction parameters causes control unit compaction to be requested:

- ABACKUP UNIT(*unittype*)
- ARECOVER TARGETUNIT(*unittype*)
- DEFINE DUMPCLASS(*class* UNIT(*unittype*))
- SETSYS ARECOVERUNITNAME(*unittype*)
- SETSYS ARECOVERML2UNIT(*unittype*)
- SETSYS ABARSUNITNAME(*unittype*)
- SETSYS BACKUP(TAPE(*unittype*))
- SETSYS CDSVERSIONBACKUP (BACKUPDEVICECATEGORY (TAPE(UNITNAME(*unittype*))))

- SETSYS MIGUNITNAME(*unittype*)
- SETSYS RECYCLEOUT(BACKUP(*unittype*))
- SETSYS SPILL(TAPE(*unittype*))
- SETSYS TAPEMIGRATION(DIRECT(TAPE(*unittype*)))
- SETSYS TAPEMIGRATION(ML2TAPE(TAPE(*unittype*)))
- SETSYS TAPEMIGRATION(NONE(ROUTETOTAPE(*unittype*)))
- SETSYS UNITNAME(*unittype*)

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either TAPEHARDWARECOMPACT or NOTAPEHARDWARECOMPACT, the DFSMSHsm default is NOTAPEHARDWARECOMPACT.

TAPEINPUTPROMPT: Specifying whether to suppress input tape messages

Explanation: TAPEINPUTPROMPT specifies whether DFSMSHsm should issue action messages to the operator asking if input tapes needed by DFSMSHsm are available. Each function using tapes (backup, migration, dump) is handled separately.

TAPEINPUTPROMPT(...TAPES(YES)) allows DFSMSHsm to continue normal operation, which is to issue action messages ARC0313A, ARC0314A, and ARC0366A. These messages prompt the operator to check that the tapes requested by DFSMSHsm are available.

Note: If all of the requested tapes are in an IBM automated tape library managed by SMS tape, messages ARC0313A, ARC0314A, and ARC0366A are not issued even though (...TAPES(YES)) has been specified.

TAPEINPUTPROMPT(...TAPES(NO)) allows you to suppress the input messages, if they are not necessary in your system configuration.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify TAPEINPUTPROMPT(...TAPES(NO)) for a category of tapes (backup, migration, dump), DFSMSHsm issues all messages related to input tapes for that category.

TAPELIBRARYPRESELECT: Preselecting output tape volumes in IBM Tape Library tape environment

Explanation: By default in an IBM Tape Library environment, DFSMSHsm waits to select an output tape until after the tape device has been allocated. This ensures the tape can be mounted and read on the allocated tape device and is resident in the same library as the allocated tape device. Because a tape volume is not selected at allocation time, the volser request at allocation time will be set to PRIVAT (scratch) and later update for open/mount. Some hardware features or configurations may require that the tape volume serial be provided at allocation time. This would require DFSMSHsm to first select the tape volume to which it will write prior to the tape device allocation.

This SETSYS parameter will allow users to have DFSMSHsm preselect tape volumes for output in an IBM Tape Library environment. If an existing tape is selected, its volser will be specified to the allocation.

The TAPELIBRARYPRESELECT (YES) option may not be necessary in environments where the following SETSYS commands are specified. In such an environment, all volumes DFSMSHsm requests for output will be SCRATCH volumes and thus the volser cannot be specified at the allocation time.

```
SELECTVOLUME(SCRATCH)
PARTIALTAPE(MARKFULL)
TAPEDELETION(SCRATCH)
DSBACKUP(TAPE(TASKS(0))) DASD(TASKS(n)))
```


Note: When configured to preselect output tape volumes, DFSMSHsm will no longer be able to verify that a tape volume is compatible with the tape drive that will be allocated nor that the tape volume is resident in the same library that contains the allocated tape drive. In environments where only one tape device type is written to by DFSMSHsm or where SMS is configured to direct allocations to appropriate libraries, such limitation will not impact DFSMSHsm's ability to use tape as expected.

SETSYS default: If you specify TAPELIBRARYPRESELECT without mentioning any sub parameter, the default is Y.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is N.

TAPEMAXRECALLTASKS: Specifying the maximum number of concurrent tape recall tasks

Explanation: **TAPEMAXRECALLTASKS(*tasks*)** is an optional parameter specifying the maximum number of tape recall tasks that can concurrently request tape mounts. For *tasks*, specify a decimal number from 1 to 15. The value you specify must be greater than 0 and less than or equal to the value you specify with the MAXRECALLTASKS parameter.

TAPEMAXRECALLTASKS is a subset of the MAXRECALLTASKS parameter. For example, if you specify MAXRECALLTASKS(10) and TAPEMAXRECALLTASKS(6), DFSMSHsm processes up to ten recall tasks concurrently, but only six can be recalls from tape migration level 2 volumes. You cannot specify a value for TAPEMAXRECALLTASKS that is greater than MAXRECALLTASKS; for example, MAXRECALLTASKS(10) and TAPEMAXRECALLTASKS(11).

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 1 tape recall task.

Note:

1. DFSMSHsm allows only a maximum of 15 recall tasks concurrently. Also, all recall tasks can process recalls from DASD migration volumes. For example, if you specify TAPEMAXRECALLTASKS(6) and fewer than six tape recall tasks are active, DFSMSHsm can process DASD recall tasks instead.
2. If you specify the same number for TAPEMAXRECALLTASKS as MAXRECALLTASKS, you may find that all tasks are waiting for tape drives and that many recalls from ML1 DASD are waiting in the queue. Specifying a value for TAPEMAXRECALLTASKS that is smaller than MAXRECALLTASKS in effect reserves a set of tasks to be always available to handle recalls from ML1 volumes.

TAPEMIGRATION: Specifying the type of migration to tape

Explanation

TAPEMIGRATION is an optional parameter specifying that you want DFSMSHsm to use migration level 2 (ML2) tape volumes, to limit the selection and allocation of an output volume during tape migration processing, and to attempt fast subsequent migration.

The following are required or optional parameters of the TAPEMIGRATION parameter:

DIRECT

Data sets on DFSMSHsm-managed primary volumes are migrated directly to ML2 tape volumes. If the primary volume is defined with the AUTOBACKUP attribute, data sets on the primary volume with the change flag set to 1 will not be migrated to an ML2 tape volume.

ML2TAPE

Data sets on ML1 volumes are migrated to ML2 tape volumes.

NONE

Data sets are not migrated to ML2 tape volumes unless migration to an ML2 tape volume is indicated by an external-to-DFSMSHsm factor. An ML2 tape volume is selected when:

- The ARCMDEXT exit is used to select migration to an ML2 tape volume from a primary volume (level 0).
- An SMS-managed data set that belongs to a management class indicating that the data set is to be migrated directly to an ML2 tape volume.

**ROUTETOTAPE
TAPE**

Specify a tape volume.

ANY

Any available tape volume is selected. If no tape volumes are available, then the value of the SETSYS MIGUNITNAME system setting is used to allocate a scratch tape.

unittype

An ML2 tape volume that is compatible with the specified unit type is selected. If no tape volumes of that unit type are available, then a scratch tape of that unit type is selected.

The following are valid types of tape units:

- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. If you specify:

- 3480 or 3480X, you use all the functions of the 3480 Magnetic Tape Subsystem.
- 3490, you use all the functions of the 3490 Magnetic Tape Subsystem.
- 3590-1, you use all the functions of the 3590 Magnetic Tape Subsystem.

RECONNECT

Data sets eligible for fast subsequent migration can be reconnected to the ML2 tape volume from which they were recalled. A data set can be reconnected if *all* fast subsequent migration requirements are satisfied. For more information about these requirements, see [Fast subsequent migration \(reconnection\) eligibility requirements](#).

NONE

Data sets are not eligible for fast subsequent migration and are not reconnected to the ML2 tape volume from which they were recalled.

ALL

All data sets that are eligible for standard migration and satisfy fast subsequent migration eligibility requirements are reconnected to the ML2 tape volume the data set was recalled from; including data sets that would have otherwise been migrated to an ML1 volume.

ML2DIRECTEDONLY

Data sets that are eligible for direct migration to ML2 tape volumes and satisfy fast subsequent migration eligibility requirements are reconnected to the ML2 tape volume the data set was recalled from. Data sets not eligible for direct migration to ML2 tape volumes are processed using the standard migration method.

SMS relationship

The TAPEMIGRATION parameter applies in some instances to SMS-managed DASD volumes or data sets. SMS-managed data sets with the LEVEL 1 DAYS NON-USAGE = NOLIMIT management class attribute are specifically prohibited from migrating to tape.

Tape library relationship

A tape unit name can be used for filtering in the ACS routines. If you specify a unit type, it will be sent to the ACS routines in SMS for the tape migration function.

SETSYS default

If you specify:

- DIRECT or ML2TAPE with no parameter, the default is TAPE(ANY).
- DIRECT or ML2TAPE with TAPE but no parameter, the default is ANY.
- NONE with no parameter, the default is ROUTETOTAPE(ANY).
- NONE with ROUTETOTAPE but no parameter, the default is ANY.
- RECONNECT with no parameter, the default is NONE.

DFSMSHsm default

If you do not specify the TAPEMIGRATION parameter on any SETSYS command, DFSMSHsm does not migrate data sets to ML2 tape volumes unless migration to an ML2 tape volume is indicated by an external-to-DFSMSHsm factor. Any tape volume is used. Fast subsequent migration is disabled. This is the equivalent to SETSYS TAPEMIGRATION(NONE(ROUTETOTAPE(ANY)) RECONNECT(NONE)).

TAPEOUTPUTPROMPT: Specifying whether to suppress output tape messages

Explanation: **TAPEOUTPUTPROMPT(TAPECOPY(YES | NO))** is an optional parameter that allows correct decisions by the operator when mounting a tape during TAPECOPY. It eliminates failures caused from mounting a standard capacity cartridge instead of an enhanced capacity cartridge or vice versa, since the operator does not always know which type of tape cartridge is required.

Subparameter	Explanation
YES	<p>DFSMSHsm issues a message to the operator during TAPECOPY processing indicating which type of target tape be mounted on a particular device; however, this message goes away after the tape has been mounted.</p> <p>Because using cartridge loaders for TAPECOPY causes the tape to be mounted by the cartridge load device before the operator verifies that the proper tape is mounted, an option to have the message issued for TAPEOUTPUTPROMPT as a write-to-operator-with-reply (WTOR) is available, allowing the operator to ensure that the proper tape is mounted even if cartridge load devices are being used. This option is available with a patch of the MCVT.</p> <p>For a description of how to patch the MCVT for this option, refer to Tuning DFSMSHsm in z/OS DFSMSHsm Implementation and Customization Guide.</p>
NO	<p>DFSMSHsm issues no message to the operator during TAPECOPY processing indicating which type of target tape is to be mounted on a particular device.</p>

SMS relationship: Parameter does not apply to SMS-managed DASD volumes or data sets.

SETSYS default: If you do not specify this parameter on the SETSYS command, the default is NO. If you do specify this parameter on the SETSYS command, a message is issued without requesting a reply when the option to patch the MCVT is not used. If the option to patch the MCVT is used, a message is issued requesting a reply.

TAPERECALLLIMITS: Specifying limits to continuously mounted ML2 tapes during multiple recalls

Explanation: **TAPERECALLLIMITS(TASK(*time*) TAPE(*time*))** is an optional parameter specifying the total time that a recall task can utilize a single tape mount for processing recall requests before criteria is used to possibly demount the tape. This parameter does not apply to tape devices prior to 3480 tape devices.

Suparameter	Explanation
TASK(<i>time</i>)	Allows you to specify the number of minutes that a DFSMSHsm recall task can process recall requests from a single tape mount before DFSMSHsm decides whether to demount the tape or to continue processing. Once that time is exhausted, each time the recall task completes the processing of its current recall request, DFSMSHsm decides whether to demount the tape or to continue processing. If the maximum number of tape recall tasks within the host are running and a higher priority tape recall request exists that can be attempted immediately, then the task is freed up to process other requests.
TAPE(<i>time</i>)	Allows you to specify the number of minutes that a recall task can process recalls from a single tape mount before additional criteria are considered. Once this time is exhausted, the recall task completes processing of its current recall request and then DFSMSHsm determines whether to demount the tape or to continue processing. If another DFSMSHsm host requires the mounted tape for a higher priority recall, then the tape is demounted. This results in a tape take-away from recall.

Once the recall processing stops for a specific recall task on a DFSMSHsm host, then all queued recall requests on that host that require the demounted tape volume are excluded from consideration for five minutes. This allows other hosts an opportunity to retry their delayed requests for the demounted tape volume.

When DFSMSHsm processes multiple recall requests that utilize one tape mount, the recall task processes the highest priority tape recall request which requires the currently mounted tape. If DFSMSHsm does force a demount of the tape volume, then the task and the tape are freed up to process other higher priority recalls.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either of the TAPERECALLLIMITS subparameters, then the default time used is 15 minutes for TASK and 20 minutes for TAPE.

Note:

1. TAPE(*time*) is longer than TASK(*time*) because TAPE(*time*) requires delaying additional queued requests.
2. The Storage Administrator is able to disable the TAPERECALLLIMIT function by setting both the TASK and the TAPE time to large values, such as 1000 or even 65535 minutes.

TAPESECURITY: Specifying the type of security for tape volumes

Explanation: **TAPESECURITY** is an optional parameter that you use to specify the type of security for your tape backup, migration level 2, and dump volumes. You can choose more than one tape security option. CDS backup tapes are not affected by this command.

Subparameter	Explanation
EXPIRATION	<p>DFSMSHsm protects each tape backup, migration level 2, and dump volume with an expiration date. Each backup version, migration copy, or dump copy on the tape volume is protected by an expiration date by placing an expiration date of 99365 in the IBM Standard Data Set Label 1 (HDR1, EOVS1, and EOF1). For backup or migration, however, you can use the tape data set exit, ARCTDEXT, to change the expiration date. You can change the expiration date of a dump tape volume using the TAPEEXPIRATIONDATE optional parameter of the DEFINE command.</p> <p>DFSMSHsm does not place a backup version or migration copy of a password-protected data set on a tape volume that is not password protected unless you specify the EXPIRATIONINCLUDE or RACFINCLUDE subparameter. Therefore, if the only tape security subparameter you specify is EXPIRATION, DFSMSHsm fails the backup or migration of a password protected data set to a tape backup or migration level 2 volume.</p>
EXPIRATIONINCLUDE	DFSMSHsm protects each tape backup, migration level 2, and dump volume with an expiration date. In addition, DFSMSHsm backs up or migrates a password-protected data set to a tape volume that is not password protected.
PASSWORD	DFSMSHsm protects each tape backup, migration level 2, and dump volume with a password. To password protect, place an X'F1' in the data set security byte in the IBM Standard Data Set Label (HDR1, EOVS1, and EOF1).
RACF	<p>DFSMSHsm protects with RACF each tape backup, migration level 2, and dump volume. DFSMSHsm protects the tape volume with RACF by adding it to a RACF tape volume set of DFSMSHsm (HSMHSM or DFHSMx). The <i>z/OS DFSMSHsm Implementation and Customization Guide</i> contains information about how to define RACF tape volume sets for DFSMSHsm and ABARS.</p> <p>If TAPESECURITY RACF RACFINCLUDE is specified, ABACKUP output tapes and ARECOVER input tapes are RACF protected, if RACF is active and if an ABARS tape volume set of HSMABR is defined.</p> <p>DFSMSHsm does not place a backup version or migration copy of a password-protected data set on a tape volume that is not password protected unless you specified the RACFINCLUDE or EXPIRATIONINCLUDE subparameter. Therefore, if you specify only the RACF subparameter, DFSMSHsm fails the backup or migration of a password-protected data set to a tape backup or migration level 2 volume.</p>
RACFINCLUDE	DFSMSHsm protects each tape backup, migration level 2, and dump volume with RACF. In addition, DFSMSHsm backs up or migrates a password-protected data set to a tape volume that is not password protected.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the default is PASSWORD.

Note:

1. Although the D parameter has been changed to EXPIRATION, if you specify D, it has the same meaning as EXPIRATION. If you specify DI, it has the same meaning as EXPIRATIONINCLUDE. If you specify the PD or PE parameter, it has the same meaning as if you specify both the PASSWORD parameter and the EXPIRATION parameter.
2. If no RACF tape volume sets are defined for DFSMSHsm, but all tapes are protected by an existing RACF generic profile, then SETSYS TAPESECURITY(RACF | RACFINCLUDE) should be specified.
3. If you want to specify RACF or RACFINCLUDE, your computing system must meet the following conditions:
 - A sufficient level of DFSMSdfp must be installed, or you must specify the PASSWORD tape security option.
 - RACF must be installed.
 - The RACF TAPEVOL resource class must be defined in the RACF class descriptor table (CDT).
 - If you do not have a sufficient level of DFSMSdfp installed and you want to RACF-protect your tape, you must specify PASSWORD with RACF in the TAPESECURITY parameter of the SETSYS command.
4. If RACF is inactive or the RACF TAPEVOL resource class is inactive, DFSMSHsm issues a message (ARC0099I).
5. The restriction about putting password-protected data sets on non password-protected tape volumes does not apply to dump volumes. Therefore, EXPIRATION and EXPIRATIONINCLUDE are equivalent options for dump processing, as are RACF and RACFINCLUDE.
6. VSAM password protected data sets are no longer supported, so DFSMSHsm checking for password protection has been removed. Thus, EXPIRATION and EXPIRATIONINCLUDE are equivalent options for migration and backup processing, as are RACF and RACFINCLUDE for VSAM data sets.

TAPESPANSIZE: Reducing tape volume spanning

Explanation: **TAPESPANSIZE(nnnn)** is an optional parameter used to reduce the occurrences of data sets spanning migration or backup tape volumes. For *nnnn*, specify a value between 0 and 9999 in units of megabytes (MB). This value represents the maximum number of megabytes of tape (ML2 or backup) that DFSMSHsm may leave unused while it tries to eliminate spanning of data sets. To state this differently, this value is the minimum size of a data set that is allowed to span tape volumes. Data sets whose size is less than the value do not normally span volumes. Only those data sets whose size is greater than or equal to the specified value are allowed to span volumes.

This parameter offers a trade-off: to reduce the occurrences of a user data set spanning tapes in exchange for some unused media at the end of each cartridge. The amount of unused media can vary from 0 to *nnnn* physical megabytes, but roughly averages 50% of the customer's median data set size. For example, if you specify 4000MB and your customer's median-sized data set is 2MB, then on average only 1MB of media is unused per cartridge.

If the SETSYS TAPEUTILIZATION parameter is set to NOLIMIT, DFSMSHsm makes no attempt to reduce data set tape volume spanning. If the SETSYS TAPEUTILIZATION parameter is set to a percent, then DFSMSHsm performs reduced data set spanning.

DFSMSHsm calculates a target data set size to use when it is determining whether a data set should fit entirely on the currently mounted tape volume. The size is calculated based on the used tracks of the data set and an estimated IDRC compaction accomplishment. If data is written to tape in IDRC format and software compaction is not being used (by either DFSMSHsm or CPU) for the data set, DFSMSHsm assumes a 2.5 to 1 compaction savings for the data set. If you want to use a different compaction assumption, see *Tuning DFSMSHsm in z/OS DFSMSHsm Implementation and Customization Guide*. If IDRC format is not used or software compaction is used for the data set, then DFSMSHsm does not consider any compaction accomplishments when it estimates the amount of tape media the data set requires.

DFSMSHsm recommends a value of 4000MB for all IBM 3490 and 3590 tape cartridges.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify TAPESPANSIZE, the default is 500MB.

TAPETYPE: Specifying whether to use WORM media for non-ABARS functions

Explanation: **TAPETYPE(WORM | NOWORM)** is an optional parameter that specifies that you want DFSMSHsm to use WORM media for non-ABARS functions.

The following are required or are optional parameters of the TAPEMIGRATION:

TAPETYPE parameter:

NOWORM

Specifies that DFSMSHsm should not use WORM media for non-ABARS functions.

WORM

Specifies that DFSMSHsm should allow non-ABARS functions to use WORM media.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: NOWORM.

DFSMSHsm default: If you do not specify the TAPETYPE parameter on any SETSYS command, DFSMSHsm does not use WORM media for non-ABARS functions by default.

Note:

To begin using WORM tape for all HSM functions, issue SETSYS TAPETYPE(WORM), otherwise, the default is NOWORM.

It is up to the Storage Administrator to decide whether using WORM media with DFSMSHsm functions is appropriate. These functions were not designed to be archival and so have some limitations users should consider before implementing them.

Since the support is passive, HSM treats WORM media the same way it treats R/W tapes.

Migration allows recall with subsequent remigration of data sets by using the same data set names. Even though the original form of a data set remains on tape, only the last version is known to DFSMSHsm.

Data sets on tape tend to move from one tape volume to another by using the Recycle function. If Recycle is allowed, there is no guaranteed relationship between a data set and tape volser.

If the storage administrator begins to use TAPETYPE(WORM) for all HSM functions and then later decides to change back to TAPETYPE(NOWORM), all WORM tapes must be marked full so they are not selected for output.

TAPEUTILIZATION: Specifying the desired tape utilization

Explanation: **TAPEUTILIZATION** is an optional parameter you can use to specify the desired tape utilization for a backup or migration tape volume written in single-file format. When a single-file format tape volume has reached the desired level of utilization (fullness), DFSMSHsm forces an end-of-volume (EOV). With this parameter, you can limit the amount of media used.

LIBRARYBACKUP is a mutually exclusive, required subparameter used to specify the desired tape utilization for a backup tape stored in an SMS-managed tape library. Because esoteric unit names are ignored in an SMS-managed tape library, this is the only vehicle by which you may limit the amount of media used in a library backup cartridge. PERCENTFULL(*pct*) and NOLIMIT are mutually exclusive, optional subparameters that can be used with LIBRARYBACKUP.

LIBRARYMIGRATION is a mutually exclusive, required subparameter used to specify the desired tape utilization for a migration tape stored in an SMS-managed tape library. Because esoteric unit names are ignored in an SMS-managed tape library, this is the only vehicle by which you may limit the amount

of media used in a library migration cartridge. PERCENTFULL(*pct*) and NOLIMIT are mutually exclusive, optional subparameters that can be used with LIBRARYMIGRATION.

UNITTYPE(*unittype*) is a mutually exclusive, required subparameter used to specify the desired tape utilization for a backup or migration tape written in single-file format outside of any library. PERCENTFULL(*pct*) and NOLIMIT are mutually exclusive, optional subparameters that can be used with UNITTYPE.

This command can be issued multiple times, once for each *unittype* required. You must issue this command for each unique unit type that needs a percentage different from the default. For example, if both 3480s and 3490s exist, issue a separate SETSYS TAPEUTILIZATION command. Issue a separate command for a different esoteric *unittype*. For example, if separate esoteric unit names are used for backup and migration, each one can have a different tape utilization value.

For *unittype*, specify one of the following unit names:

- 3480, which is a 3480 **without** the IDRC
- 3480X, which is a 3480 **with** the IDRC
- 3490
- 3590-1
- Any esoteric name that has been identified to DFSMSHsm through the SETSYS USERUNITTABLE command.

If NOLIMIT is specified, there is no tape limiting, no action is taken to reduce data set tape volume spanning and DFSMSHsm writes data until the end-of-volume is reached. It is recommended that you specify PERCENTFULL(100) on the SETSYS TAPEUTILIZATION command when reduced data set tape volume spanning is needed. Specifying NOLIMIT will disable reduced data set tape volume spanning.

CAPACITYMODE(COMPATIBILITY | EXTENDED) is an optional subparameter you can use to specify the maximum capacity to which a tape is filled. COMPATIBILITY specifies that an IBM 3590 that emulates an IBM 3490 fills a tape to no more than the capacity allowed by an IBM 3590 that emulates an IBM 3490 that is not operating in extended capacity mode. Data on these tapes is accessible to IBM 3590 drives that emulate IBM 3490 drives, regardless of whether they are CAPACITYMODE switchable. EXTENDED specifies that an IBM 3590 that emulates an IBM 3490 fills a tape to the same capacity as 3590 drives allow. Data on these tapes is accessible only to tape units that are CAPACITYMODE switchable.

The CAPACITYMODE subparameter is valid when all the following are true:

- CAPACITYMODE is used in conjunction with UNITTYPE
- The subparameter of UNITTYPE is an esoteric that is specified through SETSYS USERUNITTABLE
- The UNITTYPE contains only IBM 3590 drives that are emulating IBM 3490 drives that are CAPACITYMODE switchable.

When you specify CAPACITYMODE incorrectly, DFSMSHsm ignores the subparameter. For example, if you specify CAPACITYMODE when the UNITTYPE is a generic, such as 3490, DFSMSHsm ignores the CAPACITYMODE subparameter. Also, if you specify CAPACITYMODE for an esoteric capable of CAPACITYMODE(EXTENDED), but you do not specify COMPATIBILITY or EXTENDED, CAPACITYMODE is ignored.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: For TAPEUTILIZATION the DFSMSHsm default value is 97%. If you use the DFSMSHsm default value, the data on one cartridge can most likely be copied to another cartridge of the same category. If you do not plan to copy cartridges, you might specify NOLIMIT to use the entire cartridge capacity. However, it is recommended that you instead specify PERCENTFULL(100) to get the benefits of reduced tape spanning.

For TAPEUTILIZATION CAPACITYMODE, the DFSMSHsm default is COMPATIBILITY. During an input operation, DFSMSHsm temporarily switches the unit to CAPACITYMODE(EXTENDED) if it is given a tape in

that mode. If you do not specify CAPACITYMODE for an esoteric capable of CAPACITYMODE(EXTENDED), during a DFSMSHsm output function, the drives operate in CAPACITYMODE(COMPATIBILITY).

Note:

1. In a duplex tape environment, DFSMSHsm uses the default PERCENTFULL value of 97 percent. This value is not likely to be a problem unless you are targeting a 3590-Bx or vendor drive in 3490 emulation mode. These drives need a percent value of a few hundred to fully utilize the tape's capacity. A value of 97 percent is appropriate, however, for targeting a 3590-Ex drive, even if in 3490 emulation mode.
2. Issuing the SETSYS USERUNITTABLE command will reset the TAPEUTILIZATION for all esoteric devices to the DFSMSHsm default of 97 percent. For the command to be effective, you must enter the SETSYS USERUNITTABLE command before the SETSYS TAPEUTILIZATION command.
3. UNITTYPE(*unittype*), LIBRARYBACKUP, and LIBRARYMIGRATION are mutually exclusive subparameters. A separate SETSYS command must be issued for each subparameter. For example, to use 100% of the capacity but retain the reduced tape spanning, issue the following commands:

```
SETSYS TAPEUTILIZATION(UNITTYPE(unittype) PERCENTFULL(100))
SETSYS TAPEUTILIZATION(LIBRARYBACKUP PERCENTFULL(100))
SETSYS TAPEUTILIZATION(LIBRARYMIGRATION PERCENTFULL(100))
```

4. For drives prior to the IBM 3590 Model E1x, DFSMSHsm based the PERCENTFULL value on the cartridge type of the reported logical device type in emulation mode. For IBM 3590 Model E1x drives, DFSMSHsm uses the reported cartridge type of the physical device to determine the tape capacity. When using these tape drives, you do not need to supply a percentage greater than 100 to reasonably fill the tapes. As a result, DFSMSHsm has improved the ability to fill migration and backup tape cartridges when using IBM 3590 Model E1x drives.
5. If you have tape cartridges that have a capacity larger than the IBM enhanced-capacity tape cartridges, it is recommended that you specify the PERCENTFULL(*pct*) parameter for the unit type when those tapes will be used instead of specifying NOLIMIT. For *pct*, specify a value between 1 and 9999.

For example, if you have tape cartridges that are approximately 25% larger, using a value of PERCENTFULL(122) on the SETSYS TAPEUTILIZATION command is recommended for the following reasons:

- You want to avoid hitting the natural EOVS because that would tie tapes together and until the spanning data set is invalidated, all the tapes containing the desired data set must be handled to access the one spanning data set.
 - RECYCLE has to treat the connected tapes together until the spanning data set is invalidated. This process generally delays the tapes from being returned to scratch.
 - A chain of 240 connected tapes will likely be created if TAPEUTILIZATION is not used. If you have an error and need to take some explicit corrective action on one tape, you may have to process a very long chain of tapes.
 - The TAPECOPY function will probably be able to copy these tapes but only to similar or larger tapes. However, the use of 122% rather than 125% increases the possibility of a successful tape copy by maintaining an equivalent 97% tape utilization.
6. IBM 3590 drives introduce a new microcode support; see [Table 66 on page 1200](#) for the percent values to use with the SETSYS TAPEUTILIZATION command:

Table 66. Tape Utilization Capacity for MEDIA3 or Greater.		
Percent to use	3490 Mode	3590 Mode
3590B with MEDIA3 or MEDIA4	2200. If any of the 3590B drive controllers are running with a level of microcode prior to 1.9.19.0, use a value of 1100 instead.	97

Table 66. Tape Utilization Capacity for MEDIA3 or Greater. (continued)		
Percent to use	3490 Mode	3590 Mode
All others	97	97

7. DFSMSHsm writes to 97% of the capacity of MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13 tapes unless otherwise specified by the installation. Other percentages can be specified through the SETSYS TAPEUTILIZATION command, depending on the particular needs of the installation. DFSMSHsm uses the reported cartridge type on the physical device to determine the tape's capacity.
8. The 3490 architecture that the TS7700 emulates has maximum number of blocks, approximately 4.1 million (22 bits = 4,194,304) that can be written to a single tape volume. Writing beyond this limit results in an end of volume (EOV) condition. DFSMSHsm migration and backup processing write to tape using a 16k block size. Due to the 3490 block limit and the use of 16k blocks, if larger logical volume sizes such as 25GB or 65GB are defined for HSM use in the TS7700, the block limit may be reached prior to the full capacity of the volume being utilized.

UNITNAME: Specifying the type of unit for mounting a scratch tape during backup or dump

Explanation: **UNITNAME**(*unitname*) is an optional parameter identifying the type of unit that should be specified the first time DFSMSHsm requests that a scratch tape be mounted during backup or dump.

The BACKUP and DUMP functions use the same types of tape units. For *unitname*, substitute the type of tape unit.

The following are valid types of tape units:

- 3480
- 3480X
- 3490
- 3590-1

You can substitute an esoteric tape unit defined with the SETSYS USERUNITTABLE command. If you specify an esoteric tape unit name that does not exist in the user unit table, the command fails. If you specify:

- 3480 or 3480X, you use all the functions of the 3480 Magnetic Tape Subsystem.
- 3490, you use all the functions of the 3490 Magnetic Tape Subsystem.
- 3590-1, you use all the functions of the 3590 Magnetic Tape Subsystem.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is 3590-1.

Note:

1. For DUMP functions, if you want to specify a reel-type tape, you must use the DEFINE DUMPCCLASS command.
2. For the CDS Backup function, if you want to specify a reel-type tape, you must use the SETSYS CDSVERSIONBACKUP (BACKUPDEVICECATEGORY(TAPE(UNITNAME))) command.
3. You cannot specify an esoteric unit name that represents DASD; it must represent tape.
4. DFSMSHsm does not use UNITNAME when it requests that a scratch volume be mounted while continuing from another volume. Instead, DFSMSHsm uses the same unit the volume was mounted on.

UNLOAD and NOUNLOAD: Specifying whether to unload MSS virtual DASD volumes

The use of the UNLOAD and NOUNLOAD parameters are considered to be nonfunctional and their use is ignored by DFSMSHsm.

USECYLINDERMANAGEDSPACE: Specifying whether to allow use of cylinder managed space for migration copies and backup versions

Explanation: USECYLINDERMANAGEDSPACE is an optional parameter specifying whether you want to allow use of the cylinder-managed space for migration copies and backup versions. USECYLINDERMANAGEDSPACE can be abbreviated as USECMS.

The parameters of the USECYLINDERMANAGEDSPACE parameter are explained:

Y

specifies that migration copies and backup versions can reside in EAS. For allocation, DFSMSHsm uses EATTR=OPT for its migration copy or backup version.

N

specifies that migration copies and backup versions cannot reside in EAS. For allocation DFSMSHsm does not specify an EATTR value.

SMS relationship: This parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: N (no).

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is N (no).

Note:

1. When using the ARECOVER command, the specified USECYLINDERMANAGEDSPACE parameter will determine allocation of migrated data sets specified in the INCLUDE list.
2. When using the FREEVOL command, if USECYLINDERMANAGEDSPACE(Y) was specified and migration copies, backup versions, or VTOC copies are going to be moved, then EATTR=OPT is specified. Otherwise, an EATTR value is not specified.
3. When you issue the QUERY SETSYS command, the value of the USECYLINDERMANAGEDSPACE is displayed as USECMS=*value*.

USERUNITTABLE and NOUSERUNITTABLE: Specifying esoteric tape unit names to DFSMSHsm

The SETSYS USERUNITTABLE command has two functions:

- to identify esoteric tape unit names to DFSMSHsm.
- to control translation of esoteric output unit names.

The SETSYS USERUNITTABLE command must always be coded for esoteric unit names used by DFSMSHsm whether translation is required or not. All esoterics identified to DFSMSHsm with the SETSYS USERUNITTABLE command must appear in a single command.

To identify esoteric tape unit names to DFSMSHsm, you must first define these esoteric tape unit names to MVS during system I/O generation. Then, you must include the esoteric tape unit names in a DFSMSHsm SETSYS USERUNITTABLE command. Only after they have been successfully specified with the SETSYS USERUNITTABLE command can they be recognized and used as valid unit names with subsequent DFSMSHsm commands.

For the DFSMSHsm commands required to restrict device selection, see topic about [Implementing tape environments in z/OS DFSMSHsm Implementation and Customization Guide](#), specifically:

- For SMS-managed tape libraries, see the topic about [Information passed to an ACS routine](#).
- For other tape libraries (such as non-SMS or nonlibrary), see the topic about [Tape device selection](#).

To control the translation of output unit names, you can use the SETSYS USERUNITTABLE specifications of output unit names to input unit names. Translation occurs for translation-eligible esoteric groups. These include:

- 3480 device types with all online units of the esoteric group having cartridge loaders
- 3490 device types
- 3590 device types

Translation occurs during output processing so that when the tape units are used for input, the translated unit name is used for allocation.

There are three forms of specifications you can use to identify esoteric tape unit names to DFSMSHsm:

- first, translation is not specified (*es1*)
- second, translation is specified to a different unit name (*es2out:es2in*)
- third, translation is specified to the same unit name (*es3:es3*)

Explanation: When you specify USERUNITTABLE(*es1*, *es2out:es2in*, *es3:es3*), you are identifying four separate esoteric tape unit names in three independent specifications. You can use any of these forms of specification to identify esoteric tape unit names to DFSMSHsm. You can combine specifications as we have shown here, but remember that all esoterics used in another DFSMSHsm command must appear in a single SETSYS USERUNITTABLE command. Each separate form of specification is explained below.

When you specify a single esoteric (*es1*), the esoteric tape unit name *es1* is identified to DFSMSHsm as valid. Output allocations are made to units associated with the particular esoteric tape unit name requested for the output function. Any cartridge loader esoteric translation is assigned to the generic unit name equivalent to the type of units associated with the esoteric tape unit name. Substitute your esoteric name for *es1*.

When you specify an out-and-in esoteric pair (*es2out:es2in*), the esoteric unit names *es2out* and *es2in* are identified to DFSMSHsm as valid. Translation to an esoteric can take place. The esoteric unit name requested for the output allocation (*es2out*) may be translated to the esoteric tape unit name specified to the right of the colon (*es2in*) when output processing is completed. If translation occurs, then when the tape volume is requested to be read, input allocations are made to units associated with *es2in*. The purpose of this translation is to allow the user to use different devices for input than output. Substitute your esoteric names for *es2out* and *es2in*.

When you specify (*es3:es3*), the esoteric unit name *es3* is identified to DFSMSHsm as valid. Since the same esoteric unit name is specified for both input and output, the same units are candidates for input allocations and output allocations. Substitute your esoteric name for *es3*.

The following discussion presents two examples. One is very general and one is specific. Please note that any values specified here are only examples and should not be interpreted as the values to be used for your system.

General example

```
SETSYS USERUNITTABLE(es1, es2out:es2in, es3:es3)
```

If, for example, backup is restricted to *es1* by using the SETSYS BACKUP(TAPE(*es1*)) command, migration is restricted to *es2out* by using the SETSYS TAPEMIGRATION(ML2TAPE(TAPE(*es2out*))) command, and spill is restricted to *es3* by using the SETSYS SPILL(TAPE(*es3*)) command, then the following occurs:

- Backup output allocations are made to units associated with *es1*. If *es1* is translation-eligible, input allocations for recover are allocated using the generic unit name equivalent to *es1*.
- Migration output allocations are made to units associated with esoteric tape unit name *es2out*. If *es1* is translation-eligible, input allocations for recall are made to units associated with *es2in*.

- Dump output allocations are made to units associated with esoteric tape unit name `es3`, and input allocations for restore from the tapes written in the dumping process are also made to units associated with `es3`.

Specific example

```
USERUNITTABLE(SPOUT,DPOUT,MIGOUT:RECIN,BACKUP:RECIN)
```

This example specifies five esoteric tape unit names. When you specify single esoteric tape unit names (without a colon, like `SPOUT` and `DPOUT`), any translations that occur are allocated to generic tape units. When you specify paired esoteric tape unit names (with colon separators, like `MIGOUT:RECIN` and `BACKUP:RECIN`), any translations that occur are to the specified tape unit name (`RECIN` in this case) and that is what will be used during later input allocations.

`NOUSERUNITTABLE` specifies that no esoteric tape unit names are identified to DFSMSHsm. Any previously defined esoteric names are no longer known to DFSMSHsm.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

Tape Library Relationship: Translation can occur for SMS tapes. However, the ACS routines can modify the resulting tape devices used. Tapes are read back on drives available to the library where the tape is resident.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is `NOUSERUNITTABLE`.

Note:

1. If there is no unit online and available for the esoteric when the SETSYS USERUNITTABLE is issued, you will receive message ARC0103I indicating the USERUNITTABLE is invalid.
2. For IBM 3490 and 3590 tape devices, esoteric translation always occurs, regardless of the existence of cartridge loaders. If you specify a 3490 or 3590-1 esoteric unit name to direct output tape allocations, a translation of that esoteric unit name occurs. Subsequent input allocations use that translated unit name. Therefore for 3490 or 3590-1, if you specify an output esoteric unit name and want the same esoteric unit name to be used for that tape's input allocations, you must specify a translation to itself. For example, SETSYS USERUNITTABLE(TAPE:TAPE).
3. The 3490 control units (models A01, A02, A10, A20, D31, and D32) have the same read and write capabilities as the 3480X control unit.
4. You must ensure that each tape unit belonging to the esoteric group you specify can read and write on any tape written by any other tape unit that belongs to the same esoteric group. If the esoteric group contains both 3480 and 3480X devices, DFSMSHsm treats both types of devices as 3480s.
5. If an esoteric group includes one set of the tape drives defined with compaction on and another set defined with compaction off, this latter group of drives will not be able to read compacted data by the first group. Thus, the esoteric is considered invalid and message ARC0103I will be received when the SETSYS USERUNITTABLE command is issued.
6. You must define all esoteric tape unit names during system I/O generation. Do not remove from a subsequent system I/O generation those esoteric tape unit names DFSMSHsm already has used during its tape processing. If you do, DFSMSHsm cannot allocate the tape volume whose esoteric unit name you specified when you issued the ADDVOL command. Basically, by removing the esoteric name, you make the esoteric name an invalid unit name.
7. DFSMSHsm rejects mixed combinations of device names in an esoteric group, with the following exception: it allows the use of both 3480 and 3480X device names in a single group. Improved Data Recording Capability (IDRC), however, is not used with such a group because all devices in the group are treated as 3480s. If an esoteric group associated only with 3480Xs exists, 3480s should not be added to it, because the data already written using this esoteric tape unit name may create IDRC incompatibilities. 3592-1, 3592-2 and 3592-2E tape units are allowed into the same esoteric since

they share common write formats. 3592-2, 3592-2E, and 3592-3E tape units are allowed into the same esoteric since they also share common write formats. 3592-3E and 3592-4E tape units are allowed into the same esoteric since they also share common write formats. It is up to the user to ensure the drives in the mixed esoteric will be used with a common recording technology write using the same recording technology. For example, if any 3592-1 drives are included, then all drives in the esoteric must use EMFT1 for output. If the esoteric mixes 3592-2 and 3592-2E drives, then all drives must be setup to use EFMT1 or EFMT2. If the esoteric contains mixed 3592-3E and 3592-2E tape units, then all drives must be set up to use EFMT2 or EEFMT2, the write formats they have in common. If the esoteric contains mixed 3592-4E and 3592-3E tape units, then all drives must be set up to use EFMT3 or EEFMT3, the write formats they have in common.

If DFSMSHsm rejects an esoteric tape unit name, it does not reject the rest of the valid esoteric names specified in the USERUNITTABLE command. Those names are now recognized by DFSMSHsm as valid esoteric tape unit names. Each time you specify USERUNITTABLE, the valid esoteric tape unit names identified with this parameter replaces any esoteric tape unit names identified with a previous USERUNITTABLE parameter of the SETSYS command.

8. If you define both 3592-E05 and 3592-J1A to the same esoteric group, the E05 must use the EFMT1 recording format.
9. You can define a 3592-E05 or 3592-J1A to the 3590-1 generic group. If so, you must ensure drive compatibility or segmentation when using the 3590-1 in a non-SMS environment.
10. If an esoteric group includes a mixture of cartridge loader and non cartridge loader units, the esoteric name translation does not take place. It is important to note that for translation to occur, at least one unit associated with the esoteric tape unit name *es2out* must be varied online, and all online units associated with *es2out* must have cartridge loaders or be 3490s. The unit name used by DFSMSHsm for input is described in [Implementing tape environments in z/OS DFSMSHsm Implementation and Customization Guide](#).
11. If you reenter a SETSYS USERUNITTABLE command be aware that the prior table is entirely deleted and a new one is built from the new command and the MVS esoteric definitions. This will likely necessitate that you also enter the SETSYS TAPEUTILIZATION command because defaults are again in use. If you have removed an esoteric that is used in a tape output restriction specification, such as SETSYS BACKUP(TAPE(es1)), then you will also need to change those specifications.
12. To remove cartridge loader as a condition for device type 3480 esoteric unit name translation, refer to [Tuning DFSMSHsm in z/OS DFSMSHsm Implementation and Customization Guide](#).
13. For a summary of esoteric translation results, refer to the topic [Summary of esoteric translation results for various tape devices in z/OS DFSMSHsm Implementation and Customization Guide](#).
14. The esoteric values specified for certain SETSYS parameters cannot resolve to reel-type tapes. See Chapter 13, “Managing DFSMSHsm media,” on page 527 for details.

VERSIONS: Specifying the maximum number of backup versions to keep for any data set

Explanation: **VERSIONS(limit)** is an optional parameter specifying the maximum number of backup versions DFSMSHsm keeps for any one data set. The maximum number of backup versions is limited to either 29 versions or 100 versions based on the BCDS record length. The following values are the valid maximum allowable number of backup versions for different BCDS record lengths:

- Record length of 2040 to 6543 — 29 versions maximum
- Record length of 6544 or more — 100 versions maximum

For *limit*, substitute a decimal number from 0 to 100. A zero (0) specifies that DFSMSHsm does not create backup versions for any non-SMS-managed data set unless you specify the ALTERDS VERSIONS command.

SMS relationship: Parameter does not apply to SMS-managed DASD volumes or data sets.

SETSYS default: If you specify a value larger than the valid maximum allowable for that BCDS record length, the maximum value for that record length will be used. For example, if you specify SETSYS

VERSIONS(50) and you have a BCDS record length of 2040 to 6543, the maximum allowable number of backup versions for this record length, which is 29, will be used.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default for *limit* is 2.

Note:

1. The SETSYS VERSIONS parameter is used to specify the number of backup versions for all your non-SMS-managed data sets. An authorized user, however, can specify the VERSIONS parameter of the ALTERDS command to change that number of backup versions for a specific data set. An unauthorized user can specify the VERSIONS parameter of the HALTERDS command to change that number of backup versions for a specific data set with the same high-level qualifier as the unauthorized user.
2. For more information about cataloged and uncataloged data sets, see [“Backing up cataloged and uncataloged data sets with the same name” on page 307](#).
3. If the number of versions in the VERSIONS parameter is decreased, and the EXPIREBV command is executed prior to the time the next version is created, all excess versions—whether cataloged or uncataloged—are deleted.

VOLCOUNT: Specifying the method for determining volume count for SMS target data set allocations when performed by DFSMSdss

Explanation: VOLCOUNT(ANY | NONE) is an optional parameter that affects how DFSMSHsm will invoke DFSMSdss for the recall and recover of data sets moved by DFSMSdss. Refer to [z/OS DFSMSdss Storage Administration](#) for more specific information about the restrictions and limitations of the DFSMSdss VOLCOUNT parameter.

Subparameter	Explanation
ANY	DFSMSHsm passes the VOLCOUNT(ANY) parameter to DFSMSdss during the recall or recover of data moved by DFSMSdss. This results in the allocation of the target data set on as many volumes as required, to a maximum of 59.
NONE	The VOLCOUNT parameter is not passed to DFSMSdss. Issuing SETSYS VOLCOUNT(NONE) will reset the associated DFSMSHsm indicators and DFSMSHsm will operate as it did prior to the VOLCOUNT support being added. If, for example, VOLCOUNT(ANY) had been previously specified, issuing SETSYS VOLCOUNT(NONE) resets the associated DFSMSHsm indicators.

The VOLCOUNT parameters of ANY or NONE are ignored if specified for data types to which they do not apply. Refer to the [z/OS DFSMSHsm Implementation and Customization Guide](#) to determine data set types moved by DFSMSdss for DFSMSHsm.

SMS relationship: Parameter applies in some instances to SMS-managed DASD volumes or data sets.

SETSYS default: There is no SETSYS default.

DFSMSHsm default: If you do not specify SETSYS VOLCOUNT, the default is NONE.

VOLUMEDUMP: Specifying concurrent copy options for dump volumes

Explanation: VOLUMEDUMP is an optional parameter specifying whether DFSMSHsm will request that DFSMSdss use a type of Concurrent Copy function for dump volumes or no Concurrent Copy function for dump volumes.

Subparameter	Explanation
CC	DFSMSHsm specifies Concurrent Copy when DFSMSdss dumps volumes. DFSMSdss determines if the device and microcode can support this operation. Customers can specify SETSYS DUMPVOLUME(CC). CC is an alias of PREFERRED and can be added next to the word PREFERRED like the other aliases.
STANDARD (NOCC)	STANDARD indicates Concurrent Copy will not be used when DFSMSHsm dumps the volumes. Control returns to the requestor at the end of the physical copy.
REQUIRED (ANYREQUIRED)	DFSMSHsm requests DFSMSdss to use concurrent copy to perform the dump. If virtual concurrent copy is unavailable DFSMSdss uses cache-based concurrent copy. If both types of concurrent copy are unavailable, the dump fails.
PREFERRED (ANYPREFERRED)	DFSMSHsm requests DFSMSdss to use concurrent copy to perform the dump. If virtual concurrent copy is unavailable, DFSMSdss uses cache-based concurrent copy. If both types of concurrent copy are unavailable, the dump continues as if the CC keyword was not specified.
VIRTUALREQUIRED (VREQ)	DFSMSHsm requests DFSMSdss to use virtual concurrent copy to dump volumes. If virtual concurrent copy is unavailable, the dump fails.
VIRTUALPREFERRED (VPREF)	DFSMSHsm requests DFSMSdss to use virtual concurrent copy to dump volumes. If virtual concurrent copy is unavailable, the dump continues as if the CC keyword was not specified.
CACHEREQUIRED (CREQ)	DFSMSHsm requests DFSMSdss to use cache-based concurrent copy to dump the data sets. If cache-based concurrent copy is unavailable, the dump fails.
CACHEPREFERRED (CPREF)	DFSMSHsm requests DFSMSdss to use cache-based concurrent copy to dump the data sets. If cache-based concurrent copy is unavailable, the data set dump continues as if the CC keyword was not specified.

Note:

1. The correct level of hardware and microcode must be present in your system environment for the Concurrent Copy function to work.
2. In a mixed hardware environment, where some volumes are capable of Concurrent Copy and others are not, specifying a type of Concurrent Copy will be accepted and only apply to those volumes capable of this process, and the noncapable volumes will be dumped using traditional methods.

SMS relationship: Parameter has the same meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify this parameter on any SETSYS command, the DFSMSHsm default is NOCC.

ZCOMPRESS: Specifying when compression with zEDC should be done

Explanation: **ZCOMPRESS** is an optional parameter set that you use to specify the type of compression used during migration or backup for all data sets.

The subparameters of the ZCOMPRESS parameter follow this discussion in alphabetical order. The following are optional subparameters:

- ALL | NONE
- CLOUDMIGRATE(YES | NO)
- DASDBACKUP (YES | NO)
- DASDMIGRATE (YES | NO)
- TAPEBACKUP (YES | NO)
- TAPEMIGRATE (YES | NO)

SMS relationship:

Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets. This ZCOMPRESS parameter is used only for SMS data sets that are not compressed on LO; for non-SMS data sets, it is always in effect, as you cannot have a non-SMS data set compressed on LO.

SETSYS default: If you do not specify the ZCOMPRESS parameter, the SETSYS default is no compression with zEDC.

DFSMShsm default: If you do not specify a subparameter with this parameter on any SETSYS command, the DFSMShsm default is no compression with zEDC.

ZCOMPRESS (NONE | ALL): Specifying whether to use zEDC Services for data sets

Explanation: ALL | NONE are mutually exclusive, optional subparameters of the ZCOMPRESS parameter, specifying whether or not to use zEDC Services during migration or backup.

Subparameter	Explanation
NONE	DFSMShsm does <i>not</i> use zEDC for a data set during migration or backup. DFSMShsm will use the SETSYS COMPACT settings.
ALL	DFSMShsm will use zEDC during migration and backup, if this function is supported by the system. If zEDC Services are <i>not</i> available, DFSMShsm will use the SETSYS COMPACT settings to determine what type of software compaction is used for migration and backup.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMShsm default: If you do not specify any subparameter of the ZCOMPRESS parameter on any SETSYS command, the DFSMShsm default is NONE and the SETSYS COMPACT settings will be used.

Note:

1. DFSMShsm does *not* use zEDC for partitioned data sets.
2. When you specify the ALL or NONE subparameter with the ZCOMPRESS parameter, DFSMShsm ignores all other subparameters.

ZCOMPRESS(CLOUDMIGRATE(NO | YES)): Specifying whether to use CDA compression for data sets during migration to DIRECT cloud connections

Explanation: CLOUDMIGRATE (NO | YES) are mutually exclusive, optional subparameters of the ZCOMPRESS parameter, specifying whether or not to use CDA compression with zEDC Services when DFSMShsm migrates to DIRECT cloud.

Subparameter	Explanation
CLOUDMIGRATE(NO)	DFSMSHsm does not use CDA compression for migration to DIRECT cloud connections.
CLOUDMIGRATE(YES)	DFSMSHsm will use CDA compression with zEDC Services during migration to DIRECT cloud connections, if this function is supported by the system. If CDA compression is unavailable, the data sets will not be compressed during migration to DIRECT cloud connections.

SMS relationship: SMS-managed data sets are supported for migration to CLOUD. This parameter is inapplicable for non-SMS-managed data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS ZCOMPRESS command, the DFSMSHsm default is CLOUDMIGRATE(NO).

Note:

1. As compressed-format data sets are already compressed, they are not compressed with CDA compression during migration to DIRECT cloud connections.
2. Since encrypted data sets do not compress, CDA compression will not be used during migration to DIRECT cloud connections.
3. You can run the SETSYS ZCOMPRESS(CLOUDMIGRATE(NO|YES)) command in the ARCCMDxx parmlib member conditionally by specifying ONLYIF HSMHOST. This allows you to run the command only on designated DFSMSHsm hosts.

Related Reading: For more information about the ONLYIF command, see [Chapter 56, “ONLYIF command: Running commands in ARCCMDxx conditionally,”](#) on page 973

ZCOMPRESS (DASDBACKUP (NO | YES)) : Specifying whether to use zEDC Services for data sets during backup to DASD

Explanation: DASDBACKUP (NO | YES) are mutually exclusive, optional subparameters of the ZCOMPRESS parameter, specifying whether or not to use zEDC Services when DFSMSHsm backs up to DASD.

Subparameter	Explanation
DASDBACKUP (NO)	DFSMSHsm does <i>not</i> use zEDC for a data set backup to DASD. DFSMSHsm will use the SETSYS COMPACT settings.
DASDBACKUP (YES)	DFSMSHsm will use zEDC Services during backup to DASD, if this function is supported by the system. If zEDC is <i>not</i> available, DFSMSHsm will use the SETSYS COMPACT settings to determine what type of software compaction is used for backup.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is DASDBACKUP(NO).

Note:

1. If you do not want a specific data set to be compressed with zEDC during volume backup to DASD, use the data set backup exit (ARCBDEXT) to prevent compression.
2. As compressed-format data sets are already compressed, they are not compressed with zEDC Services during backup to DASD.

3. Since encrypted data sets do not compress, zEDC compression services will not be used during backup to DASD.

ZCOMPRESS (DASDMIGRATE (NO | YES)) : Specifying whether to use zEDC Services for data sets during migration to DASD

Explanation: **DASDMIGRATE (NO | YES)** are mutually exclusive, optional subparameters of the ZCOMPRESS parameter, specifying whether or not to use zEDC Services when DFSMSHsm migrates to DASD.

Subparameter	Explanation
DASDMIGRATE (NO)	DFSMSHsm does <i>not</i> use zEDC for a data set migration to DASD. DFSMSHsm will use the SETSYS COMPACT settings.
DASDMIGRATE (YES)	DFSMSHsm will use zEDC Services during migration to DASD, if this function is supported by the system. If zEDC is <i>not</i> available, DFSMSHsm will use the SETSYS COMPACT settings to determine what type of software compaction is used for migration.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is DASDMIGRATE(NO).

Note:

1. If you do not want a specific data set to be compressed with zEDC during volume migration to DASD, use the data set migration exit (ARCMDEXT) to prevent compression.
2. As compressed-format data sets are already compressed, they are not compressed with zEDC Services during migration to DASD.
3. Since encrypted data sets do not compress, zEDC compression services will not be used during migration to DASD.

ZCOMPRESS (TAPEBACKUP (NO | YES)) : Specifying whether to use zEDC Services for data sets during backup to tape

Explanation: **TAPEBACKUP (NO | YES)** are mutually exclusive, optional subparameters of the ZCOMPRESS parameter, specifying whether or not to use zEDC Services when DFSMSHsm backs up to tape.

Subparameter	Explanation
TAPEBACKUP (NO)	DFSMSHsm does <i>not</i> use zEDC for a data set backup to tape. DFSMSHsm will use the SETSYS COMPACT settings.
TAPEBACKUP (YES)	DFSMSHsm will use zEDC Services during backup to tape, if this function is supported by the system. If zEDC is <i>not</i> available, DFSMSHsm will use the SETSYS COMPACT settings to determine what type of software compaction is used for backup.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is TAPEBACKUP(NO).

Note:

1. If you do not want a specific data set to be compressed with zEDC during volume backup to tape, use the data set backup exit (ARCBDEXT) to prevent compression.
2. As compressed-format data sets are already compressed, they are not compressed with zEDC Services during backup to tape.
3. Since encrypted data sets do not compress, zEDC compression services will not be used during migration to DASD.

ZCOMPRESS (TAPEMIGRATE (NO | YES)) : Specifying whether to use zEDC Services for data sets during migration to tape

Explanation: TAPEMIGRATE (NO | YES) are mutually exclusive, optional subparameters of the ZCOMPRESS parameter, specifying whether or not to use zEDC Services when DFSMSHsm migrates to tape.

Subparameter	Explanation
TAPEMIGRATE (NO)	DFSMSHsm does <i>not</i> use zEDC for a data set migration to tape. DFSMSHsm will use the SETSYS COMPACT settings.
TAPEMIGRATE (YES)	DFSMSHsm will use zEDC Services during migration to tape, if this function is supported by the system. If zEDC is <i>not</i> available, DFSMSHsm will use the SETSYS COMPACT settings to determine what type of software compaction is used for migration.

SMS relationship: Parameter differs in meaning when applied to SMS-managed or non-SMS-managed DASD volumes or data sets.

SETSYS default: None.

DFSMSHsm default: If you do not specify either subparameter on any SETSYS command, the DFSMSHsm default is TAPEMIGRATE(NO).

Note:

1. If you do not want a specific data set to be compressed with zEDC during volume migration to tape, use the data set migration exit (ARCMDEXT) to prevent compression.
2. As compressed-format data sets are already compressed, they are not compressed with zEDC Services during migration to tape.
3. Since encrypted data sets do not compress, zEDC compression services will not be used during migration to tape.

Examples of how to code the SETSYS command

The following examples present common user scenarios that you can use to code the SETSYS command for your installation. Both main functions and supporting functions are illustrated in the examples. The functions are arranged in alphabetical order.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Specifying the SETSYS parameters for ABARS processing

In this example, the SETSYS parameters associated with both ABACKUP and ARECOVER processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

For ABARS Processing:

- All DFSMSdss messages are intercepted and written in the ABARS activity log. Parameter used is ABARSACTLOGMSGLVL(FULL).
- The ABARS activity log is written to DASD. Parameter used is ABARSACTLOGTYPE(DASD).
- Allow ABARS to use the default number of I/O buffers. Parameter used is ABARSBUFFERS(1).
- DFSMSHsm is to delete the ABARS activity log during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing. Parameter used is ABARSDELETEACTIVITY(Y).
- Allow ABARS to use the default base procedure name when starting the ABARS secondary address space. Parameter used is ABARSPROCNAME(DFHSMABR).
- Allow DFSMSHsm to stack the ABACKUP output. Parameter used is ABARSTAPES(STACK).
- Use 3590-1 as a unit name for ABARS. Parameter used is ABARSUNITNAME(3590-1).
- Use the default number of concurrent secondary addresses supported by DFSMSHsm. Parameter used is MAXABARSADDRESSSPACE(1).
- The following DFSMSHsm installation exits are active: ABARS expiration date setting exit, aggregate backup error exit, aggregate backup ML2 bypass exit, aggregate recovery conflict resolution exit, aggregate recovery bypass exit, and ABARS tape volume exit. Parameter used is EXITON(ED BE M2 CR SK TV).

For ABACKUP Processing:

- Use the default OPTIMIZE value when ABACKUP invokes DFSMSdss to backup level 0 DASD data sets. Parameter used is ABARSOPTIMIZE(3).
- Data sets on volumes protected by peer-to-peer remote copy or extended remote copy are backed up by ABACKUP processing. Parameters used are ABARSKIP(NOPPRC NOXRC).

For ARECOVER Processing:

- Allow DFSMSHsm to pass the VOLCOUNT parameter to DFSMSdss during the ARECOVER of L0 data sets. You want to allocate as many volumes as required for the target data set. Parameter used is ABARSVOLCOUNT(ANY).
- Use 3590-1 tape units when recovering user tape volumes. Parameter used is ARECOVERUNITNAME(3590-1).
- Use 3590-1 tape units when recovering migrated data sets to ML2 tape volumes. Parameter used is ARECOVERML2UNIT(3590-1).
- Use the default percentage of DASD space when filling non-SMS-managed recovery volumes. Parameter used is ARECOVERPERCENTUTILIZED(80).
- Allow DFSMSHsm to pass the TGTGDS parameter to DFSMSdss during restore processing of SMS-managed generation data sets to level 0 DASD. Parameter used is ARECOVERTGTGDS(SOURCE).

```

SETSYS ABARSACTLOGMSGLVL(FULL)  +
       ABARSACTLOGTYPE(DASD)    +
       ABARSBUFFERS(1)          +
       ABARSDELETEACTIVITY(Y)   +
       ABARSPROCNAME(DFHSMABR)  +
       ABARSTAPES(STACK)        +
       ABARSUNITNAME(3590-1)    +
       MAXABARSADDRESSSPACE(1)  +
       EXITON(ED BE M2 CR SK TV) +
       ABARSOPTIMIZE(3)         +
       ABARSKIP(NOPPRC NOXRC)   +
       ABARSVOLCOUNT(ANY)      +
       ARECOVERUNITNAME(3590-1) +
       ARECOVERML2UNIT(3590-1)  +
       ARECOVERPERCENTUTILIZED(80) +
       ARECOVERTGTGDS(SOURCE)

```


Specifying the SETSYS parameters for backup processing

In this example, the SETSYS parameters associated with backup processing are specified. Where appropriate, default values or recommended values are used in the examples. These parameters are in effect until another SETSYS command is issued to change them.

- Automatic backup can start between 0030 (12:30 a.m.) and 0400 (4:00 a.m.) hours. DFSMSHsm does not begin on additional volumes after 0730 (7:30 a.m.) hours. Parameter used is AUTOBACKUPSTART(0030 0400 0730).
- If the backup processing finds a data set in use, DFSMSHsm is to retry the backup after a delay of 60 minutes, and accept the backup version even if the data set is still in use. Parameter used is BACKUP(INUSE (RETRY(Y) DELAY(60) SERIALIZATION(PREFERRED))).
- DFSMSHsm can compact a data set during backup processing to DASD. Parameter used is COMPACT(DASDBACKUP).
- You want DFSMSHsm to use the backup prefix name of HSM. Parameter used is BACKUPPREFIX(HSM).
- DFSMSHsm does not compact a data set unless 40% of the total space is saved. Parameter used is COMPACTPERCENT(40).
- DFSMSHsm creates original backup tapes that will be kept on site and alternate backup tapes that will be written at a remote site. Parameter used is DUPLEX(BACKUP(Y)).
- The following installation exits are active: data set backup exit and space management and backup exit. Parameter used is EXITON(BD SA).
- DFSMSHsm creates backup versions when data sets are changed. Parameter used is INCREMENTALBACKUP(CHANGEDONLY).
- DFSMSHsm can backup eight DASD volumes concurrently on this host. Parameter used is MAXBACKUPTASKS(8).
- DFSMSHsm retains two tape allocations for 120 minutes of inactivity, until command data set backups directed to tape are held, or until DFSMSHsm shuts down before deallocating the drives. Parameter used is DSBACKUP(TAPE(DEMOUNTDELAY(MAXIDLETASKS(2) MINUTES(120)))).

```
SETSYS AUTOBACKUPSTART(0030 0400 0730) +
BACKUP(INUSE (RETRY(Y) DELAY(60) SERIALIZATION(PREFERRED))) +
BACKUPPREFIX(HSM) +
COMPACT(DASDBACKUP) +
COMPACTPERCENT(40) +
DUPLEX(BACKUP(Y)) +
EXITON(BD SA) +
INCREMENTALBACKUP(CHANGEDONLY) +
MAXBACKUPTASKS(8) +
DSBACKUP(TAPE(DEMOUNTDELAY(MAXIDLETASKS(2) MINUTES(120)))))
```

Specifying the SETSYS parameters for DFSMSHsm control data sets

In this example, the SETSYS parameters associated with DFSMSHsm control data sets processing are specified. Where appropriate, default values or recommended values are used in the examples. These parameters are in effect until another SETSYS command is issued to change them.

- DFSMSHsm invokes the DFSMSdss logical dump function for each control data set backed up. Parameter used is CDSVERSIONBACKUP(DATAMOVER(DSS)).
- DFSMSHsm backs up control data sets and the journal data set to scratch tapes. Parameter used is CDSVERSIONBACKUP(BACKUPDEVICECATEGORY(TAPE)).
- DFSMSHsm keeps twenty backup copies for each control data set and journal data set. Parameter used is CDSVERSIONBACKUP(BACKUPCOPIES(20)).
- You want DFSMSHsm to use HSM.BCDS.BACKUP as the initial characters when allocating the backup data set for the BCDS. Parameter used is CDSVERSIONBACKUP(BCDSBACKUPPSN(HSM.BCDS.BACKUP)).

- You want DFSMSHsm to use HSM.MCDS.BACKUP as the initial characters when allocating the backup data set for the MCDS. Parameter used is CDSVERSIONBACKUP(MCDSBACKUPDSN(HSM.MCDS.BACKUP)).
- You want DFSMSHsm to use HSM.OCDS.BACKUP as the initial characters when allocating the backup data set for the OCDS. Parameter used is CDSVERSIONBACKUP(OCDSBACKUPDSN(HSM.OCDS.BACKUP)).
- You want DFSMSHsm to use HSM.JRNL.BACKUP as initial characters when allocating the backup data set for the journal data set. Parameter used is CDSVERSIONBACKUP(JRNLBACKUPDSN(HSM.JRNL.BACKUP)).
- The control data set backup exit is active. Parameter used is EXITON(CB).
- No startup, space use, or data set messages appear at the system console. Parameter used is MONITOR(NOSTARTUP NOSPACE NOVOLUME).
- DFSMSHsm writes the BCDS, MCDS, and OCDS data set records in the journal data set when updated. Parameter used is JOURNAL(RECOVER).

```
SETSYS CDSVERSIONBACKUP(DATAMOVER(DSS) +
                        BACKUPDEVICECATEGORY(TAPE) +
                        BACKUPCOPIES(20) +
                        BCDSBACKUPDSN(HSM.BCDS.BACKUP) +
                        MCDSBACKUPDSN(HSM.MCDS.BACKUP) +
                        OCDSBACKUPDSN(HSM.OCDS.BACKUP) +
                        JRNLBACKUPDSN(HSM.JRNL.BACKUP)) +
                        EXITON(CB) +
                        MONITOR(NOSTARTUP NOSPACE NOVOLUME)+
                        JOURNAL(RECOVER)
```

Specifying the SETSYS parameters for dump processing

In this example, the SETSYS parameters associated with dump processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- Automatic dump can start between 0100 (1:00 a.m.) and 0400 (4:00 a.m.) hours. DFSMSHsm does not begin additional volumes after 0630 (6:30 a.m.). Parameter used is AUTODUMPSTART(0100 0400 0630).
- You intend to use the following DFSMSHsm functions: backup, dump, recovery, and expiring backup versions. Parameter used is BACKUP.
- You want DFSMSHsm to use HSM as the first qualifier in the dump copy name. Parameter used is BACKUPPREFIX(HSM).
- DFSMSHsm specifies to DFSMSdss that the fourth buffering technique is used for DFSMSdss DASD I/O for a full-volume dump. The fourth buffering technique has DFSMSdss read one cylinder at a time. DFSMSHsm specifies to DFSMSdss that the third buffering technique is used for DFSMSdss DASD I/O, and DFSMSdss is to read five tracks at a time. Parameter used is DUMPIO(4,3).
- DFSMSHsm can concurrently process two volume dump tasks. Parameter used is MAXDUMPTASKS(2).
- DFSMSHsm requests that DFSMSdss use the Concurrent Copy function for dump volumes because they have adequate cache for the changed data at the time volume dumps are running. Parameter used is VOLUMEDUMP(PREFERRED). DFSMSdss prefers to use virtual concurrent copy if available. If virtual concurrent copy is not available cache-based concurrent copy is attempted. If both types of concurrent copy are unavailable, the data set dump continues as if the Concurrent Copy keyword was not specified.
- The DFSMSHsm host is connected to the XCF group ARCDMP1 for common dump queue processing. This host is eligible to be the master scheduler for that functional group. All available dump tasks in the HSMplex hosts that are connected to the group as members can be used to process dump copy requests as they are received on this local host. The parameter used is COMMONQUEUE(DUMP(CONNECT(DMP1))).


```
SETSYS AUTODUMPSTART(0100 0400 0630) +
      BACKUP +
      BACKUPPREFIX(HSM) +
      DUMPIO(4,3) +
      MAXDUMPTASKS(2) +
      VOLUMEDUMP(PREFERRED)
      COMMONQUEUE(DUMP(CONNECT(DMP1)))
```

Specifying the SETSYS parameters for expiration processing

In this example, the SETSYS parameter associated with expiration processing is specified. Where appropriate, default values or recommended values are used in the example. This parameter is in effect until another SETSYS command is issued to change it.

- DFSMSHsm scratches data sets that have an expired expiration date when performing space management and migration cleanup. Parameter used is EXPIREDDATASETS(SCRATCH).

```
SETSYS EXPIREDDATASETS(SCRATCH)
```

Specifying the SETSYS parameters for extent reduction processing

In this example, the SETSYS parameter associated with extent reduction processing is specified. Where appropriate, default values or recommended values are used in the example. This parameter is in effect until another SETSYS command is issued to change it.

- DFSMSHsm reallocates all non-VSAM data sets that equal or exceed five extents. Parameter used is MAXEXTENTS(5).

```
SETSYS MAXEXTENTS(5)
```

Specifying the SETSYS parameters for fast replication processing

In this example, the SETSYS parameters that are associated with fast replication processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- The maximum number of DFSMSdss invocations for each FRBACKUP command that DFSMSHsm can process concurrently is 15.
- The maximum number of DFSMSdss invocations for each FRRECOV command that DFSMSHsm can process concurrently is 15.
- The maximum number of volume pairs that DFSMSHsm passes to each DFSMSdss invocation concurrently for backup and recover functions is 24.

```
SETSYS MAXCOPYPOOLTASKS(FRBACKUP(15) FRRECOV(15) DSS(24))
```

Specifying the SETSYS parameters for interval migration processing

In this example, the SETSYS parameters associated with interval migration processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- The space management volume exit is active. Parameter used is EXITON(MV).
- DFSMSHsm does interval migration if the occupied space on a primary volume is equal to or exceeds a high threshold level when the periodic hourly space check is performed. Parameter used is INTERVALMIGRATION.
- DFSMSHsm can concurrently process eight automatic interval migration tasks. Parameter used is MAXINTERVALTASKS(8).


```
SETSYS EXITON(MV) +
      INTERVALMIGRATION +
      MAXINTERVALTASKS(8)
```

Specifying the SETSYS parameters for DFSMSHsm logging and diagnosis

In this example, the SETSYS parameters associated with DFSMSHsm logging and diagnosis are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- DFSMSHsm generates and logs messages for all activities. Parameter used is ACTLOGMSGVL(FULL).
- DFSMSHsm dynamically allocates DASD data sets as the type of output data sets to receive messages. Parameter used is ACTLOGTYPE(DASD).
- DFSMSHsm is not operating in debug mode. Parameter used is NODEBUG.
- DFSMSHsm is not operating in emergency mode. Parameter used is NOEMERGENCY.
- Migration cleanup deletes the MCDS data set records for data sets that are recalled more than 10 days ago and deletes daily and volume statistics records that were created more than 90 days ago. The records cleaned up by MIGRATIONCLEANUP are used by the REPORT command and by DCOLLECT. Parameter used is MIGRATIONCLEANUP(10 90).
- No startup, space use, or data set messages appear at the system console. Parameter used is MONITOR(NOSTARTUP NOSPACE NOVOLUME).
- The problem determination aid is enabled. Parameter used is PDA.
- DFSMSHsm writes SMF records that contain DFSMSHsm statistics. Parameter used is SMF(240).
- The output from DFSMSHsm goes to system output class D, with one copy printed on a form identified by 6LPI. Parameter used is SYSOUT(D 1 6LPI).
- DFSMSHsm dumps are written to a system dump data set. Parameter used is SYS1DUMP.
- DFSMSHsm will use a fast replication message data set.

```
SETSYS ACTLOGMSGVL(FULL) +
      ACTLOGTYPE(DASD) +
      NODEBUG +
      NOEMERGENCY +
      MIGRATIONCLEANUPDAYS(10 90) +
      MONITOR(NOSTARTUP NOSPACE NOVOLUME) +
      PDA +
      SMF(240) +
      SYSOUT(D 1 6LPI)+
      SYS1DUMP
      FASTREPLICATION(MESSAGE DATASET(YES HLQ(MYHLQ)))
```

Specifying the SETSYS parameters for migration processing

In this example, the SETSYS parameters associated with migration processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- DFSMSHsm can compact data sets during migration processing to DASD. Parameter used is COMPACT(DASDMIGRATE).
- DFSMSHsm does not compact a data set unless 40% of the total space is saved. Parameter used is COMPACTPERCENT(40).
- The following installation exits are active: data set migration exit, second level migration data set exit, space management volume exit, and space management and backup exit. Parameter used is EXITON(MD MM MV SA).
- DFSMSHsm deletes data sets that have an expired expiration date. Parameter used is EXPIREDDATASETS(SCRATCH).

- DFSMSShsm runs up to eight automatic volume space management tasks concurrently. Parameter used is MAXMIGRATIONTASKS(8).
- You want DFSMSShsm to use the migration prefix name of HSM. Parameter used is MIGRATEPREFIX(HSM).
- Automatic primary space management begins if DFSMSShsm is running at 0230 (2:30 a.m.). DFSMSShsm ends automatic primary space management at 0630 (6:30 a.m.). Parameter used is PRIMARYSPMGMTSTART(0230 0630).
- All eligible data sets with a data set source size of 100 KB or less are considered as candidates for migration to a small-data-set-packing data set. Parameter used is SMALLDATASETPACKING(KB(100)).

```
SETSYS COMPACT(DASDMIGRATE) +
        COMPACTPERCENT(40) +
        EXITON(MD MM MV SA) +
        EXPIREDDATASETS(SCRATCH) +
        MAXMIGRATIONTASKS(8) +
        MIGRATEPREFIX(HSM) +
        PRIMARYSPMGMTSTART(0230 0630) +
        SMALLDATASETPACKING(KB(100))
```

Specifying the SETSYS parameters for DFSMSShsm processing in an MVS environment

In this example, the SETSYS parameters associated with DFSMSShsm processing in an MVS environment are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- DFSMSShsm uses the defaults for CSA limiting. Parameter used is CSALIMITS(ACTIVE(90) INACTIVE(30) MAXIMUM(100) MWE(4)).
- The following installation exits are active: initialization exit and shutdown exit. Parameter used is EXITON(IN SD).
- DFSMSShsm uses job entry subsystem 2. Parameter used is JES2.
- The DFSMSShsm address space cannot be swapped by the MVS system resource manager. Parameter used is NOSWAP.
- DFSMSShsm uses the default HSMplex name of ARCPLEX0. Parameter used is PLEXNAME(PLEX0).
- DFSMSShsm considers all host in an HSMplex eligible for promotion to take over the primary functions of a failed host. Parameter used is PROMOTE(PRIMARYHOST(YES)).
- During volume backup or migration in a multiple-DFSMSShsm-host environment, system facilities like GRS, perform data set serialization of system resources. Parameter used is USERDATASETSERIALIZATION.

```
SETSYS CSALIMITS(ACTIVE(90) INACTIVE(30) MAXIMUM(100) MWE(4)) +
        EXITON(IN SD) +
        JES2 +
        NOSWAP +
        PLEXNAME(PLEX0) +
        PROMOTE(PRIMARYHOST(YES)) +
        USERDATASETSERIALIZATION
```

Specifying the SETSYS parameters for recall processing

In this example, the SETSYS parameters associated with recall processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- The DFSMSShsm host is connected to the SYSARC_PLEX1_RCL common recall queue list structure. The parameter that is used is COMMONQUEUE(RECALL(CONNECT(PLEX1))).
- DFSMSShsm can reblock data sets during recall processing. The parameter that is used is CONVERSION.
- The return-priority installation exits are active. Parameter used is EXITON(RP).

- A maximum of fifteen recall tasks can process concurrently. The parameter that is used is MAXRECALLTASKS(15).
- A maximum of five tape recall tasks that can concurrently request tape mounts. The parameter that is used is TAPEMAXRECALLTASKS(5).
- DFSMSHsm passes the volume count parameter to DFSMSdss during the recall or recover of data that is moved by DFSMSdss. The parameter that is used is VOLCOUNT(ANY).

```
SETSYS COMMONQUEUE(RECALL(CONNECT(PLEX1))) +
      CONVERSION +
      EXITON(RP) +
      MAXRECALLTASKS(15) +
      TAPEMAXRECALLTASKS(5) +
      VOLCOUNT(ANY)
```

Specifying the SETSYS parameters for recover processing

In this example, the SETSYS parameters associated with recover processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- A maximum of two dump recover tasks can be used for FRRECOV and RECOVER FROMDUMP volume recovers.
- The DFSMSHsm host is connected to the XCF group ARCVREC1 for common recover queue processing.
- This host is eligible to be the master scheduler for the common recover queue.
- All available dump recover tasks in the HSMplex hosts that are connected to the group as members, can be used to process volume restores when the local host receives them. Parameter used is: COMMONQUEUE(RECOVER(CONNECT(REC1) MSC)).

```
SETSYS MAXDUMPRECOVERTASKS(2) +
      COMMONQUEUE(RECOVER(CONNECT(REC1) MSC))
```

Specifying the SETSYS parameters for recycle processing

In this example, the SETSYS parameters associated with recycle processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- A maximum of two recycle tasks can process concurrently. Parameter used is MAXRECYCLETASKS(2).
- When recycling tape volumes, each DFSMSHsm recycle task deallocates its input unit after each 5 connected sets of backup volumes, and after each 5 connected sets of ML2 volumes. Note: the values specified here are considered to be safe compromise values in single technology environments and you should refer to this parameters description before deciding what values to use for your installation. Parameter used is RECYCLEINPUTDEALLOCATION(BACKUP(5) MIGRATION(5)).
- During recycle processing of a tape backup volume or a tape level 2 volume, only tape spill backup volumes or tape level 2 migration volumes that can be mounted and written on the specified unit type are used for output. Parameter used is RECYCLEOUTPUT(BACKUP(3590-1) MIGRATION(3590-1)).
- DFSMSHsm uses the percent valid criteria to recycle both backup and migration tape volumes. Parameter used is RECYCLEPERCENT(20).
- DFSMSHsm does not wait for the recycle tape unit to be allocated for the source and target tape volumes. Parameter used is RECYCLETAPEALLOCATION(NOWAIT).

```
SETSYS MAXRECYCLETASKS(2) +
      RECYCLEINPUTDEALLOCATION(BACKUP(5) MIGRATION(5)) +
      RECYCLEOUTPUT(BACKUP(3590-1) MIGRATION(3590-1)) +
      RECYCLEPERCENT(20) +
      RECYCLETAPEALLOCATION(NOWAIT)
```


Specifying the SETSYS parameters for secondary space management

In this example, SETSYS parameters associated with Secondary Space Management processing are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- Automatic secondary space management begins if DFSMSHsm is running at 0800 (8:00 a.m.). DFSMSHsm ends automatic secondary space management at 1700 (5:00 p.m.). Note: MWEs built by secondary space management up to 1700 continue to process. If a large number of data sets needs deletion, MWE processing continues after secondary space management stops. Parameter used is SECONDARYSPMGMTSTART(0800 1700).
- DFSMSHsm keeps MCDS records for recalled data sets for 10 days. Volume statistics records are kept for 30 days and MCDS records for reconnection are kept for 3 days. Parameter used is MIGRATIONCLEANUPDAYS(10 30 3).
- A maximum of fifteen recall tasks can process concurrently. The parameter that is used is MAXRECALLTASKS(15).
- The maximum number of automatic secondary space management Statistics and Migration cleanup tasks that can run concurrently is 2.
- The maximum number of automatic secondary space management migration tasks for data sets from level 1 volumes to level 2 tape volumes that can run concurrently is 1.

```
SETSYS SECONDARYSPMGMTSTART(0800 1700) +
        MIGRATIONCLEANUPDAYS(10 30 3) +
        MAXRECALLTASKS(15) +
        MAXSMTASKS(CLEANUP(2) TAPEMOVEMENT(1))
```

Specifying the SETSYS parameters for DFSMSHsm processing in a security environment

In this example, the SETSYS parameters associated with DFSMSHsm processing in a security environment are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- DFSMSHsm asks RACF for the erase status of a users data set when backup versions and migration copies are scratched from DFSMSHsm-owned DASD volumes and if so requested, causes data management to overwrite the storage of the owned copy. Parameter used is ERASEONSCRATCH.
- DFSMSHsm does not create a backup RACF discrete profile when it backs up a cataloged RACF-indicated data set. Parameter used is NOPROFILEBACKUP.
- DFSMSHsm does not put RACF-indication on backup versions and migration copies. Parameter used is NORACFIND.
- DFSMSHsm protects tapes with RACF. In addition, DFSMSHsm backups up or migrates a password-protected data set to a tape volume that is not password-protected. Parameter used is TAPESECURITY(RACFINCLUDE).

```
SETSYS ERASEONSCRATCH +
        NOPROFILEBACKUP +
        NORACFIND +
        TAPESECURITY(RACFINCLUDE)
```

Specifying the SETSYS parameters for DFSMSHsm processing in a tape environment

In this example, the SETSYS parameters associated with DFSMSHsm processing in a tape environment are specified. Where appropriate, default values or recommended values are used in the example. These parameters are in effect until another SETSYS command is issued to change them.

- DFSMSHsm specifies to DFSMSdss that the fourth buffering technique is used for DFSMSdss DASD I/O for a full-volume dump. The fourth buffering technique has DFSMSdss read one cylinder at a time.

DFSMSHsm specifies to DFSMSdss that the third buffering technique is used for DFSMSdss DASD I/O, and DFSMSdss is to read five tracks at a time. Parameter used is DUMPIO(4,3).

- DFSMSHsm creates original backup tapes and migration tapes that will be kept on site and alternate tapes that will be written at a remote site. Parameter used is DUPLEX(BACKUP(Y) MIGRATION(Y)).
- The following installation exits are active: tape data set exit, and tape ejected exit. Note: the TV exit is not included because most customers are using DFSMSrmm. Parameter used is EXITON(TD TE).
- DFSMSHsm does not wait for an input tape unit to be allocated because in this example, the customer has plenty of tape units generally available. Parameter used is INPUTTAPEALLOCATION(NOWAIT).
- The number of allowable, not associated ML2 partial tapes is five. You want to recycle all not associated ML2 partial tapes. Parameter used is ML2PARTIALSNOTASSOCIATEDGOAL(5).
- DFSMSHsm waits 15 minutes for a tape to be mounted on a nonlibrary allocated drive. Parameter used is MOUNTWAITTIME(15).
- DFSMSHsm does not mark cartridge-type single-file formatted tape volumes full when the volume has been demounted. Parameter used is PARTIALTAPE(REUSE).
- DFSMSHsm requests scratch tape volumes when an end-of-volume condition occurs on a tape volume. Parameter used is SELECTVOLUME(SCRATCH).
- The recycled tape volumes are deleted from DFSMSHsm control and become scratch tapes. Parameter used is TAPEDELETION(SCRATCHTAPE).
- DFSMSHsm migrates data sets to 3590-1 tape volumes when migrating the volumes to ML2. DFSMSHsm attempts reconnection to ML2 tape on all data sets eligible to migrate. Parameter used is TAPEMIGRATION(ML2TAPE RECONNECT(ALL)).
- DFSMSHsm does issue action messages to the operator asking if tapes needed by DFSMSHsm recall, recover, and recycle input are available because a customer has nonrobotic tape mounting. Parameter used is TAPEINPUTPROMPT(BACKUPTAPES(NO) DUMPTAPES(NO) MIGRATIONTAPES(NO)).
- DFSMSHsm issues a message to the operator during TAPECOPY processing. Parameter used is TAPEOUTPUTPROMPT(TAPECOPY(YES)).
- DFSMSHsm reduces the occurrence of data set spanning tape volumes. The subparameter value of 1000 is well suited for 3590-1. Parameter used is TAPESPANSIZE(1000).
- DFSMSHsm allows a recall task to be used continuously for tape recalls for at least 10 minutes and have a tape allocated and mounted for at least 5 minutes before allowing another DFSMSHsm host to take the tape away for a higher priority request. Parameter used is TAPERECALLLIMITS(TASK(10) TAPE(5)).
- The 3590-1 tape unit is allocated to initially select scratch tapes during backup or dump. Parameter used is UNITNAME(3590-1).
- Identify an esoteric tape unit name to DFSMSHsm for output allocations of 3590-1 tape drives. Parameter used is USERUNITTABLE(TAPE).
- DFSMSHsm forces an end-of-volume (EOV) when a tape volume has reached the desired level of utilization (fullness). You want to use 97% of the capacity but retain the reduced tape spanning. Three separate SETSYS commands are required to specify each unit type to be defined. Parameters used are TAPEUTILIZATION(UNITTYPE(3590-1) PERCENTFULL(97)) and TAPEUTILIZATION(LIBRARYBACKUP PERCENTFULL(97)) and TAPEUTILIZATION(LIBRARYMIGRATION PERCENTFULL(97)).


```

SETSYS DUMPIO(4,3) +
        DUPLEX(BACKUP(Y) MIGRATION(Y)) +
        EXITON(TD TE) +
        INPUTTAPEALLOCATION(NOWAIT) +
        ML2PARTIALSNOTASSOCIATEDGOAL(0) +
        MOUNTWAITTIME(15) +
        PARTIALTAPE(REUSE) +
        SELECTVOLUME(SCRATCH) +
        TAPEDELETION(SCRATCHTAPE) +
        TAPEMIGRATION(ML2TAPE RECONNECT(ALL)) +
        TAPEINPUTPROMPT(BACKUPTAPES(NO) +
                        DUMPTAPES(NO) +
                        MIGRATIONTAPES(NO)) +
        TAPEOUTPUTPROMPT(TAPECOPY(YES)) +
        TAPESPANSIZE(1000) +
        TAPERECALLLIMITS(TASK(10) TAPE(5)) +
        UNITNAME(3590-1) +
        USERUNITTABLE(TAPE)
SETSYS TAPEUTILIZATION(UNITTYPE(3590-1) PERCENTFULL(97))
SETSYS TAPEUTILIZATION(LIBRARYBACKUP PERCENTFULL(97))
SETSYS TAPEUTILIZATION(LIBRARYMIGRATION PERCENTFULL(97))

```


Chapter 66. STOP command: Shutting down DFSMSHsm

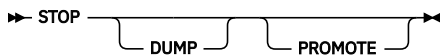
The STOP command causes an orderly shutdown of DFSMSHsm. An orderly shutdown occurs when DFSMSHsm ends each attached subtask and ends all started secondary address spaces. Because many DFSMSHsm commands and automatic functions can run for a long time, DFSMSHsm periodically checks whether you issued a STOP command. This allows DFSMSHsm to shut down as soon as possible. After DFSMSHsm detaches all subtasks and all secondary address spaces end, DFSMSHsm issues a message indicating that shutdown is complete. DFSMSHsm then returns control to the system. The STOP command should be entered only once.

If the DUMP parameter is specified with the STOP command, DFSMSHsm ends processing immediately for all primary and secondary address spaces, and there is no orderly shutdown. All secondary address spaces currently processing are canceled immediately.

When the STOP command is issued without any parameters in a sysplex environment, DFSMSHsm assumes that no promotion is desired on that host.

Another way to shut down DFSMSHsm is to use the MVS STOP command.

Syntax of the STOP command



Optional parameters of the STOP command

This topic describes the optional parameters of the STOP command.

DUMP: Requesting a dump of the DFSMSHsm address space

Explanation

DUMP is an optional parameter requesting a dump of the DFSMSHsm address space before DFSMSHsm stops processing. The SYSUDUMP, SYSMDUMP, or SYSABEND DD statement you specify in the DFSMSHsm startup procedure determines where DFSMSHsm sends the dump. If, however, you specify SETSYS SYS1DUMP, the DFSMSHsm dumps resulting from an abnormal end or error condition are written in a system dump data set. Use caution when specifying the DUMP parameter on the STOP command when there is processing going on in a secondary address space. There is a large expenditure of system resources when processing in a secondary address space, and the DUMP parameter causes all secondary address spaces to end processing immediately.

Defaults

None.

Usage notes:

1. When the STOP command is issued with the DUMP parameter in a sysplex environment, this host is eligible to have its functions taken over by another host.

PROMOTE: Specifying that a specific host in a sysplex is eligible to be promoted for another host

Explanation

PROMOTE is an optional parameter requesting that this host in a sysplex is eligible to have its primary functions or secondary space management functions taken over by another host.

Defaults

None.

Example of how to code the STOP command

The following example shows how to code the STOP command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Stopping DFSMSHsm with a dump

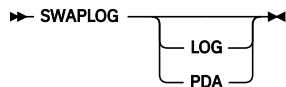
In this example, DFSMSHsm is stopped and a dump is taken.

```
STOP DUMP
```


Chapter 67. SWAPLOG command: Switching between the DFSMSHsm log or PDA data sets

The SWAPLOG command is used to swap the two log data sets or swap the two PDA data sets. You can choose which data sets to swap by using the LOG or PDA optional parameters. If no optional parameter is specified on the SWAPLOG command, only the log data sets are swapped.

Syntax of the SWAPLOG command



Optional parameters of the SWAPLOG command

This topic describes the optional parameters of the SWAPLOG command.

LOG: Specifying to swap the log data sets

The two DFSMSHsm log data sets are called the LOGX data set and the LOGY data set. The names of the data sets are specified in the procedure used to start DFSMSHsm and also in the procedure used to print the LOGY data set.

DFSMSHsm always writes information in the LOGX data set. When the LOGX data set becomes full, DFSMSHsm automatically swaps the two log data sets. You can use the SWAPLOG command to swap the two log data sets. In addition, you can specify the LOGSW=YES parameter on the START DFSMSHsm command to cause DFSMSHsm to swap the log data sets during DFSMSHsm initialization. When the log data sets are swapped, DFSMSHsm does the following:

1. Stops writing information in the LOGX data set
2. Closes the LOGX data set
3. Renames the LOGX data set to a temporary name
4. Renames the LOGY data set to LOGX
5. Renames the temporary data set name to LOGY
6. Opens the new LOGX data set
7. Resumes writing information in the new LOGX data set.

After the DFSMSHsm log data sets have been swapped, start the HSMLOG procedure that is supplied with the DFSMSHsm licensed program. This procedure normally resides in PARMLIB. The HSMLOG procedure runs the ARCPRLLOG program to do an initial formatting of the information in the LOGY data set and to print the log data set. After running the ARCPRLLOG program, the HSMLOG procedure deletes the old LOGY data set and reallocates a new LOGY data set. The ARCPRLLOG program is also supplied with the DFSMSHsm licensed program.

If you do not print the information in the LOGY data set after the log data sets are swapped and a later swap occurs, you do not have a copy of that information because DFSMSHsm writes over the old information in the LOGY data set.

The Chapter 8, “Aggregate backup and recovery support (ABARS),” on page 319 contains additional information about the contents of the DFSMSHsm log data sets and how to print information in the DFSMSHsm log.

PDA: Specifying to swap the PDA data sets

The two DFSMSHsm PDA data sets are called the ARCPDOY data set and the ARCPDOX data set.

DFSMSHsm always writes information in the ARCPDOX data set. When the ARCPDOX data set becomes full, DFSMSHsm automatically swaps the two PDA data sets. You can use the SWAPLOG command to swap the two PDA data sets.

The [*z/OS DFSMSHsm Diagnosis*](#) contains additional information about the PDA facility.

Examples of how to code the SWAPLOG command

The following examples present how to code the SWAPLOG command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Swapping the DFSMSHsm LOGX and LOGY data sets (example 1)

In this example, since no optional parameter is specified, the DFSMSHsm log data sets are swapped.

```
SWAPLOG
```

Swapping the DFSMSHsm LOGX and LOGY data sets (example 2)

In this example, the DFSMSHsm log data sets are swapped.

```
SWAPLOG LOG
```

Swapping the DFSMSHsm PDA data sets

In this example, the DFSMSHsm PDA data sets are swapped.

```
SWAPLOG PDA
```

Swapping both the DFSMSHsm log and PDA data sets

In this example, both the DFSMSHsm log and PDA data sets are swapped.

```
SWAPLOG LOG PDA
```


Chapter 68. TAPECOPY command: Making copies of selected cartridge-type tape volumes

The TAPECOPY command provides a way of copying migration level 2 and backup cartridge-type single-file tape volumes. It is an alternative method to tape duplexing, which creates two tapes during the migration, backup, or recycle processes. To distinguish between copies, the input tape is called the original, and the output tape is called the alternate.

You can issue the TAPECOPY command generically for a functional category of tape volumes. Generic processing selects only those tapes that are marked full and do not already have an alternate tape copy. When processing one or more specific volumes, tapes do not have to be already marked full, and they can have a previous tape copy alternate.

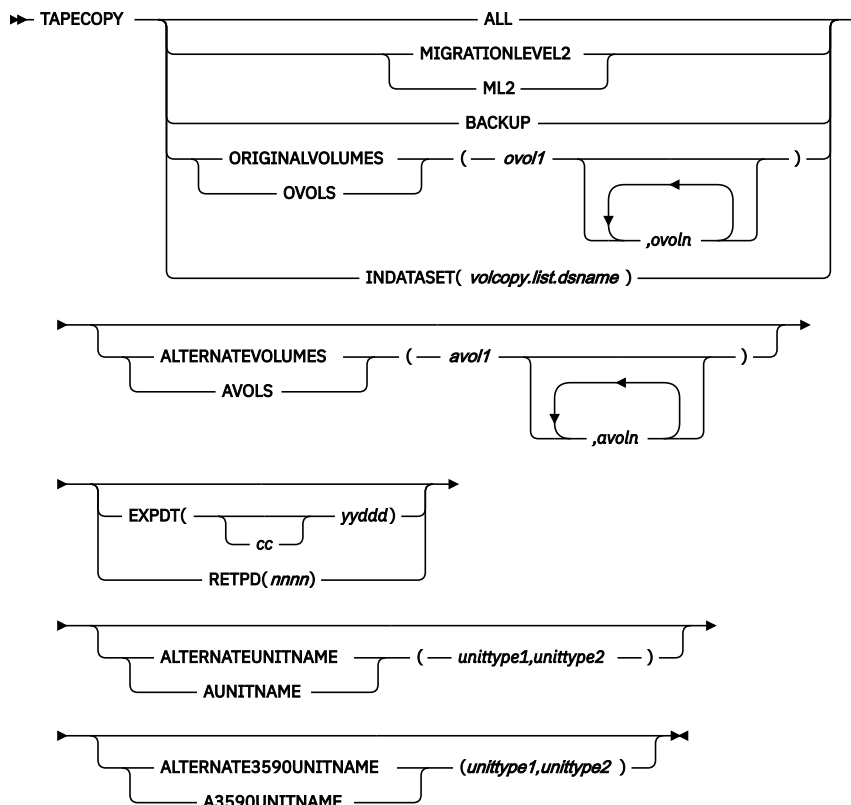
DFSMSHsm gives special consideration to migration level 2 tapes in a SETSYS PARTIALTAPE(MARKFULL) environment. When DFSMSHsm migrates data sets versus volumes; it does not mark output tapes as full even when running in a PARTIALTAPE(MARKFULL) environment. To allow these tapes to be copied, generic processing marks all partial tapes as full so that the tapes are selected.

The source volume for TAPECOPY can reside inside or outside a system-managed tape library. The ACS routines can be used to control whether a system-managed tape volume is used for TAPECOPY output.

Note: If *unittype* is not specified on the TAPECOPY command, it is not provided as input when DFSMSHsm invokes the ACS routines for the TAPECOPY command.

See [Set up the ACS routines in z/OS DFSMSHsm Implementation and Customization Guide](#) for more information on configuring your SMS environment with HSM.

Syntax of the TAPECOPY command



Required parameters of the TAPECOPY command

This topic describes the required parameters of the TAPECOPY command.

ALL: Specifying to copy all eligible cartridge-type single-file tape volumes

Explanation: ALL specifies that all cartridge-type single-file tape volumes that are marked as full in the OCDS TTOC record and do not have an alternate volume are to be copied. If the environment is PARTIALTAPE(MARKFULL) when this command is issued, all partially full migration level 2 tapes will be marked full.

Tape library relationship: A UNITNAME can be used for filtering in the ACS routines. If you specify the unit type for this command, it will be sent to the ACS routines in SMS for the TAPECOPY function.

When ALL is specified, DFSMSHsm uses scratch tapes for the output or target volumes.

Defaults: None.

MIGRATIONLEVEL2: Specifying to create copies of all eligible migration level 2 volumes

Explanation: MIGRATIONLEVEL2 specifies that all cartridge-type single-file migration level 2 tape volumes that are marked as full in the OCDS TTOC record and do not have alternate volumes are to be copied. If the environment is PARTIALTAPE(MARKFULL) when this command is issued, all partially full migration level 2 tapes will be marked full.

When MIGRATIONLEVEL2 is specified, DFSMSHsm uses scratch tapes for the output or target volumes.

Defaults: None.

BACKUP: Specifying to create copies of all eligible backup tape volumes

Explanation: BACKUP specifies that all cartridge-type single-file backup tape volumes that are marked as full in the OCDS TTOC record and do not have alternate volumes are to be copied. When BACKUP is specified, DFSMSHsm uses scratch tapes for the output or target volumes.

Defaults: None.

ORIGINALVOLUMES: Copying specified volumes

Explanation: ORIGINALVOLUMES(*ovol1...ovoln*) is a required parameter (if you are specifying which volumes are to be copied) requesting that alternate tape volume copies be created. The volumes to be copied are specified by their volume serial numbers (*ovol*). The alternate tape volumes are made even though the original volumes may not be marked full or may already have an alternate tape volume reference in the OCDS TTOC record.

The optional parameter ALTERNATEVOLUMES may be used to specify which output volumes are to be used in place of using scratch tapes. If specified, the number of entries in ORIGINALVOLUMES and ALTERNATEVOLUMES must match. The TAPECOPY function will maintain a one-for-one correspondence between the two sets of entries.

Defaults: None.

INDATASET: Specifying to use a data set containing a volume list

Explanation: INDATASET(*volcopy.list.dsname*) is a required parameter specifying to use a list of original tape volumes contained in the data set supplied and create alternate tape volume copies of them. The alternate tape volumes are made even though the original volumes may not be marked as full.

volcopy.list.dsname is the data set name of the cataloged data set. This data set must *not* be a migrated data set when the command is issued.

When allocating the volume list data set, use the following parameters for:

RECFM

Specify F, FB, FS, or FBS

LRECL

Specify 80

BLKSIZE

Specify a multiple of LRECL

DSORG

Use physical sequential (PS) organization.

When building the volume list contents, use the following entry format:

Columns**Contents****1-6**

Original volume source to be copied. Specify this volume by its volume serial number.

8-13

Alternate volume target specified by its volume serial number. You can also use a volume serial number of PRIVAT, meaning that DFSMSHsm is to use a scratch tape for the target. If a volume serial number is not specified, the default is the value found in the alternate volume field of the OCDS TTOC record.

15-21

This field is used to specify an alternate tape volume expiration date. The date is placed in the tape label but has no relation to when DFSMSHsm will remove the tape from its inventory. The alternate tape is deleted by DFSMSHsm at the same time the original tape is deleted. Use the format as *ccyyddd* and right justify the date within the field. If no expiration date is specified in this field, the expiration date specified or defaulted for the command is used.

The date is checked for validity, and the tape listed in the input record is not copied if the check fails. If the *cc* field (below) is blank, the date limits are the current date to 31 December 1999. Date limits are otherwise the current date to 31 December 2155.

cc

This is a field used to specify the first two digits of the year. If the current date is before 1 January 2000, this field can be left blank; the year will be interpreted as 19yy.

yy

This field specifies the last two digits of the year.

ddd

This field specifies the Julian day in the year.

Note:

1. If an expiration date is supplied, it is processed as described under the EXPDT parameter. The exception is that if an expiration date is invalid, only the tape listed in the input record is failed.
2. If the expiration date is not supplied in an input record, the EXPDT parameter on the TAPECOPY command is used.
3. Past dates are not recognized as expiration dates.
4. Values for *ccyyddd* of 1999365 and 1999366 are recognized as “never expire” dates. Values of 1998000 and 1999000 are recognized as meaningful to certain tape management systems.
5. If no expiration date is supplied, DFSMSHsm uses the expiration date specified or defaulted for the command.

Defaults: None.

Example: The following is an example of the format for the input volume list data set:


```

Columns:1----6 8---13 15---21
Format: source target ccyyddd

      * * * Top of Data * * *
Line 1: JHG191 DBK001 2101001
Line 2: ML1002 PRIVAT 2101059
Line 3: ML2100 DBK070
Line 4: ML2101
      * * * End of Data * * *

```

Legend:**Line 1**

JHG191 is the source volume, DBK001 is the target volume and retain the target volume until January 1, 2101.

Line 2

ML1002 is the source volume, use a scratch tape for the target volume and retain the target volume until February 28, 2101.

Line 3

ML2100 is the source volume, DBK070 is the target volume and retain until the date (if any) specified with the EXPDT parameter.

Line 4

ML2101 is the source volume, use a scratch tape and retain the target volume until the date (if any) specified with the EXPDT parameter.

Optional parameters of the TAPECOPY command

This topic describes the optional parameters of the TAPECOPY command.

ALTERNATEUNITNAME: Specifying alternate unit names for a tape copy

Explanation: ALTERNATEUNITNAME(*unittype1*,*unittype2*) is used to specify the unit name associated with only a 3480, 3480X or 3490 cartridge-type single-file tape. When ALTERNATEUNITNAME is specified, the alternate tape will be allocated using the specified unit name. This will be done as long as there is no incompatibility in the unit name.

If there are two unit names specified with ALTERNATEUNITNAME, DFSMSHsm will use *unittype1* when the source volume is a standard cartridge, and it will use *unittype2* when the source volume is an enhanced capacity cartridge.

Defaults: None.

Note:

1. The following table shows the compatibility between original volume unit names and alternate volume unit names. This table applies when TAPECOPY output is being directed outside a system managed tape library.

Table 67. Compatibility between original volume unit names and alternate volume unit names					
Original Volume Unit Name	Alternate Volume Unit Name				
	3480	SETSYS NOTAPEHWC 3480X	SETSYS TAPEHWC 3480X	3490	3590-1
3480	C	C	I	I	I
3480X (non-IDRC)	C	C	I	I	I
3480X (IDRC)	I	I	C	I	I
3490	I	I	I	C	I

Table 67. Compatibility between original volume unit names and alternate volume unit names (continued)

Original Volume Unit Name	Alternate Volume Unit Name				
	3480	SETSYS NOTAPEHWC 3480X	SETSYS TAPEHWC 3480X	3490	3590-1
3590-1	I	I	I	I	C
Note: C = Compatible I = Incompatible					

Important Note:

You can force DFSMSHsm to use a unit specified with ALTERNATEUNITNAME regardless of compatibility as per the table above, by using the following PATCH:

```
PATCH .MCVT.+1BC BITS(..1.....)
```

Note, however, that you can force DFSMSHsm to use this unit only if the following conditions are met:

- The alternate unit name is a native device (*not* virtual).
- The unit name is *not* defined as an esoteric to DFSMSHsm (in other words, SETSYS UUT is *not* used to define the unit name).
- The unit name does *not* contain mixed device types (for example, 3590 and 3592).

If all of these conditions above are true and you have the patch set in the MCVT, DFSMSHsm will allow replacement of the tape using the alternate unit name specified, regardless of the existing original unit. All records associated with the original volume and its data sets will be updated to reflect the new unit name and type.

2. The 3480, 3480X, and 3490 can be considered valid generic unit names for the ALTERNATEUNITNAME parameter.

ALTERNATE3590UNITNAME: Specifying 3590 alternate unit name for a tape copy

Explanation: ALTERNATE3590UNITNAME(*unittypel*,*unittypel2*) is used to specify the unit name associated with only the IBM 3590 tape subsystem. When ALTERNATE3590UNITNAME is specified, the alternate tape will be allocated using the specified unit name. This will be done as long as there is no incompatibility in the unit name.

Defaults: None.

ALTERNATEVOLUMES: Specifying which volumes to copy

Explanation: ALTERNATEVOLUMES(*avol1*,...*avoln*) is an optional parameter that defines which output volumes are to be used in place of scratch tapes. These tape volumes are indicated by their volume serial numbers *avol*. If specified, the number of entries in ORIGINALVOLUMES and ALTERNATEVOLUMES must match. The TAPECOPY function will maintain a one-for-one correspondence between the two sets of entries. The alternate tape volumes are made even though the original volumes may not be marked as full.

When considering 3480, 3480X, and 3490 volumes, the original volumes and the alternate volumes must use the same recording technology. If the original volume is written using the Improved Data Recording Capability (IDRC) of the 3480X, then the alternate volumes must be written using the IDRC. The reverse is true as well.

For 3490 volumes or 3590-1 volumes, the original volumes and alternate volumes must use the same media type. If the original volume is written using a standard capacity cartridge, then the alternate volume must be written with a standard capacity cartridge (an enhanced capacity cartridge cannot be used). The reverse is true as well.

Defaults: None.

EXPDT and RETPD: Specifying a tape expiration date or a retention period

Explanation: EXPDT(ccyyddd) and RETPD(nnnn) are mutually exclusive, optional parameters specifying either an expiration date or a retention period for the alternate tape volume. The date is placed in the tape label but has no relation to when DFSMSHsm will remove the tape from its inventory. The alternate tape is deleted by DFSMSHsm at the same time the original is deleted.

It is important for you to understand that specifying an expiration date for the alternate tape volume that would cause the alternate tape volume to return to scratch before the original tape volume returns to scratch can result in data loss. For example, consider the following scenario:

- Tape 1 is an original volume that has Tape 2 as its alternate volume.
- Tape 2 is returned to scratch by the Tape Management System because the specified expiration date is reached. Tape 2 is now known to the Tape Management System as a scratch tape; however, DFSMSHsm regards this tape as a valid alternate of Tape 1. (The expiration date does not relate to when the tape is removed from the DFSMSHsm inventory.)
- Tape 2 is now mounted in response to a scratch request from either DFSMSHsm or another user request. After it has been written to, Tape 2 is not a valid alternate for Tape 1 anymore because it has been overwritten; however, DFSMSHsm still considers Tape 2 a valid alternate of Tape 1. This, in itself, poses a problem if the alternate tape is needed, but the potential for problems continues.
- Tape 1 is now returned to scratch by the ARCTVEXT exit, which also results in Tape 2 being returned to scratch. Now, Tape 2 is again a scratch volume and the data written to Tape 2 after its return to scratch the first time can be overwritten again if mounted in response to a scratch request.

EXPDT is an optional parameter that specifies a different expiration date (other than 1999365) for the alternate tape volume. The date is checked for validity, and the entire command is rejected if the test fails. Date limits for ccyyddd are from the current date to 31 December 2155.

Values for ccyyddd of 1999365 and 1999366 are recognized as “never expire” dates. Values of 1998000 and 1999000 are recognized as meaningful to certain tape management systems.

RETPD is an optional parameter that sets the expiration date by specifying a retention period. The valid range is 0–9999. RETPD is converted to an expiration date and is processed as described under EXPDT.

Defaults: None.

Table 68 on page 1232 shows all of the conditions for *each* tape volume processed by DFSMSHsm. If the tape data set exit (ARCTDTEXT) is in effect, DFSMSHsm invokes the exit after processing any of the conditions shown in Table 68 on page 1232 and puts the date specified by the exit on the tape being copied.

Table 68. DFSMSHsm Processing by Condition				
Conditions				
Is an expiration date specified in the INDATASET record?	Is an expiration date specified on the TAPECOPY command?	Is the SETSYS TAPESECURITY (EXPIRATION) command in effect?	Is an expiration date specified for the original volume?	DFSMSHsm...
no	no	no	no	Does not protect the alternate volume with an expiration date.
no	no	no	yes	Uses its default date of 1999365.
no	no	yes	no	
no	no	yes	yes	

Table 68. DFSMSHsm Processing by Condition (continued)

Conditions				
no	yes	no	no	Uses the expiration date specified on the TAPECOPY command.
no	yes	no	yes	
no	yes	yes	no	
no	yes	yes	yes	
yes	no	no	no	Uses the expiration date specified in the INDATASET record.
yes	no	no	yes	
yes	no	yes	no	
yes	no	yes	yes	
yes	yes	no	no	
yes	yes	no	yes	
yes	yes	yes	no	
yes	yes	yes	yes	

Examples of how to code the TAPECOPY command

The following examples present different ways to code the TAPECOPY command:

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Specifying to copy all eligible cartridge-type single-file tape volumes

The following example illustrates how to make a copy of all the cartridge-type single-file migration and backup tapes that do not already have a copy:

```
TAPECOPY ALL
```

Specifying to copy a set of tape volumes

The following example illustrates how to make a copy of more than one original volume on specific alternate volumes:

```
TAPECOPY ORIGINALVOLUMES(JHG191,JHG195,JHG198) +  
ALTERNATEVOLUMES(DBK001,DBK005,DBK008)
```

Specifying to copy a list of original volumes

This example illustrates using a data set containing a list of volumes to process that are to be retained until December 31, 2010. The expiration date for the alternate tape volume does not relate to when the tape volume is removed from the DFSMSHsm inventory. In this example, if the original volume is removed from the DFSMSHsm inventory after December 31, 2010, there can be an exposure to data loss.

```
TAPECOPY INDATASET(G737495.VOLIST.DATA) +  
EXPDT(2010365)
```


Specifying the TAPECOPY command with enhanced capacity cartridge support

The 3490E backup and migration volumes residing on standard tape cartridges are copied using the esoteric tape unit name ES1. Those backup and migration volumes residing on enhanced capacity cartridges are copied using the esoteric tape unit name ES2. The 18-track format cartridges are copied to units allocated using ES1.

```
TAPECOPY BACKUP ALTERNATEUNITNAME(ES1, ES2)
```

All backup volumes are copied using esoteric tape unit name ES1, regardless of the type of cartridge (standard vs enhanced capacity). If only one unit is specified with ALTERNATEUNITNAME, it is the device type used for all TAPECOPY processing for backup volumes. In this case, potential exists for TAPECOPY failures if a 3490E backup volume is an enhanced capacity cartridge, and ES1 is an esoteric tape unit that is being used for output to standard tapes (for example, devices are preloaded with standard tapes).

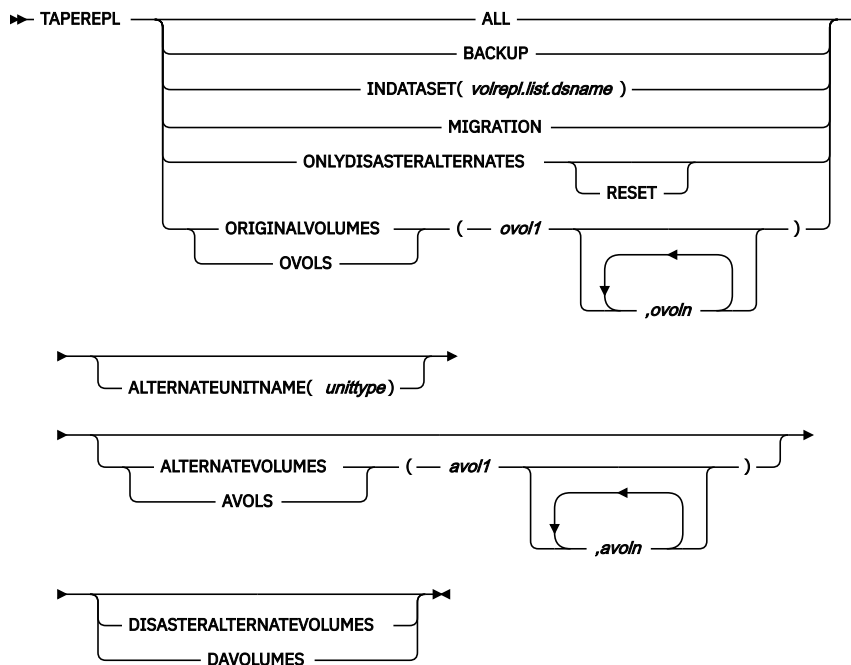
```
TAPECOPY BACKUP ALTERNATEUNITNAME(ES1)
```


Chapter 69. TAPEREPL command: Replacing cartridge-type tape volumes with their alternate volumes

The TAPEREPL command provides a way of replacing an original migration level 2 volume or a backup volume with its alternate volume. It also plays an important part during an actual disaster recovery or when you are running a disaster recovery test by allowing you to replace original volumes with their alternate volumes.

For a discussion of the steps to follow when making disaster backup copies of DFSMSHsm-owned tape volumes, see “Disaster alternate volumes” on page 442.

Syntax of the TAPEREPL command



Required parameters of the TAPEREPL command

This topic describes the required parameters of the TAPEREPL command.

ALL: Specifying replacement of all cartridge-type single-file tape volumes

Explanation: ALL is a required parameter specifying replacement of all cartridge-type single-file tape volumes that have alternate-tape-volume references in their OCDS TTOC records.

Defaults: None.

BACKUP: Specifying replacement of all cartridge-type single-file backup tape volumes

Explanation: BACKUP is a required parameter specifying that all references to cartridge-type single-file backup tapes with alternate volumes be replaced with references to the alternate tape volumes.

Defaults: None.

INDATASET: Specifying the use of a data set containing a volume list

Explanation: INDATASET is a required parameter that causes a list of original tape volume references in the supplied data set to be replaced with a list of alternate tape volume references. *volrepl.list.dsname* is the data set name of the cataloged data set. This data set must *not* be a migrated data set when the command is issued.

When allocating the volume list data set, use the following parameters for:

RECFM

Specify F, FB, FS, or FBS

LRECL

Specify 80

BLKSIZE

Specify a multiple of LRECL

DSORG

Use physical sequential (PS) organization.

When building the volume list contents, use the following entry format:

Columns

Contents

1-6

Original tape volume serial number to be replaced.

8-13

Alternate tape volume serial number to use. If an alternate tape volume serial number is not specified, the value found in the alternate volume field of the original volume's OCDS TTOC record is used.

Defaults: None.

Example: The following is an example of the format for the input volume list data set:

```
Columns: 1---6 8---13
Content: orig.  alt.
         volser volser

         * * * Top of Data * * *
Line 1:  ML1100 DBK070
Line 2:  ML2101
         * * * End of Data * * *
```

Legend:

Line 1

specifies replacement of the original tape volume labeled ML1100 with the alternate tape volume DBK070.

Line 2

specifies replacement of the original tape volume labeled ML2101 with the alternate tape volume specified in the OCDS TTOC record for volume ML2101.

MIGRATION: Specifying replacement of all cartridge-type single-file migration tape volumes

Explanation: MIGRATION is a required parameter specifying that all references to cartridge-type single-file migration tapes with alternate volumes be replaced with references to the alternate tape volumes.

Defaults: None.

ONLYDISASTERALTERNATES: Converting disaster alternate volumes to original volumes

Explanation: ONLYDISASTERALTERNATES is a required parameter that converts all disaster alternate volumes to original volumes. No optional parameters are used with ONLYDISASTERALTERNATES.

RESET is an optional subparameter that changes original volumes flagged as having disaster alternates back to the status of original volumes having normal alternates.

Defaults: None.

ORIGINALVOLUMES: Specifying volumes to be replaced

Explanation: ORIGINALVOLUMES is a required parameter specifying that all references to the original tape volume be replaced with references to the alternate tape volume. The alternate tape volume can be specified using the ALTERNATEVOLUMES parameter or, in its absence, the one found in the original volume's OCDS TTOC record.

Defaults: None.

Optional parameters of the TAPERREPL command

This topic describes the optional parameters of the TAPERREPL command.

ALTERNATEUNITNAME: Specifying an alternate unit name for a tape replacement

Explanation: ALTERNATEUNITNAME (*unittype*) is used to specify the unit name associated with 3480, 3480X or 3490 cartridge-type single-file tape. When ALTERNATEUNITNAME is specified, the tape unit name will be changed in the OCDS TTOC. This will be done as long as there is no incompatibility in the unit name.

Defaults: None.

Note:

1. The following table shows the compatibility between original volume unit names and alternate volume unit names.

Table 69. Compatibility between original volume unit names and alternate volume unit names					
Original Volume Unit Name	Alternate Volume Unit Name				
	3480	SETSYS NOTAPEHWC 3480X	SETSYS TAPEHWC 3480X	3490	3590-1
3480	C	C	I	I	I
3480X (non-IDRC)	C	C	I	I	I
3480X (IDRC)	I	I	C	I	I
3490	I	I	I	C	I
3590-1	I	I	I	I	C
Note: C = Compatible I = Incompatible					

2. The 3480, 3480X, and 3490 can be considered valid generic unit names for the ALTERNATEUNITNAME parameter.

3. If DISASTERALTERNATEVOLUMES is also specified, then the ALTERNATEUNITNAME will only be used with the DAVOLUMES while in DISASTERMODE. The ALTERNATEUNITNAME does *not* permanently replace the original unit. The LIST TTOC command will still show the original unit.

When the time comes to permanently replace the original volumes with the alternate volumes, and you issue a TAPERREPL command without DAVOLUMES, the ALTERNATEUNITNAME must be specified again or the original unit name will be used. The original unit name is used if ALTERNATEUNITNAME is not specified on the command.

ALTERNATEVOLUMES: Specifying the replacement volumes to use

Explanation: ALTERNATEVOLUMES is an optional parameter that identifies specific alternate tape volumes to substitute for the original tape volumes. The alternate tape volumes are selected by their volume serial numbers *avol*. If specified, the number of entries in ORIGINALVOLUMES and ALTERNATEVOLUMES must match. The TAPERREPL function maintains a one-for-one correspondence between the two sets of entries.



Attention: The volumes that are specified in the ALTERNATEVOLUMES parameter will replace the volumes in ORIGINALVOLUMES even if they do not contain valid DFSMSHsm data.

Defaults: None.

DISASTERALTERNATEVOLUMES: Identifying the alternate volume as the disaster alternate volume

Explanation: DISASTERALTERNATEVOLUMES is an optional parameter that causes each alternate volume to be flagged as a disaster alternate volume. If DISASTERALTERNATEVOLUMES is specified, a flag is set for each ML2 tape and backup tape that has an alternate volume. The designation of "disaster alternate volume" helps distinguish between alternate volumes initially present at the recovery site and alternate volumes created later at the recovery site. If DISASTERALTERNATEVOLUMES is not specified, each original volume is replaced by its alternate volume in any of the DFSMSHsm volume and data set records that refer to it. For more information on the use of disaster alternate volumes, refer to the [Chapter 9, "Making disaster backup copies of DFSMSHsm-owned tape volumes,"](#) on page 435.

Defaults: None.

Examples of how to code the TAPERREPL command

The following examples present different ways to code the TAPERREPL command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Specifying replacement of a damaged tape

The following example illustrates how to replace a damaged tape by its alternate, which is identified in the OCDS TTOC record for the damaged tape:

```
TAPERREPL ORIGINALVOLUMES(HB0123)
```

Specifying replacement of a set of tape volumes

The following example illustrates how to replace more than one original tape volume with respective alternate volumes:

```
TAPERREPL ORIGINALVOLUMES(JHG191,JHG195,JHG198) +  
ALTERNATEVOLUMES(DBK001,DBK005,DBK008)
```


Specifying replacement of a list of original volumes

This example illustrates the use of a data set containing a list of volumes to process.

```
TAPERREPL INDATASET(G737495.VOLIST.DATA)
```

Chapter 70. TRAP command: Requesting a dump when a specified error occurs

The TRAP command is to be used only for maintenance purposes and is explained in the [*z/OS DFSMSHsm Diagnosis*](#).

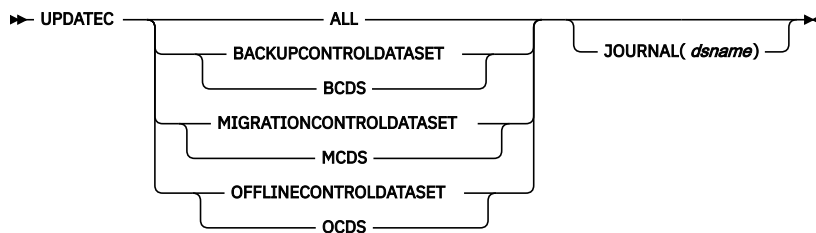
Chapter 71. UPDATEC command: Recovering the control data sets

The UPDATEC command rebuilds an updated control data set and combines the entries in the specified journal data set with the restored backup copy of the backup control data set (BCDS), migration control data set (MCDS), or offline control data set (OCDS). Use the UPDATEC command to fix any major problems that occur in the control data sets.

When you issue the UPDATEC command, DFSMSHsm places itself in emergency mode and does not write in the journal data set.

Note: For performance reasons, the enhanced CDS recovery function should be used to recover control data sets instead of the UPDATEC command. For information on using enhanced CDS recovery functions, see [“Recovering control data sets using the enhanced CDS recovery function”](#) on page 508.

Syntax of the UPDATEC command



Required parameters of the UPDATEC command

This topic describes the required parameters of the UPDATEC command.

ALL, BACKUPCONTROLDATASET, MIGRATIONCONTROLDATASET, and OFFLINECONTROLDATASET: Specifying which control data set to update

Explanation

ALL, BACKUPCONTROLDATASET, MIGRATIONCONTROLDATASET, OFFLINECONTROLDATASET are mutually exclusive, required parameters that specify which control data sets DFSMSHsm should update.

ALL

DFSMSHsm combines the entries from the journal data set with the restored copies of all three control data sets.

BACKUPCONTROLDATASET

DFSMSHsm combines the entries from the specified journal data set with the restored backup copy of the BCDS.

MIGRATIONCONTROLDATASET

DFSMSHsm combines the entries from the specified journal data set with the restored backup copy of the MCDS.

OFFLINECONTROLDATASET

DFSMSHsm combines the entries from the specified journal data set with the restored backup copy of the OCDS.

Defaults

None.

Optional parameters of the UPDATEC command

This topic describes the optional parameters of the UPDATEC command.

JOURNAL: Specifying the journal data set to use

Explanation

JOURNAL(*dsname*) is an optional parameter specifying the fully qualified name of the journal data set to use with UPDATEC. The parameter variable is explained:

dsname

Fully qualified name of the journal data set that DFSMSHsm should use when it updates the BCDS, MCDS, and OCDS.

Defaults

If you do not specify JOURNAL, DFSMSHsm uses the journal data set identified with the JOURNAL DD statement in the DFSMSHsm startup procedure.

Usage notes:

1. Specify JOURNAL only if you do not want to use the journal data set identified with the JOURNAL DD statement in the DFSMSHsm startup procedure.
2. If a journal is used other than the one currently identified in the journal DD statement (a journal other than the current journal means a backup journal), and if that journal is on tape, the following procedure is required:
 - a. The tape must be mounted without the write-protect ring.
 - b. After mounting the tape, you will receive a message (IEC510D) stating that the program is attempting to write on the tape, but the write-protect ring is not inserted. Reply to this message with Rnn,F which rewinds the tape. After the tape rewinds, insert the write-protect ring and remount the tape. After the tape is remounted, ready the device.

Examples of how to code the UPDATEC command

The following examples present different ways to code the UPDATEC command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Updating the MCDS, BCDS, and OCDS with a specified journal data set

In this example, the MCDS, BCDS, and OCDS are updated by using a specified journal data set.

```
UPDATEC ALL JOURNAL(ALT.JRNL)
```

Updating the MCDS

In this example, the MCDS is updated using the journal data set identified by the journal DD statement in the DFSMSHsm startup procedure.

```
UPDATEC MIGRATIONCONTROLDATASET
```

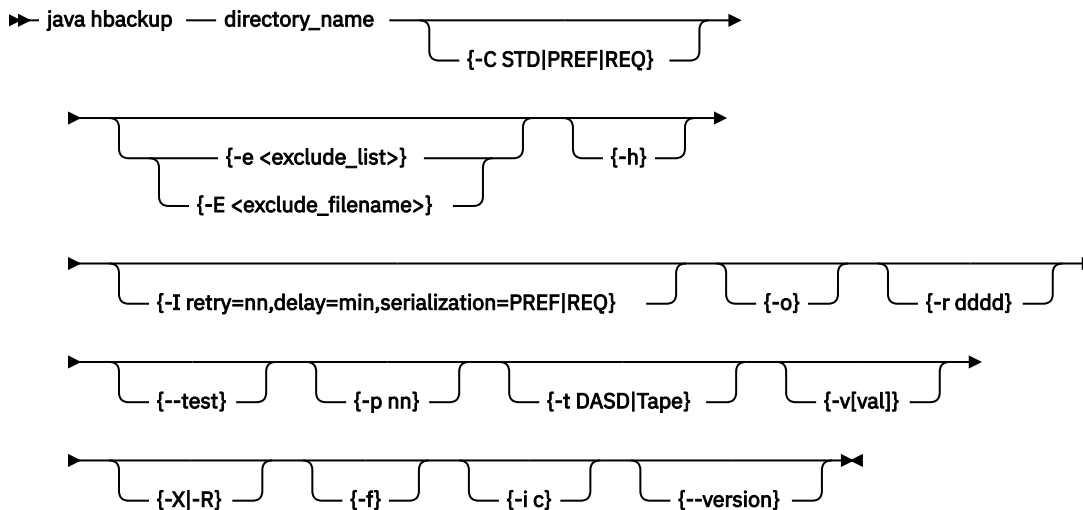
Chapter 72. Storage administrator mode for UNIX commands

The UNIX commands, **hbackup**, **hbdelete**, and **hrecover** can be run as storage administrator commands by specifying the **-o** option on the command. When the **-o** option is specified, the command performed is equivalent to the BACKDS, BDELETE, or RECOVER commands. Authorization to the appropriate RACF Facility class is checked, and if allowed, the command bypasses further permission checking on the UNIX file that is being processed.

Chapter 73. Java *hbackup* command for Incremental UNIX file backup

The Java™ *hbackup* program examines the requested directory subtree for files that are changed since their last backup. The files and directories that are examined can be modified by exclude criteria, either on the command or in a file. Only files that DFSMSHsm supports for backup are examined. The program runs under the authority of the user and you should permit the user ID READ access to SUPERUSER.FILESYS in the UNIXPRIV class to avoid permission errors during the examination.

Syntax of the Java *hbackup* command



Note: Requires CLASSPATH and LIBPATH to be set correctly.

Required parameters of the Java *hbackup* command

This topic describes the required parameter of the *hbackup* command.

directory_name: Specifying the name of the directory

Explanation

directory_name is a required parameter that specifies the starting directory for the *hbackup* program to start examining for files. Only one directory name can be specified.

Optional parameters of the Java *hbackup* command

This topic describes the optional parameters of the Java *hbackup* command.

-X

Specifies that subdirectories should be examined for files in need of backup. However, if a subdirectory is the mount point for a different file system, then that subdirectory should not be entered. (Equivalent to RECURSE(NOCROSSMOUNTS) on the BACKDS Storage Administrator command.) Default processing is not to examine subdirectories. This option cannot be specified along with the -R option.

-R

Specifies that subdirectories should be examined for files in need of backup. All subdirectories are entered until no more are found. (Equivalent to RECURSE(CROSSMOUNTS) on the BACKDS Storage Administrator command.) Default processing is not to examine subdirectories. This option cannot be specified along with the -X option.

-C STD|PREF|REQ

Specifies what form of zFS File Snapshot processing to use for the z/OS UNIX files. Options are:

STD

Do not use zFS File Snapshot to process the UNIX files. Serialization is held on the z/OS UNIX file during the backup.

PREF

Use the zFS File Snapshot processing if possible. When the zFS File Snapshot is initiated, serialization on the file is released, and other processes can write to the file without changing the contents that are being backed up. When the backup is complete, the zFS File Snapshot resources are released.

REQ

zFS File Snapshot processing must be used. If the feature is unavailable, the backup is failed.

-e

Specifies exclude criteria *<exclude_list>* for file names and directories to be excluded from backup processing. Comma separated exclude criteria patterns can be any of the following:

- Patterns to match files in the exclude list can contain any valid file name characters and might contain the wildcard characters: question mark (?) and asterisk (*). Directory names cannot be specified in a file name pattern.
- An asterisk (*) can be used to match any number of characters within a file name. An asterisk may be used at any position in the pattern.
- A question mark (?) can be used to represent any single character in the file name. Multiple question mark characters are allowed.
- A backslash character (\) must be used to escape any question mark, asterisk, open curly brace, open square bracket, and backslash characters to indicate it is not a special character. (for example \?, *, \{, \[, or \\\). Other CCSIDs might have a different character for hex value x'5C' (CCSID 037 backslash) that need to be specified. For example, CCSID 1141 x'5C' would be the umlaut O (Ö).
- If the requested file name pattern does not include any wildcard characters, then that exact name is used to compare with files that are found in the subdirectories processed.
- Directory names that are specified in the exclude list must end in a forward slash character. If the directory name begins with a forward slash character, then it is considered as an absolute path. If the directory name does not begin with a forward slash character, it is considered as a relative path that begins at the directory that is requested on the command. Wildcard characters cannot be used in directory names.
- The comma (,) character cannot be used in file name patterns or directory names.
- The syntax */.../* can be used to indicate any number of directories matched. (Equivalent to */**/* in glob format.)
- A combination of directory names and file name pattern.
- Trailing white space in a criteria is trimmed off.

-E

Specifies a file name, *exclude_filename*, that contains the exclude criteria to be used during processing. The exclude criteria must be specified one per line. A line starting with a blank character is considered a comment and is ignored. Valid criteria must conform to the types of patterns described in -e description. The user must have read permission to the file.

-f

Force a new backup. Send all supported, matching files to DFSMSHsm for backup.

-h

Write out this explanation of options.

-I

UNIX files that are in use at the time of backup will have their backup retried up to a specified maximum number of times, delaying a specified number of minutes in between each retry. On retry, an attempt to obtain exclusive serialization to the UNIX file will be made. If on retry exclusive serialization is obtained then a normal backup of the UNIX file will be made. If exclusive serialization cannot be obtained, then additional retries of the backup will be made until the maximum number of retries has been made. If after the maximum number of retries exclusive serialization cannot be obtained, the backup request will either fail or a backup of the file will be taken anyway, depending on if the command specified serialization was required or preferred.

The -I option has suboptions *retry*, *delay*, and *serialization* that enable the user to specify the maximum number of retries, the time to delay in between retries, and whether exclusive serialization is required or preferred. The -I option and its associated suboptions has the following syntax.

```
-I retry=nn,delay=min,serialization=PREF|REQ
```

retry

Specifies the maximum number of retries. DFSMShsm will retry a maximum of *nn* times to back up a UNIX file after the first attempt fails because of the UNIX file is in use. For *nn*, specify a value 0 to 99. The default value is 0.

delay

Specifies the number of minutes to delay in between retries. DFSMShsm delays for *min* minutes before retrying a backup attempt which failed because the UNIX file is in use. For *min*, specify a value from 0 to 999. The default value is 15.

serialization

Specifies whether exclusive serialization of the UNIX file is required or preferred to make the backup. The default is serialization is required. The following are the possible values for the serialization suboption:

PREF

Serialization is preferred. If the UNIX file is still in use after the max number of retries for the backup have been made, the UNIX file will be backed up anyway. A message indicating a fuzzy backup was made will be issued. If the UNIX file is not in use at that time, a normal backup will be made. If *retry=0* is in effect and exclusive serialization could not be obtained on the first attempt then a fuzzy backup will be made.

REQ

Serialization is required. If the UNIX file is still in use after the max number of retries for the backup have been made, the backup request will fail.

The -I suboptions can be specified in any order and any omitted suboption will have their default values used. However, at least one suboption must be specified.

All supported UNIX file types can be backed up with the -I option, however only Regular Files will be checked to be in use. Symbolic links, FIFOs, and directories will not be checked to be in use and will not have their backup retried.

zFS File Snapshot is not supported for in use UNIX files. If *serialization=PREF* and *-C PREF* or *-C REQ* are specified on the command and the UNIX file is in use at the time of backup, the backup request will still fail.

-o

Storage Administrator mode. Any files that are sent to DFSMShsm for backup are processed as if they are sent by using a BACKDS Storage Administrator command. If using this option, you should also switch to UID(0) by using the *su UNIX* command first so that all files are accessible to the program.

-p

Specifies the maximum number of outstanding requests that are sent to the DFSMSShsm host at any one time. Default value is 100.

-r

Specifies the number of days to keep this backup past when it would normally be expired or rolled off. (RETAININDAYS) Default: None

-t

Specifies the target of the backup. It can be either DASD, tape, or not specified in where DFSMSShsm chooses the backup target. Default: None

-test

Requests only the matched file names are displayed, but not sent to the DFSMSShsm host for backup.

-i

Specifies the host ID of the DFSMSShsm host that the backup request will be directed to. Host ID must be 1 character A-Z, 0-9, @, #, or \$.

The host ID must correlate to a DFSMSShsm host that is active on the system from which the java hbackup command was issued. If the DFSMSShsm host on the system with the specified host ID is inactive or does not exist, message ARC0055A will be issued to the operator console and processing will not continue until the message is responded to. If a filemode host is active on the system, then the backup request will be directed to the filemode host that has the specified host ID. Using this option will not direct backup requests to DFSMSShsm hosts on other systems within the PLEX.

This option, on its own, does not provide functionality to divide and distribute UNIX file backups to multiple DFSMSShsm hosts on a system. The user must decide how to divide and distribute UNIX file backups and which specific DFSMSShsm hosts to direct those backup requests to.

-version

Display the current version of the *hbackup* program.

-v{val}

Issue verbose messages during processing. An optional value might be specified for a level of verbosity. For example, -vDTL . If -v is specified without a value, it defaults to -vSTAT.

val might be:

- STAT - Display statistics.
- DTL - Display detailed information about file selection.
- TIMES - Display timestamps on messages.
- ALL - Display all message types.

Note: The *hbackup* program examines the following types of files:

- Regular files.
- Empty directories.
- Symbolic links.
- FIFO files

UNIX hard links are always sent to DFSMSShsm because only the DFSMSShsm host knows whether the backup version in the DFSMSShsm inventory is the most recent for that file name.

Specifying the environment configuration

Ensure that the environment variables are configured correctly for the Java *hbackup* program to run. They are:

CLASSPATH

Include the */usr/lpp/dfsms/bin/hbackup.jar* path in the CLASSPATH environment variable.

```
CLASSPATH="$CLASSPATH":/usr/lpp/dfsms/bin/hbackup.jar; export CLASSPATH="$CLASSPATH"
```


LIBPATH

Include the `/usr/lpp/dfsms/bin` path in the LIBPATH environment variable.

```
LIBPATH="$LIBPATH":/usr/lpp/dfsms/bin; export LIBPATH="$LIBPATH"
```

Example of how to code the Java *hbackup* command

The following is an example of the Java *hbackup* command.

```
//BPXBAKJ1 JOB (XXXX), 'BACKUP JOB', NOTIFY=&SYSUID, MSGLEVEL=(1,1),
// MSGCLASS=H
//*****
/* Search directory sub-tree for files in need of Incremental backup
/* using Java hbackup program.
//*****
//STEP1 EXEC PGM=BPXBATCH, REGION=0M
//SYSPRINT DD SYSOUT=*
//STDOUT DD SYSOUT=*
//STDERR DD SYSOUT=*
//STDENV DD DUMMY
//STDPARM DD *
SH /bin/sh -c '
set -x;
CLASSPATH="$CLASSPATH":/usr/lpp/dfsms/bin/hbackup.jar ;
export CLASSPATH="$CLASSPATH" ;
LIBPATH="$LIBPATH":/usr/lpp/dfsms/bin ;
export LIBPATH="$LIBPATH" ;
su ;
java hbackup -v -o --test /test/ ;
'
/*
```


Chapter 74. UPDTCDS Command: Updating the control data sets

The UPDTCDS command allows you to update specific fields within the control data sets. Specifically, it enables you to update the expiration date of copy pool dump copies. When the UPDTCDS command is issued, DFSMSHsm updates all of the corresponding control data set records with the specified value.

Syntax of the UPDTCDS command

```

➤ UPDTCDS — COPYPOOL — (cpname) — VERSION( version) —➤
      |
      +— CP —

```

```

➤ DUMPEXPIRATION(DUMPCCLASS( class) NEWDATE( newdate)) —➤

```

Required parameters of the UPDTCDS command

This topic describes the required parameter of the UPDTCDS command.

COPYPOOL: Specifying the name of the copy pool

Explanation

COPYPOOL(*cpname*) is a required parameter that identifies the name of the copy pool that is to be modified.

VERSION: Specifying the version

Explanation

VERSION(*version*) is a required parameter that identifies the version of the specified copy pool.

DUMPEXPIRATION: Indicating that the copy pool dump expiration is to be modified

Explanation

DUMPEXPIRATION is a required parameter that indicates that the expiration date for the specified dump class is to be modified to the specified new date, as follows:

DUMPCCLASS(*class*)

Specifies the dump class that is to be modified. It is required.

NEWDATE(*newdate*)

Specifies the new expiration date. The date must be equal to or later than the date on which the command is issued. The new date format is *yyyy/mm/dd*, where

yyyy

Is the year

mm

Is the month

dd

Is the day of the month.

A leading zero is not required for a one-digit month or day.

Example of how to code the UPDTCDS command

The following example shows how to code the UPDTCDS command.

Note: Any values specified here are only examples and should not be interpreted as the values to be used for your system.

Updating the copy pool dump expiration date

In this example, the UPDTCDS command updates the expiration date for copy pool COPYPOOL1, version 124, class ONSITE to 2050/12/31.

```
UPDTCDS COPYPOOL(COPYPOOL1) VERSION(124) DUMPEXPIRATION(DUMPCCLASS(ONSITE) NEWDATE(2050/12/31))
```

You can use the LIST COPYPOOL(*cpname*) command to display the current versions, dump classes and expiration dates for existing dump copies associated with a copy pool.

Chapter 75. Using the AUDIT command

This topic contains diagnosis, modification, and tuning information.

For control data sets, the AUDIT command detects, reports, diagnoses, and (in many cases) provides repairs for discrepancies within and between the migration control data set (MCDS); the backup control data set (BCDS); the offline control data set (OCDS); VTOCs of DASD volumes; TTOCs for DFSMSHsm tape volumes; and the computer system catalogs, including the master catalog and user catalogs.

For common queues, the AUDIT command enables DFSMSHsm to dynamically correct inconsistencies with minimal impact on processing. Common queues have interrelated entries that may become corrupted due to abends and unexpected losses of connectivity. A corrupted common queue may cause certain requests to not be processed. DFSMSHsm automatically corrects some inconsistencies, but for others, it is necessary to issue the AUDIT COMMONQUEUE command.

These topics can help you make the best use of the AUDIT command for your application. In topics one and two, for each of the AUDIT options, we describe the checking done, and which error conditions can be detected by using that option. For each reported condition, AUDIT may be able to determine a repair action. The actual form of the repair commands, if any, depends on the control information involved.

The options described in [“Using the enhanced audit command to audit DFSMSHsm control information” on page 1256](#) are enhancements to the original AUDIT command described in [“Using the original AUDIT command to audit DFSMSHsm data set information” on page 1280](#). [Table 70 on page 1255](#) compares the information driving the AUDIT options.

Table 70. Comparison of AUDIT Options		
Driving Information	Enhanced AUDIT Option	Original AUDIT Option
Aggregate backup records	ABARSCONTROLS	None
Copy pool records	COPYPOOLCONTROLS(<i>cpname</i>)	None
Specific data set names	None	DATASET NAMES LEVELS
Catalog entries	None	MASTERCATALOG USERCATALOG
Backup Control Data Set	DATASETCONTROLS(BACKUP)	BCDS
Migration Control Data Set	DATASETCONTROLS(MIGRATION)	MCDS
Tape table of contents (ML2 volume)	DIRECTORYCONTROLS	VOLUMES OCDS(ML2) OCDS(ALL)
Tape table of contents (backup volume)	DIRECTORYCONTROLS	BACKUPVOLUMES BACKUPTYPE OCDS(DAILY) OCDS(SPILL) OCDS(ALL)
Volume table of contents (DASD VTOC)	DIRECTORYCONTROLS	VOLUMES
DFSMSHsm images of user data sets	MEDIACONTROLS MEDIACONTROLS(SDSP)	None
Backup-volume records	VOLUMECONTROLS(BACKUP)	None
Common queues	COMMONQUEUE	None

Table 70. Comparison of AUDIT Options (continued)

Driving Information	Enhanced AUDIT Option	Original AUDIT Option
Migration-volume records	VOLUMECONTROLS(MIGRATION)	None
Eligible-volume records	VOLUMECONTROLS(RECOVERABLE)	None

“Error codes (*ERR) and diagnosis” on page 1295 describes all the conditions reported by AUDIT, as well as repair actions taken and diagnostic suggestions for those cases where AUDIT cannot resolve a discrepancy.

“Unexpected errors during an audit” on page 1337 identifies errors that may occur to prevent AUDIT from completing its checking.

“Other considerations when using the AUDIT command” on page 1340 presents considerations for using AUDIT in a multiple-DFSMSHsm-host environment and suggestions for printing output.

Note: If AUDIT is executing and backup of the DFSMSHsm control data sets has been started (by the BACKVOL CDS command or AUTOBACKUP), all DFSMSHsm functions on the host that started this backup are halted until the AUDIT function and the backup of the DFSMSHsm CDS’s have completed.

Using the enhanced audit command to audit DFSMSHsm control information

Table 71 on page 1256 lists the AUDIT commands audit and report errors within DFSMSHsm controls and where to find more information.

Table 71. AUDIT Commands that Audit and Report Errors	
If the command you are interested in is. . .	Then see. . .
AUDIT ABARSCONTROLS (agname)	“Using the AUDIT ABARSCONTROLS command” on page 1257
AUDIT COMMONQUEUE	“Using the AUDIT COMMONQUEUE command” on page 1258
AUDIT COPYPOOLCONTROLS	“Using the AUDIT COPYPOOLCONTROLS command” on page 1259
AUDIT DATASETCONTROLS (BACKUP)	“Using the AUDIT DATASETCONTROLS (BACKUP) command” on page 1260
AUDIT DATASETCONTROLS (MIGRATION)	“Using the AUDIT DATASETCONTROLS (MIGRATION) command” on page 1261
AUDIT DIRECTORYCONTROLS VOLUMES	“Using the AUDIT DIRECTORYCONTROLS command” on page 1263
AUDIT MEDIACONTROLS (CLOUD)	“Using the AUDIT MEDIACONTROLS (CLOUD) command” on page 1267
AUDIT MEDIACONTROLS (SDSP) VOLUMES	“Using the AUDIT MEDIACONTROLS (SDSP) VOLUMES command” on page 1268
AUDIT MEDIACONTROLS (VOLUMES)	“Using the AUDIT MEDIACONTROLS VOLUMES command” on page 1269
AUDIT VOLUMECONTROLS (BACKUP)	“Using the AUDIT VOLUMECONTROLS (BACKUP) command” on page 1272
AUDIT VOLUMECONTROLS (MIGRATION)	“Using the AUDIT VOLUMECONTROLS (MIGRATION) command” on page 1273

Table 71. AUDIT Commands that Audit and Report Errors (continued)

If the command you are interested in is. . .	Then see. . .
AUDIT VOLUMECONTROLS (RECOVERABLE)	“Using the AUDIT VOLUMECONTROLS (RECOVERABLE) command” on page 1274

When the FIX parameter is specified, AUDIT often completes repairs. There may be times, however, when you want to audit without fixing errors. During an audit, the AUDIT function serializes control records, which allows it to compare several sources of information. While the control records are serialized, they are static and the comparisons are valid. Once the audit has completed, the system returns to its normal dynamic mode, and these comparisons may no longer be valid.

Your decision concerning the use of the FIX parameter to repair errors depends mostly upon two factors:

- First, how busy is DFSMSHsm? The higher the activity, the faster your audit report becomes obsolete.
- Second, what is the scope of your audit? If you just want to check the status of one DASD volume, then running AUDIT without the FIX parameter will be helpful. If you audit many controls, your report becomes obsolete quickly.

When auditing a common queue, FIX does not create the same concerns as when auditing other controls. In general, FIX should always be specified when auditing a common queue.

When using the AUDIT functions listed in topic One, you receive important information in the report header and the report body:

- The header of your audit report confirms the performed audit and shows the time, the date, and the system identification. The second line of the report header reflects the AUDIT command and parameters that you have entered.
- The body of your audit report normally consists of two lines printed for each detected error. The first line gives you the error code and a brief description of the error. The lines immediately following the error code list the appropriate repair commands whenever the AUDIT function is able to identify them.

AUDIT COMMONQUEUE does not generate a report. The only output of AUDIT COMMONQUEUE is message ARC1544I.

During an audit, AUDIT executes the following sequence:

1. DETECTION: AUDIT checks for inconsistent sources of information that might prevent DFSMSHsm from retrieving a data set (or group of data sets), or accessing a DFSMSHsm-owned volume.
2. DIAGNOSIS: AUDIT attempts to refer to one or more additional sources of information (a reality outside of the conflicting sources), when it detects an inconsistency. In many instances, AUDIT is able to list either a repair action (FIX) or a recommendation for further diagnosis of the detected error.
3. FIX: AUDIT initiates repair action, when you specify the FIX parameter and when AUDIT determines the appropriate FIX.

You can determine whether repair action has been initiated by examining two items on the audit report. When FIX is specified for a common queue, all detected errors will be fixed. No specific information is given concerning each error.

1. Was the FIX parameter specified?
2. Did AUDIT list the appropriate repair action?

Using the AUDIT ABARSCONTROLS command

When you specify AUDIT ABARSCONTROLS (*agname*), AUDIT makes the following checks for each ABR (Q) record in the BCDS (or for only those ABR records associated with the specified aggregate group name):

- If no ABR records are found for a specified aggregate group, AUDIT reports ***ERR 166**.
- If no ABR record is found at all, AUDIT reports ***ERR 167**.

- If the control file indicated in an ABR record is not cataloged, AUDIT reports ***ERR 168**.
- If at least one type of data file (DFSMSdss or internal) is present for an ABR record,
 - If a DFSMSdss data file is indicated, but the file is not cataloged, and it was created by DFSMSHsm, AUDIT reports ***ERR 168**;
 - If an internal data file is indicated, but the file is not cataloged, and it was created by DFSMSHsm, AUDIT reports ***ERR 168**.
- If no data file is present for an ABR record, AUDIT reports ***ERR 165**.
- If an instruction/activity log file is indicated, but the file is not cataloged, and it was created by DFSMSHsm, AUDIT reports ***ERR 168**.

The FIX or NOFIX option is ignored for ABARSCONTROLS since only diagnosis messages are issued.

Diagnosis and repair action for AUDIT ABARSCONTROLS

When you request AUDIT ABARSCONTROLS, AUDIT does not fix any of the errors detected. See “Error codes (*ERR) and diagnosis” on page 1295 for some troubleshooting hints for any reported conditions.

Using the AUDIT COMMONQUEUE command

When you specify the AUDIT COMMONQUEUE parameter, AUDIT scans the entries in the common queue. When you specify AUDIT COMMONQUEUE(RECALL), AUDIT scans the entries in the Common Recall Queue (CRQ) for logical inconsistencies within the structure.

Diagnosis and repair action for AUDIT COMMONQUEUE(RECALL)

The following indicate that the CRQ may be corrupted:

- Message ARC1506E was issued.
- Message ARC1187E was issued.
- Recall requests are unexpectedly not being selected for processing.

Note: If recall requests are unexpectedly not being selected for processing, issue the QUERY ACTIVE command and examine message ARC1541I. This message displays the factors that affect the selection of requests from the CRQ. A status of anything other than CONNECTED or a hold level of anything other than NONE are probable reasons why certain recall requests are not being selected.

Note: An unexpected loss of connectivity to the CRQ may also introduce errors, but in most cases DFSMSHsm automatically corrects those errors. When this occurs, you may see the multiple issuance of message ARC1102I. Message ARC1102I is issued when DFSMSHsm attempts to recall a data set that has already been recalled. It is normal for this message to occur at the time of a loss of connectivity and it does not indicate a problem with the CRQ list structure.

If you believe that the CRQ has been corrupted, issue the AUDIT COMMONQUEUE(RECALL) FIX command. When this command is issued, DFSMSHsm scans the entries in the CRQ and attempts to correct logical inconsistencies within the structure. AUDIT cannot correct all types of errors and may report that zero errors were detected even though there are errors in the CRQ structure.

Note: The following parameters are not used with AUDIT COMMONQUEUE:

- ALL
- OUTDASE | SYSOUT | TERMINAL
- REPORT(ALL | ERRORS)
- SERIALIZATION(DYNAMIC | CONTINUOUS)

If after issuing the AUDIT command, message ARC1506E or message ARC1187E occurs repeatedly, or certain data sets are unexpectedly not being selected for recall processing, refer to [Chapter 4, “Other space management considerations,”](#) on page 127 for more information about correcting errors within the CRQ.

Using the AUDIT COPYPOOLCONTROLS command

When you specify AUDIT COPYPOOLCONTROLS (*cpname*), AUDIT reads the fast replication control data set records to confirm the existence and accuracy of all DFSMSHsm records associated with the copy pool. No orphaned records are identified.

You can use the AUDIT COPYPOOLCONTROLS command to verify that if a Fast Replication copy pool version indicates that there is an associated catalog information data set, then that corresponding catalog information data set exists on an ML1 volume.

When you specify AUDIT COPYPOOLCONTROLS without a specific copy pool name, AUDIT reads sequentially through all existing fast replication records and confirms the existence and accuracy of all DFSMSHsm records associated with copy pools. The AUDIT function detects and reports orphaned records and other discrepancies. When AUDIT can determine the appropriate solution for a discrepancy, the solution is included in the AUDIT output. All errors detected by AUDIT are identified with an error number.

The following checks are reported:

- If no copy pool 'F' (FRB) record exists, AUDIT reports ***ERR 106**.
- If the specified copy pool does not exist, AUDIT reports ***ERR 170**.
- If the number of dump version does not match the number of version found, AUDIT reports ***ERR 171** and ***ERR 172**.
- If the number of 'H' (FRVP) records indicated does not match the number of records found, AUDIT reports ***ERR 173**.
- If no 'F' (FRB) records for a copy pool are found and an orphaned 'H' (FRVP) record exists, AUDIT reports ***ERR 174**.
- If the 'H' (FRVP) record indicates 'Prepare' and the corresponding 'F' (FRB) record does not, AUDIT reports ***ERR 175**.
- If a 'H' (FRVP) record exists for a version and the corresponding 'F' (FRB) record is not found, AUDIT reports ***ERR 176**.
- If no 'I' (FRTV) record exists for the specified VOLUME *volser*, AUDIT reports ***ERR 177**.
- If the associated FRSV (J) record, as indicated in the FRTV(I) record, is not found, ***ERR 178** is issued.
- If wrong target volume or source volume is indicated in the 'H' (FRVP) and 'I' (FRTV) records, AUDIT reports ***ERR 179**.
- If no 'J' (FRSV) record exists for specified VOLUME *volser*, Copy pool *cpname*, and VERSION *vsr*, AUDIT reports ***ERR 180**.
- If no 'F' (FRB) record exists for *cpname*, AUDIT reports ***ERR 181**.
- If no 'P' (MCP) record exists for VOLUME *volser*, AUDIT reports ***ERR 182**.
- If no VOLUME *volser* from the 'J' (FRSV) record L0 field exists in corresponding 'H' (FRVP) record, AUDIT reports ***ERR 183**.
- If an orphaned 'J' (FRSV) record is detected, AUDIT reports ***ERR 184**.
- If copy pool *cpname* not found in 'J' (FRSV) record copy pool array for VOLUME *volser*, AUDIT reports ***ERR 185**.
- If an orphaned 'K' (FRD) record is detected and copy pool *cpname* no longer exists, AUDIT reports ***ERR 186**.
- If a 'K' (FRD) record exists and a 'F' (FRB) record indicates no dump copy, AUDIT reports ***ERR 187**.
- If no 'K' (FRD) record exists and a 'F' (FRB) record for copy pool *cpname* indicates that it does exist, AUDIT reports ***ERR 188**.
- If no 'H' (FRVP) record exists for the specified copy pool name, AUDIT reports ***ERR 189**.
- If the specified volume is not found in any associated 'H' (FRVP) record for the specified copy pool name, AUDIT reports ***ERR 190**.

- If an associated P (MCP) record for a volume *volser* that is in a storage group that is defined in a copy pool does not indicate the volume is part of a copy pool, ***ERR 191** is reported.
- If an FRSV record is found DFSMSHsm attempts to read associated FRBs records, if any. For each FRB record found , AUDIT reports ***ERR 192**.
- If the DFSMSHsm records indicate that volume *volser* is a copy pool volume but no FRSV (J) record exists, ***ERR 193** is issued.
- If the copy pool record indicates that the catalog information was captured but a data set containing this information was not found, AUDIT reports ***ERR 194**.
- If the copy pool record indicates that the catalog information was captured but the corresponding catalog information data set indicates that it contains catalog information for a different copy pool backup version, AUDIT reports ***ERR 195**.
- ***ERR 200** indicates that there was an unexpected error while trying to process the catalog information data set. This message should be preceded by an ARC1812I.
- ***ERR 202** is issued for each FRTV record that is not associated with any of the copy pool records that were found during the audit.
- ***ERR 204** is issued if auditing determines that a single FRTV (target volume) record is associated with more than one source volume. This error is issued for each copy pool version in which the error was detected.
- ***ERR 215** is issued if a CLOUD DGN record has any inconsistent entries as CLOUD dump.
- ***ERR 217** is issued if the object number in a cloud DGN record does not match the object number of the cloud object prefix.
- ***ERR 218** is issued if any dump copy in a TAPE DGN record is CLOUD type.
- ***ERR 219** is issued if the corresponding cloud dump object of a CLOUD DGN record can not be found.

Diagnosis and repair action for AUDIT COPYPOOLCONTROLS

See “Error codes (*ERR) and diagnosis” on page 1295 for troubleshooting hints for reported conditions and to help determine if corrective action is taken by audit processing.

Using the AUDIT DATASETCONTROLS (BACKUP) command

When you specify the AUDIT DATASETCONTROLS (BACKUP) command, the following checks are performed for each valid data set (B) record in the BCDS (or for only those specified by the DS NAMES or LEVELS parameter):

- If a backup version (C) record indicates the version is on a given tape backup volume, but the TTOC (T) base record for that volume does not exist, AUDIT reports ***ERR 17**.
- If the latest backup version (C) record indicates a version number that is higher than the version number in the MCBGEN field of the associated data set backup (B) record, AUDIT reports ***ERR 35**.
- If there is a discrepancy between the MCBVER and MCCGEN fields of a backup version entry, AUDIT reports ***ERR 36**.
- If a backup version (C) record has a sequence number that is not the same as the sequence number of the associated retained version (Z) record, AUDIT reports ***ERR 37**.
- If a retained version (Z) record exists and the associated backup version (C) record's sequence number field indicates no retained version (Z) records exist, AUDIT reports ***ERR 38**.
- If a retained version (Z) record exists and the associated data set (B) record does not contain a value greater than zero in the sequence number field, AUDIT reports ***ERR 38**.
- If a data set record (or retained backup record) refers to a backup version for which there is no version record, AUDIT reports ***ERR 39**.
- If a data set record refers to a backup version record (or retained backup record) that does not refer back to the data set record, AUDIT reports ***ERR 39**.

- If the entry in a data set record's (or retained backup record's) version index and the backup version record itself indicate that version is on a certain backup volume, but there is no backup volume (**X**) record for that volume serial number, AUDIT reports ***ERR 41**. (The error can apply to any of the tape volumes spanned by a multivolume backup version.)
- If the entry in a data set record's version index and the backup version record itself indicate that version is on a certain migration volume, but there is no migration volume (**V**) record for that volume serial number, AUDIT reports ***ERR 42**.
- If a backup version is found on its indicated migration volume, but no move (**M**) record exists for that version, AUDIT reports ***ERR 43**.
- If the entry in a data set record's (or retained backup record's) version index and the backup version record itself indicate that version is on a certain DASD backup volume, but the backup volume record indicates the volume is a tape (or vice versa), AUDIT reports ***ERR 44**.
- If the entry in a data set record's version index and the backup version record itself indicate that version is on a certain migration volume, but the volume is not mounted or (according to its VTOC) does not contain the backup version, AUDIT reports ***ERR 47**.
- If a specified data set name has no data set record, AUDIT reports ***ERR 48**.
- If the backup version record (B) indicates retained backup version records exist, but they are not found, AUDIT will report MSG 912.

When the scope of the audit is *not* limited by either DSNAMES() or LEVELS(), AUDIT makes additional checks:

- If an invalid record type is found in the BCDS, AUDIT reports ***ERR 30**.
- If a (**L**) record calls for backing up a migrated data set, but no MCDS record exists for that data set, AUDIT reports ***ERR 46**.

When you specify the AUDIT DATASETCONTROLS(BACKUP) parameter, AUDIT also checks to ensure version (C) records have an associated entry in a data set (B) record (or in a retained backup version (Z) record). The check is made for each version (C) record or for only those specified by the LEVELS parameter. A fully qualified backup version name can be used with the LEVELS parameter to process a single MCC record. If the backup version record refers to a BCDS data set (B) record that does not exist, or the (B) record exists but does not refer to the version (C) record, AUDIT reports ***ERR 40**.

If the backup version record refers to a retained version (Z) record that does not exist, or the (Z) record exists but does not refer to the backup version (C) record, AUDIT reports ***ERR 40**.

Important: Each FIXCDS command generated with this error is dependent on previous FIXCDS commands. If NOFIX is used, the offsets in the output data set are invalid.

Diagnosis and repair action for AUDIT DATASETCONTROLS (BACKUP)

When you request AUDIT DATASETCONTROLS (BACKUP):

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 30, 35, 36, 40, 41, 42, 43, 44, and 46.
- AUDIT recommends an additional diagnostic step for error messages (*ERR) 17 if the backup tape involved is a volume written in single file format. If you want AUDIT to continue its attempt to identify a repair action, you must specify an additional AUDIT command using the recommended parameter.
- For other reported conditions, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#) for some troubleshooting hints.

Using the AUDIT DATASETCONTROLS (MIGRATION) command

When you specify the AUDIT DATASETCONTROLS (MIGRATION) parameter, AUDIT makes the following checks for each valid data set (**D**) record in the MCDS (or for only those specified by the DSNAMES or LEVELS parameter):

- If a data set record indicates a valid migrated copy, but the data set is uncataloged, AUDIT reports ***ERR 03**.
- If a data set record indicates a valid migrated copy, but the data set's catalog entry does not refer to volume serial MIGRAT, AUDIT reports ***ERR 09**.
- If a data set record indicates the migration copy is on a given tape volume, but the TTOC base record (or indicated extension) does not exist, AUDIT reports ***ERR 17**.
- If a specified data set name has no data set record, AUDIT reports ***ERR 19**.
- If a data set record indicates that a migration copy is on a certain volume, but there is no migration volume record for that volume serial number, AUDIT reports ***ERR 20**. (The error could apply to any of the tape volumes spanned by a multivolume migration copy.)
- If a data set record indicates the migration copy is on a given DASD (or tape) migration volume, but the migration volume record indicates the volume is a tape (or DASD) volume, AUDIT reports ***ERR 21**.
- If a data set record created by HSM Release 3 or later indicating that a migration copy is *not* in an SDSP data set, but no alias (**A**) record exists for the migration copy, AUDIT reports ***ERR 22**.
- If a data set record indicates the existence of a VSAM-associations (**O**) record, but no such record exists, AUDIT reports ***ERR 23**.
- If a VSAM-associations record refers to a migrated VSAM component, but there is no catalog entry for the component, or the catalog entry is something other than a non-VSAM entry referring to single volume serial MIGRAT, AUDIT reports ***ERR 24**.
- If a VSAM-associations record refers to a migrated VSAM component, and its catalog entry properly indicates a migration copy, but no alias record exists for the VSAM component, AUDIT reports ***ERR 25**.
- If a data set record indicates the existence of a back up migrated data set (**L**) record, but no such record exists, AUDIT reports ***ERR 26**.
- If a migration volume's TTOC has no valid entry for a migration copy supposedly starting on that tape, or continuing from another tape, AUDIT reports ***ERR 29**. This error may occur if AUDIT DATASETCONTROLS is executed while migration is currently updating a given TTOC record. If this should occur, run AUDIT DATASETCONTROLS while migration is inactive.
- If a MCD record (D) indicates that a user data set has been encrypted but there is no MCDX record (3), AUDIT reports ***ERR 31**.
- If a MCD record (D) indicates that a user data set has associated 'Retained' MCA records, but there is a mismatch between the number of 'Retained' MCA records the MCD reports and the number of actual copies, AUDIT reports ***ERR 33**.

When the scope of the audit is *not* limited by either DSNAMES() or LEVELS(), AUDIT makes additional checks:

- If a VSAM-associations record refers to a data set record that does not exist, or that does not refer back to the VSAM-associations record, AUDIT reports ***ERR 27**.
- If an alias record refers to a data set record that does not exist, AUDIT reports ***ERR 28**.
- If an invalid record type is found in the MCDS, AUDIT reports ***ERR 30**.
- If a MCDX record (3) refers to a MCD record (D) that does not exist, AUDIT reports ***ERR 32**.

Diagnosis and repair action for AUDIT DATASETCONTROLS (MIGRATION)

When you request AUDIT DATASETCONTROLS (MIGRATION):

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 20, 21, 22, 24, 25, 26, 27, 28, 30, and 32.
- AUDIT recommends an additional diagnostic step for error messages (*ERR) 17 and 29 if the migration tape involved is a volume written in single file format. If you want AUDIT to continue its attempt to identify a repair action, you must specify an additional AUDIT command using the recommended parameter.

- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295.](#)

Using the AUDIT DIRECTORYCONTROLS command

When you specify the DIRECTORYCONTROLS VOLUMES(*volser*...) parameter, for each volume serial number specified AUDIT cross-checks information in the volume's directory (a VTOC or TTOC) with control records in the MCDS or BCDS, whichever applies in a given situation. AUDIT then reports any discrepancies found.

To determine the type of directory for each volume serial specified, AUDIT first checks for a volume record. If no migration volume record or backup volume record exists for a specified volume serial number, AUDIT reports ***ERR 110**. When AUDIT finds a volume record, its content determines what further checking is done.

When AUDIT can determine the appropriate fix for an error, the fix is included in the list report.

You can also use the AUDIT DIRECTORYCONTROLS command to verify that the catalog information data sets found on ML1 volumes correspond to a valid copy pool version.

Errors detected when using AUDIT DIRECTORYCONTROLS for a DASD migration volume

For each volume record indicating a DASD migration volume, AUDIT makes the following checks for each DSCB having the attributes DFSMSHsm assigns to its data sets (other DSCBs are ignored):

- If the first qualifier of a data set name indicates a migration copy, but there is no alias record in the MCDS, AUDIT reports ***ERR 140**.
- If the DSCB for a migration copy or backup version indicates an empty data set (TTR of 0), AUDIT reports ***ERR 141**.
- If the first qualifier of a data set name indicates a backup version, but there is no backup version record in the BCDS, AUDIT reports ***ERR 142**.
- If the backup version record corresponding to a DSCB does not indicate a version scheduled for deletion, and there is no move record for that version, AUDIT reports ***ERR 43**.
- If the first qualifier of a data set name indicates a VTOC copy for a certain volume, but there is no eligible volume (**P**) record for that volume in the BCDS, AUDIT reports ***ERR 143**.
- If the eligible volume record corresponding to a VTOC copy DSCB indicates
 - The volume has not been backed up; or
 - Its related VTOC copy data sets are on other volumes; or
 - Its VTOC copy data set is not (determined by the qualifiers in its name) the one we started from
 AUDIT reports ***ERR 144**.
- If the first qualifier of a data set name indicates a Dump VTOC copy for a certain volume, but there is no eligible volume record for that volume in the BCDS, AUDIT reports ***ERR 145**.
- If the eligible volume record corresponding to a Dump VTOC copy DSCB indicates there are no valid dump generations for that volume, AUDIT reports ***ERR 146**.
- If the eligible volume record indicates there is at least one dump generation, but no dump generation record refers to the Dump VTOC copy we started from, AUDIT reports ***ERR 146**.
- When there is a catalog entry for the dsname indicating it is migrated, but the dsname's first qualifiers are not recognized as DFSMSHsm generated,
 - If there is no valid data set record in the MCDS, AUDIT reports ***ERR 147**.
 - If there is a valid data set record in the MCDS referring to the volume we started from, but indicating the migration copy is in an SDSP data set, AUDIT reports ***ERR 148**.
 - When there is a valid data set record in the MCDS, but it indicates a migration copy on some other DASD volume (not in an SDSP data set),

- If the other volume is mounted, but the dsname cannot be found in its VTOC, AUDIT reports ***ERR 147**.
- If the other volume is mounted and the dsname is found in its VTOC, AUDIT reports ***ERR 149**.
- If the other volume is not mounted, AUDIT reports ***ERR 148**.
- When there is a valid data set record in the MCDS, but it indicates a migration copy on a tape volume,
 - If a valid TTOC entry can be found for that tape volume, AUDIT reports ***ERR 149**.
 - If no valid TTOC entry can be found, AUDIT reports ***ERR 147**.
- If the catalog information data set was found on an ML1 volume but the data set is not cataloged, AUDIT reports ***ERR 196**.
- If there is no corresponding Fast Replication backup version for the catalog information data set, AUDIT reports ***ERR 197**.
- ***ERR 198** indicates that the copy pool backup version specified by the catalog information data set indicates that the catalog information was not captured. This catalog information data set does not correspond to a valid existing copy pool backup version.
- If the copy pool version that the catalog information data set indicates that it is associated to is not the catalog information data set that the copy pool version record indicates, AUDIT reports ***ERR 199**.
- ***ERR 200** indicates that there was an unexpected error while trying to process the catalog information data set. This message should be preceded by an ARC1812I message.
- If the catalog locate for the catalog information data set indicates that this data set is cataloged on a different volume, AUDIT reports ***ERR 201**.

Diagnosis and repair action for AUDIT DIRECTORYCONTROLS for DASD

When you request AUDIT DIRECTORYCONTROLS VOLUMES(*volser...*) for a DASD:

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 43, 141, 142, 143, 144, 145, 146, 147, and 149.
- AUDIT may recommend an additional diagnostic step for error messages (*ERR) 140 and 148. If you want AUDIT to continue its attempt to identify a repair action, you must specify an additional AUDIT command using the recommended parameter.
- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Errors detected when using AUDIT DIRECTORYCONTROLS for a tape volume

For each volume record indicating a *tape migration* volume, AUDIT makes the following checks:

- If no TTOC base record exists, AUDIT reports ***ERR 17**.
- When the TTOC exists, AUDIT makes the following checks for each valid data set entry in that TTOC:
 - If there is a catalog entry indicating a migrated data set, but no data set record in the MCDS, AUDIT reports ***ERR 113**.
 - If there is a catalog entry indicating a migrated data set, and a data set record indicating a valid migration copy on *some other* volume:
 - If the data set record indicates that other volume is a tape, but there is no TTOC base record for that other volume; or
 - If the data set record indicates that other volume is a DASD volume, but its VTOC has no DSCB for the migration copy

AUDIT reports ***ERR 115**.

- If a TTOC entry is not flagged as valid, but has an alias record in the MCDS, AUDIT reports ***ERR 114**.

For each volume record indicating a *tape backup* volume, AUDIT makes the following checks:

- If no TTOC base record exists, AUDIT reports ***ERR 17**.

- When the TTOC exists, AUDIT makes the following checks for each valid data set entry in that TTOC:
 - If, in the BCDS, a backup version record exists, but
 - No data set record exists; or
 - The existing data set record does not refer to the backup version record
 AUDIT reports ***ERR 116**.
 - If the TTOC entry indicates either a VTOC or VCAT copy data set for a given volume:
 - If there is no eligible volume record for that volume, AUDIT reports ***ERR 117**
 - If the existing eligible volume record does not refer to the VTOC or VCAT copy we started from, AUDIT reports ***ERR 118**.
- If a TTOC entry is not flagged as valid, but has a backup version record in the BCDS, AUDIT reports ***ERR 116**.

For either a migration or backup volume, AUDIT makes the following checks, after finding the base TTOC record:

- If the alternate volume serials for duplex volumes do not match between the TTOC and either the MCT or MCV, AUDIT reports ***ERR 58**.
- If one or more TTOC extension records are missing from the TTOC set, AUDIT reports ***ERR 111**.
- If there is an extraneous TTOC extension record, AUDIT reports ***ERR 112**.
- When the TTOC base record indicates the first entry, either a migrated data set or a backup version continues (spans) from another tape volume, referred to as the predecessor volume
 - If the data set entry is INVALID, AUDIT reports ***ERR 125**.
 - If there is no TTOC for the predecessor volume and the data set entry is VALID, AUDIT reports ***ERR 119**.
 - If the TTOC for the predecessor volume is found, but does not indicate the volume being audited as a successor, and the data set entry is VALID, AUDIT reports ***ERR 120**.
 - If the last data set in the predecessor's TTOC is not the same dsname as the first data set in the TTOC of the volume being audited, AUDIT reports ***ERR 121**.
- When the TTOC base record indicates the last entry, either a migrated data set or a backup version continues (spans) to another tape volume, referred to as the successor volume
 - If the data set entry is INVALID, AUDIT reports ***ERR 125**.
 - If there is no TTOC for the successor volume and the data set entry is VALID, AUDIT reports ***ERR 122**.
 - If the TTOC for the successor volume is found, but does not indicate the volume being audited as a predecessor, and the data set entry is VALID, AUDIT reports ***ERR 123**.
 - If the first data set in the successor's TTOC is not the same dsname as the last data set in the TTOC of the volume being audited, AUDIT reports ***ERR 124**.
- If the total valid block count, as determined by the AUDIT function, differs from the value currently reflected in the in the base TTOC record, the AUDIT function corrects the total valid block count in the base TTOC record. This correction can also occur in predecessor or successor TTOC base records.

Diagnosis and repair action for AUDIT DIRECTORYCONTROLS for tape

When you request AUDIT DIRECTORYCONTROLS VOLUMES(*volser*...) for a tape volume:

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 111, 112, 113, 114, 115, 116, 118, and 125.
- AUDIT may recommend an additional diagnostic step for error message (*ERR) 17, 119, 120, 121, 122, 123, and 124. If you want AUDIT to continue its attempt to identify a repair action, you must specify an additional AUDIT command using the recommended parameter.

- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis”](#) on page 1295.

Using the AUDIT FILECONTROLS(BACKUP) command

When you specify the AUDIT FILECONTROLS(BACKUP) command, the following checks are performed for each valid UNIX file (B) record in the BCDS (or for only those specified by the FILE or FILELEVEL parameter):

The following checks are performed for UNIX file records in BCDS:

If the latest backup version MCC record indicates a version number that is higher than the version number in the MCBGEN field of the associated MCB record, AUDIT reports ***ERR 35**.

If there is a discrepancy between the MCBVER and MCCGEN fields of a backup version entry, AUDIT reports ***ERR 36**.

If a backup version MCC record has a sequence number that is not the same as the sequence number of the associated retained version record, AUDIT reports ***ERR 37**.

If a retained version record exists and the associated backup version record's sequence number field indicates no retained version records exist, AUDIT reports ***ERR 38**.

If MCB record (or retained backup record) refers to a backup version for which there is no version record, AUDIT reports ***ERR 39**.

If MCB record refers to a backup version record (or retained backup record) that does not refer to the MCC record, AUDIT reports ***ERR 39**.

If the entry in a data set record's version index and the backup version record itself indicate that version is on a certain migration volume, but there is no migration volume record for that volume serial number, AUDIT reports ***ERR 42**.

If a backup version is found on its indicated migration volume, but no move record exists for that version, AUDIT reports ***ERR 43**.

If the entry in MCC's index and the backup version record itself indicate that version is on a certain migration volume, but the volume is not mounted or (according to its VTOC) does not contain the backup version, AUDIT reports ***ERR 47**.

If a specified file path has no MCB, AUDIT reports ***ERR 48**.

If UFN record is found in BCDS but has no associated MCB record, AUDIT reports ***ERR 212**.

If UFN record is not found, AUDIT reports ***ERR 213**.

If UFN record contains valid hash value for parent, child, sibling or last child but UFN record is missing for it, AUDIT reports ***ERR 214**.

The FIX option in AUDIT FILECONTROLS has no effect with FILE and FILELEVEL keywords.

Note: When AUDIT FILECONTROLS runs, it is normal to see the messages such as *ERR 223, *ERR 224, *ERR 225, *ERR 228, and *ERR 229. Read the action needed for each of these messages below and take the actions accordingly. Also it is unusual to see the messages *ERR 226 and *ERR 227. If found, take the right action as suggested.

When AUDIT FILECONTROLS is not running with FIX option (default is NOFIX) and it is found that a UFN record does not have the previous sibling key, then the audit issues ***ERR 223** that contains a PATCH command to fix such nodes. The recommended way to fix such records is by running an AUDIT FILECONTROLS with FIX option. When the count of these error messages reaches a limit of 1000, then audit stops abruptly and issues *ERR 224. This is an indication that there are many UFN nodes without previous sibling reference and user may choose to re-run the AUDIT FILECONTROLS with FIX option.

When AUDIT FILECONTROLS or AUDITFILECONTROL FILELEVEL, is not running with FIX option (default is NOFIX) and a UFN record is found in soft deleted state (UFN_INVALID = ON), the same is reported by the error message ***ERR 225**. When the count of these error messages reaches a limit of 1000, then audit

stops abruptly and issues *ERR 224. In that case, it is recommended to run AUDIT FILECONTROLS with FIX option to clean-up such records from BCDS.

When the count of *ERR 223, or *ERR 225 messages reaches a limit of 1000, the audit stops abruptly and issues ***ERR 224** message recommending AUDIT FILECONTROLS run with FIX option.

If there was a CDS update failure while deleting a UFN record, such a node is identified during the AUDIT process and reported in the form of ***ERR 226** error messages. The further Delete or Add (backup) requests for the Unix files associated to those UFN nodes are handled appropriately. No immediate action required by the user unless there are many such messages.

The *ERR 226 error messages are not usually expected as they indicate CDS update failures. If there are more than 50 such messages, then the audit reports ***ERR 227** error messages. Once this condition is detected, Audit stops issuing *ERR 226 messages but instead, it continues to issue *ERR 227 messages. The count of *ERR 227 messages will be useful to know the number of such failures. Contact IBM support when *ERR 227 is detected.

When AUDIT FILECONTROLS runs with FIX option and updates a UFN record with its previous sibling reference, then the audit issues an informational message ***ERR 228** to report the same to the user. The message text contains the key of UFN record that is updated.

When AUDIT FILECONTROLS runs with FIX option and removes a UFN record that was marked soft delete, the audit issues an informational message ***ERR 229** to report the same to the user. The message text contains a 60-character UFN file / directory name (shortened if original name was more than 60 characters) associated to the UFN node that is now removed from BCDS.

Using the AUDIT MEDIACONTROLS (CLOUD) command

When you specify the AUDIT MEDIACONTROLS (CLOUD(*cloud_network_connection_name*)) parameter, AUDIT checks the control information located in each APPMETA object for each possible DFSMSHsm image of a user data set in each of the HSMplex containers in the specified cloud. AUDIT then produces a list of errors.

Errors detected when using AUDIT MEDIACONTROLS for cloud storage

DFSMSHsm checks each container that matches the name SYSZARC.<plexname>.MIG.<yyyyddd> for migrated data set images and the name SYSZARC.<plexname>.DMP.<yyyyddd> for dump data set images. For each pseudo-directory that is an object name with a forward-slash init, DFSMSHsm checks that group of objects for validity as a migrated data set image or a dump data set image.

For migrated data set images:

- If no object ending with APPMETA (which contains the CDD record) can be found,
 - Using the object prefix the MCDS is checked
 - If there is no alias record in the MCDS then AUDIT reports ***ERR 210** with the container and prefix names
 - If there is no data set record in the MCDS then AUDIT reports **ERR 210** with the container and prefix names
 - If there is a data set record in the MCDS:
 - If there is no catalog entry for the data set, AUDIT reports ***ERR 207**.
 - If a catalog entry indicates the data set is not migrated, AUDIT reports ***ERR 208**.
 - If a catalog entry indicates the data set is migrated, then AUDIT reports ***ERR 230** with the container and the user data set name
- Using the CDD information, the MCDS is checked.
 - If there is no record in the MCDs,
 - If there is no data set record in the MCDS:
 - If a catalog entry indicates the data set is migrated, AUDIT reports ***ERR 133**.

- If a catalog entry indicates the data set is *not* migrated, AUDIT reports ***ERR 208**.
- If a catalog entry indicates the data set is *not* migrated, AUDIT reports ***ERR 207**.
- If there *is* a data set record in the MCDS:
 - If there is no catalog entry for the data set, AUDIT reports ***ERR 207**.
 - If a catalog entry indicates the data set is not migrated, AUDIT reports ***ERR 208**.
 - If a catalog entry indicates the data set is migrated, but the data set record in the MCDS indicates the data set has been recalled, AUDIT reports ***ERR 133**.
 - If a catalog entry indicates the data set is migrated, but the data set record in the MCDS indicates the data set is migrated to ML1 or ML2, or is a different migration copy, AUDIT reports ***ERR 211**.
 - If both MCDF_CLD_CDA_TCT and MCDF_CLD_DIRECT flags are ON, AUDIT reports ***ERR 231** with the data set name.

For dump data set images:

Build the DGN record key based on the object prefix name, if no DGN record can be found using the key, AUDIT reports ***ERR 216**.

If the DGN record is found:

- If the object number in DGN record does not match the object number of the cloud object prefix, AUDIT reports ***ERR 217**.
- If the DGN record has any inconsistent entries as CLOUD dump, AUDIT reports ***ERR 215**.
- If the dump class record is not found based on the dump class name in the DGN record, AUDIT reports ***ERR 220**.
- If it is a copy pool DGN record:
 - If the FRSV record is not found, AUDIT reports ***ERR 193**.
 - If the FRSV record is found but dump copy number is 0, AUDIT reports ***ERR 180**.
 - If the FRB record is not found, AUDIT reports ***ERR 221**.
- If it is not a copy pool DGN record, read the MCP record using the volser:
 - If the MCP record is not found, or the DGN record prefix is not in the MCP record, AUDIT reports ***ERR 222**.

Diagnosis and repair action for AUDIT MEDIACONTROLS for CLOUD

When you request AUDIT MEDIACONTROLS FOR CLOUD:

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (***ERR 133, 207, 208, 210, 211, 216**)
- For other reported conditions and troubleshooting hints, see "ERROR codes (*ERR) and diagnosis".

Using the AUDIT MEDIACONTROLS (SDSP) VOLUMES command

When you specify AUDIT MEDIACONTROLS (SDSP) VOLUMES(*volser...*), for each specified volume AUDIT cross-checks its SDSP data set (if any) for consistency with the other DFSMSHsm control records to which it refers. (Note that VOLUMES is a required parameter.) AUDIT then produces a list of errors.

For each specified volume serial number, AUDIT makes the following checks:

- If there is no migration volume record or the record does not indicate a DASD ML1 volume, AUDIT reports ***ERR 50**.
- If the volume is mounted, but its VTOC does not contain a DSCB for an SDSP data set, AUDIT reports ***ERR 160**.
- For an existing SDSP data set, AUDIT makes the following checks for each migration copy's set of records:

- When the catalog entry indicates that the data set is migrated:
 - If there is no data set record in the MCDS or the data set record indicates an invalid copy, AUDIT reports ***ERR 147**.
 - If the data set record indicates a valid migration copy on some other mounted DASD volume, but that migration copy cannot be found or cannot be opened or does not have valid control information, AUDIT reports ***ERR 147**.
 - If the data set record indicates a valid migration copy elsewhere on this volume or on some other mounted volume, and that migration copy exists and can be opened,
 - If the time stamp for that migration copy is *earlier* than that for the copy in the SDSP we started from, AUDIT reports ***ERR 147**.
 - If the time stamp for that migration copy is *later* than that for the copy in the SDSP we started from, AUDIT reports ***ERR 149**.
 - If the data set record indicates a valid migration copy on a tape volume or on a DASD volume that is not mounted, AUDIT reports ***ERR 148**.
- If there is no catalog entry for the data set, AUDIT reports ***ERR 161**.
- If the catalog entry indicates the data set is *not* migrated, AUDIT reports ***ERR 162**.
- If the migration copy has one or more *gaps* in the sequence of data records, AUDIT reports ***ERR 163**.

Diagnosis and repair action for AUDIT MEDIACONTROLS (SDSP) VOLUMES

When you request AUDIT MEDIACONTROLS (SDSP):

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 147, 149, 161, and 162.
- AUDIT may recommend an additional diagnostic step for error message (*ERR) 148. If you want AUDIT to continue its attempt to identify a repair action, you must specify an additional AUDIT command using the recommended parameter.
- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Using the AUDIT MEDIACONTROLS VOLUMES command

When you specify the AUDIT MEDIACONTROLS VOLUMES(*volser...*) parameter without the SDSP qualifier, AUDIT checks control information in the common dataset descriptor (CDD, required at the start of every DFSMSHsm-generated copy) in each possible DFSMSHsm image of a user data set on the specified volumes. AUDIT then produces a list of errors.

To determine the type of media for each volume serial number specified, AUDIT first checks for a volume record.

- If no migration volume record and no backup volume record exists for a specified volume serial number, AUDIT reports ***ERR 110**.
- If the volume record indicates a level 0 (not DFSMSHsm-owned) volume, AUDIT reports ***ERR 110**.
- If a backup volume record exists, indicating a DASD volume, AUDIT reports ***ERR 50**.

When AUDIT finds a proper volume record, its content drives further checking.

Errors detected when using AUDIT MEDIACONTROLS for a DASD migration volume

For each specified volume serial number that identifies a mounted DASD migration volume, AUDIT makes the following checks for each data set with a DSCB having the attributes DFSMSHsm assigns to its data sets (other DSCBs are ignored):

- For a dsname with leading qualifiers indicating a *migration copy*:

- If AUDIT cannot identify the first record as a CDD, it reports ***ERR 150**.
- When the first record is a CDD:
 - If there is no data set record in the MCDS:
 - If a catalog entry indicates the data set is migrated, AUDIT reports ***ERR 133** (see also [Diagnosing Error 133](#)).
 - If a catalog entry indicates a nonmigrated data set, AUDIT reports ***ERR 149**.
 - If the data set record exists, but is not flagged as valid:
 - If a catalog entry indicates the data set is migrated, AUDIT reports ***ERR 133** (see also [Diagnosing Error 133](#)).
 - If there is no catalog entry, AUDIT reports ***ERR 149**.
 - If a catalog entry indicates a nonmigrated data set, AUDIT reports ***ERR 149**.
 - If there is a valid data set record in the MCDS:
 - If a catalog entry indicates the data set is migrated, but there is no alias record, AUDIT reports ***ERR 140**.
 - If a catalog entry indicates the data set is migrated, but the data set record in the MCDS refers to a different copy, AUDIT reports ***ERR 149**.
 - If there is no catalog entry, AUDIT reports ***ERR 151**.
 - If a catalog entry indicates a nonmigrated data set, AUDIT reports ***ERR 152**.
- For a dsname with leading qualifiers indicating a *backup version*:
 - If AUDIT cannot identify the first record as a CDD, it reports ***ERR 150**.
 - If the first record is a CDD, but the corresponding data set record in the BCDS has no reference to a backup version with this name, AUDIT reports ***ERR 142**.
- For a dataset with
 - A dsname *not* generated by DFSMSHsm, and
 - A CDD as a first record, and
 - A catalog entry indicating the data set is migrated, but
 - No data set record in the MCDS,AUDIT reports ***ERR 133** (see also [Diagnosing Error 133](#)).

Diagnosis and repair action for AUDIT MEDIACONTROLS for DASD

When you request AUDIT MEDIACONTROLS for DASD:

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 133, 140, 142, 149, 151, and 152.

Diagnosing Error 133: If AUDIT reports ***ERR 133** for a migrated VSAM data set, but cannot determine the name of its base cluster to recreate a data set record, AUDIT reports also condition **169**.

- For other reported conditions and troubleshooting hints. see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Errors detected when using AUDIT MEDIACONTROLS for a tape volume

For each specified volume serial number identifying a DFSMSHsm-owned backup or migration tape volume, AUDIT next verifies the type and format of tape. If the volume record indicates a tape *other than* one in single file format, AUDIT reports ***ERR 130**.

For a tape volume in single file format, AUDIT mounts the tape and reads blocks sequentially, searching for CDD control information marking the start of one or more blocks constituting a data set copy. (A given tape may contain the start or continuation of a copy spanning volumes.) When a CDD is found, AUDIT

uses the data there—and the number of blocks read since the *last* CDD found—to check other control information, and to verify or reconstruct an entry in the TTOC records for the volume.

If AUDIT cannot allocate or mount a tape volume, it reports ***ERR 131**. If AUDIT *can* allocate and mount the volume, it continues.

- If it finds no corresponding TTOC base record, AUDIT first reports ***ERR 51** (for a migration volume) or ***ERR 71** (for a backup volume).
- If AUDIT finds the TTOC base record and that record indicates that there were in-storage records that may not have been written to the OCDS, AUDIT reports ***ERR 164**.

No TTOC record exists for a migration volume

If the TTOC base is missing, AUDIT makes the following checks:

- For a CDD with a corresponding data set record, flagged as *valid*, in the MCDS:
 - If the catalog entry indicates the data set is migrated, but the data set record indicates a migration copy on some other volume or elsewhere on this volume, AUDIT reports ***ERR 149**.
 - If there is no catalog entry, or the catalog entry indicates a nonmigrated data set, AUDIT reports ***ERR 132**.
- For a CDD with a corresponding data set record, flagged as *invalid*, in the MCDS, if the catalog entry indicates the data set is migrated, AUDIT reports ***ERR 133** (see also [Diagnosing Error 133](#)).
- For a CDD with a corresponding catalog entry indicating a migrated data set, but with no data set record in the MCDS, AUDIT reports ***ERR 133** (see also [Diagnosing Error 133](#)).
- For a CDD with a corresponding data set record, flagged as *valid*, in the MCDS:
 - If the catalog entry indicates the data set is migrated, but the data set record indicates a migration copy on some other volume, but the file block id points to this position and the time and date are same as CDD time and date, AUDIT reports ***ERR 203** and patch the MCDVSN field with the volser of the tape being audited.

If no CDD record is found on the tape, AUDIT reports condition **134**.

Checking TTOC entries for a migration volume

If AUDIT finds the TTOC base record, it checks the sequence of CDDs against the sequence of TTOC entries found in the base record and zero or more TTOC extensions. If a TTOC extension is missing, AUDIT reports ***ERR 111**.

- For a CDD with a corresponding data set record in the MCDS:
 - If the block count in the *previous* TTOC entry does not agree with the number of blocks read since the *previous* CDD found, AUDIT reports ***ERR 135**.
 - If there is no catalog entry or the catalog entry indicates a nonmigrated data set, AUDIT reports ***ERR 132**.
 - If the catalog indicates the data set is migrated, but the data set record is flagged invalid, AUDIT reports ***ERR 133** (see also [Diagnosing Error 133](#)).
 - If the catalog indicates the data set is migrated, but the corresponding TTOC entry is flagged invalid, AUDIT reports ***ERR 136**.
- For a CDD with a corresponding catalog entry:
 - If the catalog entry indicates a nonmigrated data set, AUDIT reports ***ERR 132**.
 - If the catalog entry indicates a migrated data set, but there is no data set or alias record in the MCDS, AUDIT reports ***ERR 133** (see also [Diagnosing Error 133](#)).

At the end of the volume:

- If the valid block count in the TTOC does not agree with the number of valid blocks read, AUDIT reports ***ERR 137**.

- If the total block count in the TTOC does not agree with the total number of blocks read, AUDIT reports ***ERR 138**.

No TTOC record exists for a backup volume

If the TTOC base is missing, AUDIT generates for each CDD found:

- A TTOC entry flagged *valid* if the BCDS contains a data set record referring to a backup version record for the current position on the tape;
- A TTOC entry flagged *invalid* if the BCDS contains a data set record for the CDD, but none of the backup version records refers to the current position on the tape.

If no CDD record is found on the tape, AUDIT reports condition **134**.

Checking TTOC entries for a backup volume

If AUDIT finds the TTOC base record, it checks the sequence of CDDs against the sequence of TTOC entries found in the base record and zero or more TTOC extensions. If a TTOC extension is missing, AUDIT reports ***ERR 111**.

For a CDD with a corresponding data set record in the BCDS:

- If the block count in the *previous* TTOC entry does not agree with the number of blocks read since the *previous* CDD has been found, AUDIT reports ***ERR 135**.
- If the data set record refers to a backup version record for the current position on the tape, but the corresponding TTOC entry is flagged invalid, AUDIT reports ***ERR 136**.

At the end of the volume:

- If the valid block count in the TTOC does not agree with the number of valid blocks read, AUDIT reports ***ERR 137**.
- If the total block count in the TTOC does not agree with the total number of blocks read, AUDIT reports ***ERR 138**.

Diagnosis and repair action for AUDIT MEDIACONTROLS for tape

When you request AUDIT MEDIACONTROLS for a tape volume:

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 51, 71, 111, 132, 133, 134, 135, 136, 137, 138, 164, and 203.

Diagnosing Error 133: If AUDIT reports ***ERR 133** for a migrated VSAM data set, but cannot determine the name of its base cluster to recreate a data set record, AUDIT also reports ***ERR 169**.

- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Using the AUDIT VOLUMECONTROLS (BACKUP) command

When you specify the AUDIT VOLUMECONTROLS (BACKUP) parameter, AUDIT first cross-checks BCDS backup volume records for consistency with other related DFSMSHsm control records.

If there is no backup volume record for a specified volume serial number, AUDIT reports ***ERR 70**.

For an existing backup volume record (for a specified volume serial, or for each backup volume of a specified type), AUDIT makes the following checks:

- If the alternate volume serials for duplex volumes do not match between the TTOC and either the MCT or MCV, AUDIT reports ***ERR 58**.
- If the backup volume record indicates a tape volume, but there is no TTOC record, AUDIT reports ***ERR 71**.
- If the backup volume record indicates a RACF protected volume, but the TTOC record does not, or vice versa, AUDIT reports ***ERR 52**.

- If the backup volume and TTOC records indicate RACF protection, but there is no entry in RACF TAPEVOL for the volume, AUDIT reports ***ERR 53**.
- If there *is* a RACF TAPEVOL entry for the volume, but the backup volume and TTOC records do *not* indicate RACF protection, AUDIT reports ***ERR 53**.
- If there is no entry in the appropriate backup volume cycle (R) record, AUDIT reports ***ERR 77**.
- If the backup volume record indicates a tape, and the existing backup volume cycle entry indicates a DASD volume, or vice versa, AUDIT reports ***ERR 74**.
- If the backup volume record indicates a volume that's full, but the backup volume cycle entry does not, or vice versa, AUDIT reports ***ERR 75**.
- If the backup volume record indicates a single file format tape, but the backup volume cycle entry does not, or vice versa, AUDIT reports ***ERR 76**.

Checking backup volume cycle records

Next, AUDIT checks each entry in the backup volume cycle records:

- If an entry has no corresponding backup volume record, AUDIT reports ***ERR 77**.
- If the backup volume record exists, but indicates a different usage from that of the entry, AUDIT reports ***ERR 72**.

For each backup volume cycle record set:

- If the last extension of a backup volume cycle set indicates there should be another extension, or there is a gap in the extension numbers for a record set, AUDIT reports ***ERR 78**.
- If some extension, other than the last, indicates that there is no additional extension, AUDIT reports ***ERR 79**.

Checking TTOC records for backup volumes

Finally, checking is driven by TTOC records (if any) in the OCDS:

- If the AUDIT is limited to a specific BACKUPTYPE, or to specific volume serial numbers, AUDIT checks only the corresponding TTOC records.
- If the scope of the AUDIT is *not* limited to specific volume serial numbers or to a specified BACKUPTYPE, or if BACKUPTYPE(ALL) is specified, AUDIT checks *all* TTOC records for backup volumes.

For each TTOC record:

- If there is no backup volume record, AUDIT reports ***ERR 71**.
- If the backup volume record exists, but indicates a different usage from that of the TTOC record, AUDIT reports ***ERR 73**.

Diagnosis and repair action for AUDIT VOLUMECONTROLS (BACKUP)

When you request AUDIT VOLUMECONTROLS (BACKUP):

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 52, 53, 71, 72, 74, 75, 76, 77, 78, and 79.
- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Using the AUDIT VOLUMECONTROLS (MIGRATION) command

When you specify the AUDIT VOLUMECONTROLS (MIGRATION) parameter, AUDIT first cross-checks MCDS migration volume records for consistency with other related DFSMSHsm control records.

For a specified volume serial number: if there is no migration volume record or if the record indicates a level 0 volume, AUDIT reports ***ERR 50**.

For an existing migration volume record (for a specified volume serial, or for each migration volume), AUDIT makes the following checks:

- If the migration volume record indicates a tape volume, but there is no TTOC record, AUDIT reports ***ERR 51**.
- If the migration volume record indicates a RACF protected volume, but the TTOC record does not, or vice versa, AUDIT reports ***ERR 52**.
- If the migration volume and TTOC records indicate RACF protection, but there is no entry in RACF TAPEVOL for the volume, AUDIT reports ***ERR 53**.
- If there is a RACF TAPEVOL entry for the volume, but the migration volume and TTOC records do *not* indicate RACF protection, AUDIT reports ***ERR 53**.
- If the alternate volume serials for duplex volumes do not match between the TTOC and either the MCT or MCV, AUDIT reports ***ERR 58**.

Checking the level 2 control record

If the level 2 control (**S**) record exists, AUDIT checks each entry:

- If there is no migration volume record for a given entry (key range), AUDIT reports ***ERR 54**.
- If the corresponding migration volume record indicates usage *other than* a DASD level 2 volume, AUDIT reports ***ERR 55**.

Checking migration level 1 free space records

- If the last extension of the free space (**1**) record set indicates there should be another extension, AUDIT reports ***ERR 56**.
- If some extension, other than the last, indicates that there is no additional extension, AUDIT reports ***ERR 57**.

Diagnosis and repair action for AUDIT VOLUMECONTROLS (MIGRATION)

When you request AUDIT VOLUMECONTROLS (MIGRATION):

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 52, 53, 56, and 57.
- AUDIT may recommend an additional diagnostic step for error message (*ERR) 51. If you want AUDIT to continue its attempt to identify a repair action, you must specify an additional AUDIT command using the recommended parameter.
- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Using the AUDIT VOLUMECONTROLS (RECOVERABLE) command

When you specify the AUDIT VOLUMECONTROLS (RECOVERABLE) parameter, AUDIT cross-checks BCDS control records that concern recoverable (“eligible”) volumes, that is, volumes that have been backed up or dumped. These control records are checked against other DFSMSHsm controls to which they refer. AUDIT then produces a list of errors.

If a specified volume serial has not been backed up or dumped, AUDIT reports ***ERR 105**.

If an existing eligible volume (**P**) record indicates that the volume is owned by DFSMSHsm, but there is no migration volume record in the MCDS or backup volume record in the BCDS for that volume, AUDIT reports ***ERR 90**.

AUDIT detects “errors” for FRBACKUP volumes that are recoverable (***ERR 192**). When the FIX option is used and an FRBACKUP COPYPOOL volume is found to be recoverable, the only action that needs to be taken is to confirm with AUDIT COPYPOOLCONTROLS(*copypoolname*) that this volume is indeed recoverable.

Checking backup related controls

For an eligible volume record (for a specified volume serial, or for each eligible volume) indicating the volume has been *backed up* at least once, AUDIT makes the following checks for each indicated VTOC copy data set:

- If the data set is stored on a migration volume for which there is no migration volume record in the MCDS, AUDIT reports ***ERR 91**.
- If the data set is stored on a DASD backup volume for which there is no backup volume record in the BCDS, AUDIT reports ***ERR 92**.
- If the data set is stored on backup tape:
 - If there is no backup volume record for the tape volume, AUDIT reports ***ERR 92**.
 - If the data set spans tape volumes, and there is no backup volume record for the second tape, AUDIT reports ***ERR 93**.

Checking dump related controls

For an eligible volume record (for a specified volume serial, or for each eligible volume) indicating the volume has been *dumped* at least once, AUDIT makes the following checks:

- If the eligible volume refers to a dump generation for which there is no dump generation (**G**) record, AUDIT reports ***ERR 94**.
- For a dump generation record with a dump copy referring to a dump volume:
 - If there is no dump volume (**Y**) record, AUDIT reports ***ERR 95**.
 - If the dump volume record exists, but is not flagged as belonging to a dump copy, AUDIT reports ***ERR 96**.
 - If the generation referred to from a valid entry in a dump volume record is not the same as the generation referring to that dump volume, or the generation indicates a dump copy starts on a dump volume but the file sequence number is not the same as the entry number of the copy in the dump volume record, or the generation indicates a dump copy spans to a dump volume but the entry referring to the generation is not the first in the dump volume record, AUDIT reports ***ERR 97**.
 - If the dump class name in the dump volume record is not the same as the class name in the generation referring to that dump volume, AUDIT reports ***ERR 99**.
 - If the volume sequence number in the dump volume record is not the same as the sequence of that volume in the dump copy within the dump generation referring to that volume, AUDIT reports ***ERR 100**.
 - If there is no migration volume record for the volume containing the dump copy's VTOC copy data set, AUDIT reports ***ERR 101**.

If the scope of the AUDIT is *not* limited to specific volumes, the next check is driven by dump generation records and their dump device categories. For TAPE dump generation records:

- If a dump generation record refers to a volume for which there is no eligible volume record, AUDIT reports ***ERR 102**.
- If a dump generation record refers to a volume with an eligible volume record that does *not* refer back to that generation, AUDIT reports ***ERR 103**.
- If any dump copy in a dump generation record is CLOUD type, AUDIT reports ***ERR 218**.

For CLOUD dump generation records:

- If a dump generation record has any inconsistent entries as CLOUD dump, AUDIT reports ***ERR 215**.
- If the object number in a dump generation record does not match the object number of the cloud object prefix, AUDIT reports ***ERR 217**.
- If the corresponding cloud dump object can not be found, AUDIT reports ***ERR 219**.

If the scope of the AUDIT is *not* limited to specific volumes, or a generation record for a dumped volume cannot be found, or a dump volume record referred to by a generation does not refer back to that generation, the last check is driven by dump volume records flagged as valid:

- If the dump volume refers to a generation for which there is no dump generation record, AUDIT reports ***ERR 104**.
- When the dump generation record exists:
 - If the dump volume is not part of any dump copy in the generation, AUDIT reports ***ERR 98**.
 - If the dump class name in the dump volume record is not the same as the class name in the generation referring to that dump volume, AUDIT reports ***ERR 99**.
 - If the volume sequence number in the dump volume record is not the same as the sequence of that volume in the dump copy within the dump generation referring to that volume, AUDIT reports ***ERR 100**.

Diagnosis and repair action for AUDIT VOLUMECONTROLS (RECOVERABLE)

When you request AUDIT VOLUMECONTROLS (RECOVERABLE):

- AUDIT usually identifies a repair (and takes repair action when FIX is specified) for error messages (*ERR) 91 and 101.
- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Summary of errors detected by the enhanced AUDIT commands

Table 72 on page 1276 summarizes the errors detected and corrected by the audits listed in [“Using the enhanced audit command to audit DFSMSHsm control information” on page 1256](#). In this table, a D signifies that the error is detected, and a C signifies that the error is corrected when you specify the FIX parameter.

Table 72. Summary of Errors Detected by the Listed AUDIT Options											
Error Number	ABARS	CPCTL	Copy Pool	DSCTL (BACKUP)	DSCTL (MIG)	DIRCTL	MEDCTL (SDSP)	MEDCTL	VOLCTL (BACKUP)	VOLCTL (MIG)	VOLCTL (RECOV)
*ERR 03					D						
*ERR 09					D						
*ERR 17				D	D	D					
*ERR 19					D						
*ERR 20					C						
*ERR 21					C						
*ERR 22					C						
*ERR 23					D						
*ERR 24					C						
*ERR 25					C						
*ERR 26					C						
*ERR 27					C						
*ERR 28					C						
*ERR 29					D						
*ERR 30				C	C						
*ERR 35				C							
*ERR 36				C							

Table 72. Summary of Errors Detected by the Listed AUDIT Options (continued)

Error Number	ABARS	CPCTL	Copy Pool	DSCTL (BACKUP)	DSCTL (MIG)	DIRCTL	MEDCTL (SDSP)	MEDCTL	VOLCTL (BACKUP)	VOLCTL (MIG)	VOLCTL (RECOV)
*ERR 37				D							
*ERR 38				D							
*ERR 39				D							
*ERR 40				C							
*ERR 41				C							
*ERR 42				C							
*ERR 43				C		C					
*ERR 44				C							
*ERR 45				D							
*ERR 46				C							
*ERR 47				D							
*ERR 48				D				D			
*ERR 50							D	D		D	
*ERR 51								C		C	
*ERR 52									C	C	
*ERR 53									C	C	
*ERR 54										D	
*ERR 55										D	
*ERR 56										C	
*ERR 57										C	
*ERR 58						D			D	D	
*ERR 70									D		
*ERR 71								C	C		
*ERR 72									C		
*ERR 73									D		
*ERR 74									C		
*ERR 75									C		
*ERR 76									C		
*ERR 77									C		
*ERR 78									C		
*ERR 79									C		
*ERR 90											D
*ERR 91											C
*ERR 92											D
*ERR 93											D
*ERR 94											D
*ERR 95											D
*ERR 96											D
*ERR 97											D

Using the AUDIT command

Table 72. Summary of Errors Detected by the Listed AUDIT Options (continued)

Error Number	ABARS	CPCTL	Copy Pool	DSCTL (BACKUP)	DSCTL (MIG)	DIRCTL	MEDCTL (SDSP)	MEDCTL	VOLCTL (BACKUP)	VOLCTL (MIG)	VOLCTL (RECOV)
*ERR 98											D
*ERR 99											D
*ERR 100											D
*ERR 101											C
*ERR 102											D
*ERR 103											D
*ERR 104											D
*ERR 105											D
*ERR 106			D								
*ERR 110						D		D			
*ERR 111						C		C			
*ERR 112						C					
*ERR 113						C					
*ERR 114						C					
*ERR 115						C					
*ERR 116						C					
*ERR 117						D					
*ERR 118						C					
*ERR 119						D					
*ERR 120						D					
*ERR 121						D					
*ERR 122						D					
*ERR 123						D					
*ERR 124						D					
*ERR 125						C					
*ERR 130								D			
*ERR 131								D			
*ERR 132								C			
*ERR 133								C			
*ERR 134								C			
*ERR 135								C			
*ERR 136								C			
*ERR 137								C			
*ERR 138								C			
*ERR 140						D		C			
*ERR 141						C					
*ERR 142						C		C			
*ERR 143						C					
*ERR 144						C					

Table 72. Summary of Errors Detected by the Listed AUDIT Options (continued)

Error Number	ABARS	CPCTL	Copy Pool	DSCTL (BACKUP)	DSCTL (MIG)	DIRCTL	MEDCTL (SDSP)	MEDCTL	VOLCTL (BACKUP)	VOLCTL (MIG)	VOLCTL (RECOV)
*ERR 145						C					
*ERR 146						C					
*ERR 147						C	C				
*ERR 148						D	D				
*ERR 149						C	C	C			
*ERR 150								D			
*ERR 151								C			
*ERR 152								C			
*ERR 160							D				
*ERR 161							C				
*ERR 162							C				
*ERR 163							D				
*ERR 164								C			
*ERR 165	D										
*ERR 166	D										
*ERR 167	D										
*ERR 168	D										
*ERR 169								D			
*ERR 170			D								
*ERR 171			D								
*ERR 172			D								
*ERR 173			D								
*ERR 174			D								
*ERR 175			D								
*ERR 176			D								
*ERR 177			D								
*ERR 178			D								
*ERR 179			D								
*ERR 180			D								
*ERR 181			D								
*ERR 182			D								
*ERR 183			D								
*ERR 184			D								
*ERR 185			D								
*ERR 186			D								
*ERR 187			D								
*ERR 188			D								
*ERR 189			D								
*ERR 190			D								

Table 72. Summary of Errors Detected by the Listed AUDIT Options (continued)

Error Number	ABARS	CPCTL	Copy Pool	DSCTL (BACKUP)	DSCTL (MIG)	DIRCTL	MEDCTL (SDSP)	MEDCTL	VOLCTL (BACKUP)	VOLCTL (MIG)	VOLCTL (RECOV)
*ERR 191			D								
*ERR 192			D								
*ERR 193			D								
*ERR 194			D								
*ERR 195			D								
*ERR 196			D								
*ERR 197			D								
*ERR 198			D								
*ERR 199			D								
*ERR 200			D								
*ERR 201			D								
*ERR 202			C								
*ERR 203								C			
*ERR 204		D	D								
*ERR 205		D									
*ERR 206								D			
*ERR 207								C			
*ERR 208								C			
*ERR 209								D			
*ERR 210								C			
*ERR 211								C			
*ERR 215		D						D			
*ERR 216								C			
*ERR 217		D						D			
*ERR 218		D						D			
*ERR 219		D						D			
*ERR 220								D			
*ERR 221								D			
*ERR 222								D			
*ERR 230								D			
*ERR 231								D			

Using the original AUDIT command to audit DFSMSHsm data set information

This topic describes the AUDIT parameters used to detect and to report discrepancies in data set information recorded in DFSMSHsm control data sets, the VTOCs identifying data sets, and catalogs.

When the FIX parameter is specified, AUDIT can correct certain errors.

In general, the primary options listed in this topic perform less extensive diagnosis than those listed in topic One. There is some overlap between the parameters in the two topics. You may want to refer

to “Using the enhanced audit command to audit DFSMSHsm control information” on page 1256 as you choose the AUDIT parameters that will best serve your needs.

Table 73 on page 1281 lists the original AUDIT commands audit and report errors within DFSMSHsm controls and where to find more information.

<i>Table 73. Original AUDIT Commands that Audit and Report Errors</i>	
If the command you are interested in is . . .	Then see. . .
AUDIT DATASETNAMES and AUDIT LEVELS	“Using the AUDIT DATASETNAMES and AUDIT LEVELS commands” on page 1282
AUDIT VOLUMES	“Using the AUDIT VOLUMES command” on page 1284
AUDIT BACKUPVOLUMES and AUDIT BACKUPTYPE	“Using the AUDIT BACKUPVOLUMES and AUDIT BACKUPTYPE commands” on page 1286
AUDIT MASTERCATALOG and AUDIT USERCATALOG	“Using the AUDIT MASTERCATALOG and AUDIT USERCATALOG commands” on page 1288
AUDIT MIGRATIONCONTROLDATASET	“Using the AUDIT MIGRATIONCONTROLDATASET command” on page 1290
AUDIT BACKUPCONTROLDATASET	“Using the AUDIT BACKUPCONTROLDATASET command” on page 1291
AUDIT OFFLINECONTROLDATASET	“Using the AUDIT OFFLINECONTROLDATASET command” on page 1292
AUDIT ALL	Equivalent to AUDIT MCDS, then AUDIT BCDS, then AUDIT OCDS(ALL).

The parameters specified in the AUDIT command determine which data sets are audited. SMS-managed data sets are processed similarly to the audit of non-SMS-managed data sets. You can audit:

- Specific data sets
- All cataloged data sets that have the same set of initial characters of the data set name
- All data sets that reside on a specified non-SMS-managed volume
- All data sets that reside on backup volumes of a specified backup type
- All data sets cataloged in a specified catalog
- All data sets that have a data set record in the MCDS
- All data sets that have a data set record in the BCDS
- All data sets that have an entry in an OCDS TTOC record of a specified type

Audit information is written to a SYSOUT or user specified data set, in tabular form, with each row representing the result for one data set. In most cases, you can request that the audit output, in a different format, be sent to the terminal from which the AUDIT command has been issued. However, large reports, such as AUDIT ALL, cannot be sent to the terminal.

Table 74 on page 1281 explains the information that can be reported for a data set.

<i>Table 74. Headings of Output When You Audit Data Set Information</i>		
Printer Output Heading	Terminal Label	Description
ERROR TYPE	ERROR TYPE	**NONE under this heading specifies that no error has been found for the specified data set. *ERR nn specifies the error number detected for the data set.
DATA SET NAME	DSN	The entry under this heading is the name of the data set being audited.

Table 74. Headings of Output When You Audit Data Set Information (continued)		
Printer Output Heading	Terminal Label	Description
AUDITED VOLUME	AUDITED VOL	This is the serial number of the primary, migration, or backup volume being audited.
DATA SET ON VOLUME	DSN ON VOL	A serial number under this heading is the serial number of the primary, migration, or backup volume that contains the data set being audited. In addition, a volume serial number indicates that the data set is cataloged. ????? indicates that the existence of the data set on the volume indicated by the catalog could not be verified. If *ERR 16 or *ERR 17 occurs, this field can contain additional diagnostic information. See “Error codes (*ERR) and diagnosis” on page 1295.
CATALOGED TO VOLUME	CTLG VOL	A serial number under this heading is the serial number of the volume where the data set resides as recorded in the catalog. MIGRAT indicates that the data set is on a migration volume. -NONE-indicates that no record of the data set exists in the catalog. OFFLNE indicates that the catalog could not be opened.
MIGRATED TO VOLUME	MIG VOL	A serial number under this heading is the serial number of the migration volume that contains the data set as recorded in the MCDS. RECALLED indicates that the data set is on a mounted primary volume as indicated in the MCDS. DELETED indicates that the data set has been deleted as indicated in the MCDS. -NONE-specifies that no record of the data set exists in the MCDS.
BACKED UP TO VOLUME	BACK VOL	A serial number under this heading is the serial number of the backup volume that contains the most recent backup version of the data set as recorded in the BCDS. -NONE-indicates that no BCDS backup version record has been found for the data set. (This is not an error if the data set is a VTOC copy data set or a VCAT copy data set, or if the data set resides on a DASD backup volume.)
DUPLICATE DSN ON VOLUME	DUP DSN ON VOL	If the catalog indicates that a data set of the same name resides on a volume other than the one being audited and that fact is verified, the serial number of the other volume is listed under this heading.

Note: All errors listed in AUDIT reports are described in [“Error codes \(*ERR\) and diagnosis”](#) on page 1295.

Using the AUDIT DATASET NAMES and AUDIT LEVELS commands

To get an audit of specific data sets, specify the AUDIT DATASET NAMES parameter. To get an audit of all data sets having the same set of initial characters of their data set names, specify the AUDIT LEVELS parameter.

When you specify either of these options, AUDIT tries to read MCDS data set (**D**) records, if any, and catalog entries, if any, corresponding to the data sets or range of data sets you specify.

If there is no data set record in the MCDS for a specified data set:

- If there is also no catalog entry, the data set (according to its name) is a backup version, but no move (**M**) record exists for that version, AUDIT reports ***ERR 01**.
- If there is no catalog entry, and the data set is *not* (according to its name)
 - A generation data set (GDS)
 - SYSCTLG
 - SYS1.VTOCIX
 - A VTOC copy data set

- A dump VTOC copy data set

AUDIT reports ***ERR 01**.

- If a catalog entry exists, and indicates the data set is not migrated, AUDIT reports condition **04** (not necessarily an error).
- If a catalog entry exists, but indicates the data set *is* migrated, AUDIT reports ***ERR 10**.

When there *is* an MCDS data set record:

1. If the data set record indicates a migration copy on tape, but the TTOC (**T**) records for the indicated migration volume do not contain an entry for the data set, AUDIT reports ***ERR 16**.
 2. If the data set record indicates a migration copy on tape, but the positioning information in the data set record does not agree with the information in the TTOC, or the indicated TTOC extension cannot be found, AUDIT reports ***ERR 17**.
- When neither condition “1” on page 1283 nor condition “2” on page 1283 holds:
 - If there is no catalog entry, but the data set record indicates a migrated copy, AUDIT reports ***ERR 03**.
 - If both the catalog entry and the data set record indicate the data set is not migrated, but AUDIT cannot (using the OBTAIN function) verify the data set is on the volume indicated in the catalog, AUDIT reports ***ERR 06**.
 - If a catalog entry exists, indicating the data set is not migrated, but the data set record indicates the data set *is* migrated:
 - If AUDIT cannot verify the data set is on the volume indicated in the catalog, AUDIT reports ***ERR 09**.
 - If the data is on the volume indicated in the catalog, AUDIT reports ***ERR 18**.
 - If a catalog entry exists, indicating the data set is migrated, but the data set record indicates the data set is now on a primary volume, AUDIT reports ***ERR 11**.

If the specified data set is not supported by DFSMSshm (that is, not a VSAM base cluster and not a non-VSAM data set), AUDIT reports ***ERR 15**.

Information reported when using AUDIT DATASET NAMES and AUDIT LEVELS

When you specify the DATASET NAMES parameter, the audit information is reported for each specified data set, even if there is no audit error. With the LEVELS parameter, to request that audit information be reported for all the data sets audited or for only those data sets for which audit errors are detected, specify the REPORT parameter.

Figure 96 on page 1283 is a sample of the printer output when you audit data sets.

```
-- DFSMSshm AUDIT -          DATASET -- LISTING - AT 18:09:53 ON 91/01/24

ERROR          DATA SET NAME          DATA SET ON   CATALOGED     MIGRATED       BACKED UP
TYPE           TYPE                     VOLUME        TO VOLUME     TO VOLUME     TO VOLUME

**NONE         H952762.PSFB.F40AU001.DSET01          PRIM01        PRIM01        -NONE-        BATP01
*ERR 10        H952762.PSFB.F40AU001.DSET02          ??????       MIGRAT        -NONE-        BATP01
- END OF -     DATASET - LISTING -
```

Figure 96. Sample List of Output for AUDIT DATASET NAMES

Figure 97 on page 1283 is a sample of the terminal output when you audit data sets.

```
AUD=DS          DSN=H952762.PSFB.F40AU001.DSET01          ERROR TYPE= **NONE
DSN ON VOL=PRIM01  CTLG VOL=PRIM01  MIG VOL= -NONE-      BACK VOL=BATP01

AUD=DS          DSN=H952762.PSFB.F40AU001.DSET02          ERROR TYPE= *ERR 10
DSN ON VOL=?????? CTLG VOL=MIGRAT  MIG VOL= -NONE-      BACK VOL=BATP01
ARC0802I DFSMSshm AUDIT ENDING
```

Figure 97. Sample Terminal Output for AUDIT DATASET NAMES

Table 74 on page 1281 explains the headings and labels used.

Diagnosis and repair action for AUDIT DATASET NAMES and AUDIT LEVELS

When you request AUDIT DATASET NAMES or AUDIT LEVELS, AUDIT does not fix any of the detected errors. For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Using the AUDIT VOLUMES command

To request an audit of the data sets on one or more non-SMS-managed primary volumes or migration volumes, specify the AUDIT VOLUMES parameter.

For a specified volume serial, AUDIT first reads the volume record from the MCDS. If you do not include a volume serial number with the VOLUMES parameter, all the non-SMS-managed primary volumes and migration volumes that have volume records in the MCDS are audited.

In the following discussions, the master catalog and any associated user catalogs are called simply “the catalog”.

[“Errors encountered during AUDIT VOLUMES, AUDIT BACKUPVOLUMES, or AUDIT BACKUPTYPE” on page 1339](#) describes conditions which can cause AUDIT to fail its checking.

Errors detected when using AUDIT VOLUMES for a primary or migration DASD

Given a volume record indicating a DASD, the volume table of contents (VTOC) for the volume drives the audit. AUDIT handles each Format 1 DSCB in the VTOC as follows:

If the DSCB indicates any of the following, AUDIT does not check the data set further:

- SYSCTLG
- SYS1.VTOCIX
- SYS1.VVDS
- Model DSCB
- Multivolume data set

If the DSCB indicates a utility data set, AUDIT does not check it further. If FIX is specified, AUDIT scratches and uncatalogs the data set, overriding any expiration date, and issuing a message to identify the data set.

For each remaining DSCB, AUDIT tries to read a corresponding data set record from the MCDS.

When there is no data set record for the data set:

- If the data set (according to its name) is a backup version, but no move record exists for that version, AUDIT reports ***ERR 01**.
- If there is no catalog entry, AUDIT reports condition **01**.
- If a catalog entry exists, but indicates the data set is on a primary volume *other than* the volume being audited, AUDIT reports ***ERR 05**.
- If a catalog entry exists, but indicates the data set is migrated, AUDIT reports ***ERR 10**.
- If the data set is not supported by DFSMSHsm (that is, not a VSAM base cluster and not a non-VSAM data set), AUDIT reports condition **15**.

When there is an MCDS data set record for the data set:

- If the data set record indicates the data set has been migrated, then recalled:
 - If the recall volume is the same as the primary volume being audited, but there is (now) no catalog entry, AUDIT reports ***ERR 02**.
 - If a catalog entry exists, but indicates the data set is on a primary volume different from the recall volume, AUDIT reports ***ERR 07**.
 - If a catalog entry exists, but indicates the data set is on a primary volume different from the one being audited, AUDIT reports ***ERR 08**.

- If a catalog entry exists, but indicates the data set is now migrated, AUDIT reports ***ERR 11**.
- If the data set record indicates the data set is now migrated:
 1. If the data set record indicates a migration copy on tape, but the TTOC records for the indicated migration volume do not contain an entry for the data set, AUDIT reports ***ERR 16**.
 2. If the data set record indicates a migration copy on tape, but the positioning information in the data set record does not agree with the information in the TTOC, or the indicated TTOC extension cannot be found, AUDIT reports ***ERR 17**.

When neither condition “1” on page 1285 nor condition “2” on page 1285 holds:

- If there is (now) no catalog entry, AUDIT reports ***ERR 03**.
- If a catalog entry exists, indicating the data set is on the volume being audited, AUDIT reports ***ERR 18**.

When you specify the AUDIT VOLUMES command to audit a migration level 1 volume, be aware of the following:

- Data sets in a small-data-set-packing data set are not audited.
- Data sets that have been backed up with the BACKDS command and temporarily reside on migration level 1 volumes are audited.

Errors detected when using AUDIT VOLUMES for a tape migration volume

Given a volume record indicating a tape migration volume, the tape table of contents (TTOC) records for the volume drive the audit. For each valid data set entry in the TTOC, AUDIT makes the following checks:

When there is no data set record in the MCDS for the data set:

- If there is (now) no catalog entry, AUDIT reports condition **01**.
- If a catalog entry exists, but indicates the data set *is* migrated, AUDIT reports ***ERR 10**.

When there *is* a data set record in the MCDS, indicating the data set is now migrated:

- If the data set record indicates a migration copy on tape, but the positioning information in the data set record does not agree with the information in the TTOC, AUDIT reports ***ERR 17**.
- If there is (now) no catalog entry, AUDIT reports ***ERR 03**.
- If a catalog entry exists, indicating the data set is on a primary volume, AUDIT reports ***ERR 18**.

Information reported when using AUDIT VOLUMES

Figure 98 on page 1285 is a sample of the printer output when you audit primary or migration volume information.

```
-- DFSMSshm AUDIT -          VOLUME          -- LISTING - AT 10:21:20 ON 91/02/02

AUDITING PRIMARY VOLUME, VOLSER = PRIM02

ERROR          DATA SET NAME          AUDITED          CATALOGED          MIGRATED          DUPLICATE DSN
TYPE                                VOLUME                                TO VOLUME          TO VOLUME          ON VOLUME

*ERR 02          H952762.PSFB.F40AU001.DSET03          PRIM02          -NONE-          RECALLED
- END OF -          VOLUME - LISTING -
```

Figure 98. Sample List of Output for AUDIT VOLUMES

Figure 99 on page 1285 is a sample of the terminal output when you audit a primary or migration volume.

```
ARC0801I DFSMSshm AUDIT STARTING

AUDITING VOLUMES, VOLSER = PRIM02

AUD=VL   DSN=H952762.PSFB.F40AU001.DSET03   ERROR TYPE= *ERR 02
AUDITED  VOL=PRIM02   CTLG VOL= -NONE-   MIG VOL=RECALLED DUP DSN ON VOL=
```

Figure 99. Sample Terminal Output for AUDIT VOLUMES

Table 74 on page 1281 explains the headings and labels used.

Diagnosis and repair action for AUDIT VOLUMES

When you request AUDIT VOLUMES:

- AUDIT attempts repair action when FIX is specified, for error messages (*ERR) 01 and 02 for a primary volume, and 03 for a migration volume. In each case, AUDIT issues a message that describes the type of correction attempted and whether or not the fix has been successful. When a fix is unsuccessful, the message contains the return code from the routine attempting the fix.
- For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Using the AUDIT BACKUPVOLUMES and AUDIT BACKUPTYPE commands

When you specify AUDIT BACKUPVOLUMES to request an audit of the data sets with copies on one or more backup volumes, AUDIT first tries to read from the BCDS the backup volume record for each specified volume, or for every backup volume. The device type in each *volume record* determines further checking.

When you specify AUDIT BACKUPTYPE, AUDIT first tries to read from the BCDS the backup cycle volume (BVR) records for

- A specified day, or
- All days in the backup cycle, or
- All spill volumes, or
- All daily and spill backup volumes.

For each entry in a backup cycle volume record, AUDIT then tries to read the corresponding backup volume record. The device type in each *entry* determines further checking.

[“Errors encountered during AUDIT VOLUMES, AUDIT BACKUPVOLUMES, or AUDIT BACKUPTYPE” on page 1339](#) describes conditions which can cause AUDIT to fail its checking.

Error detected when using AUDIT BACKUPVOLUMES or AUDIT BACKUPTYPE for a backup DASD

Given an indication of a backup DASD, the volume table of contents (VTOC) for the volume drives the audit. AUDIT handles each Format 1 DSCB in the VTOC as follows:

If the DSCB indicates a utility data set, AUDIT does not check it further. If FIX is specified, AUDIT scratches and uncatalogs the data set, overriding any expiration date, and issuing a message to identify the data set. (The utility data set names detected on primary and migration volumes should not occur on a backup volume.)

If the DSCB indicates any of the following, AUDIT does not check the data set further:

- A VSAM data set
- An unmovable data set

For each remaining DSCB, AUDIT tries to read a corresponding backup version record from the BCDS. If such a record exists, but the volume indicated in that record is not the same as the volume being audited, AUDIT reports ***ERR 14**.

Error detected when using AUDIT BACKUPVOLUMES or AUDIT BACKUPTYPE for a tape backup volume

Given an indication of a tape backup volume, AUDIT uses the backup category in the backup volume record to read the tape table of contents (TTOC) records for the volume. Those TTOC records drive the audit for that volume.

For each valid data set entry in the TTOC, AUDIT tries to read a backup version record from the BCDS. If such a record exists, but none of the volumes (including any spanned volumes) indicated in that record is the same as the volume being audited, AUDIT reports ***ERR 14**.

Information reported when using AUDIT BACKUPVOLUMES and AUDIT BACKUPTYPE

When you specify AUDIT BACKUPVOLUMES or AUDIT BACKUPTYPE, the order of the output for the different requests is as follows:

- AUDIT BACKUPVOLUMES

The output is ordered according to the sequence of the volume serial numbers in the BCDS.

- AUDIT BACKUPVOLUMES(*volser...*)

The order is the order you specify.

- AUDIT BACKUPTYPE(SPILL)

The order is that of the entries in the spill backup cycle volume record.

- AUDIT BACKUPTYPE(DAILY(*day*))

The order is that of the entries in the daily BVR for the specified day.

- AUDIT BACKUPTYPE(DAILY)

The order is for all volumes for day 1, then all for day 2, and so forth, until all daily backup volumes have been audited. In addition, for each day, the order of the output is that of the entries in the BVR for that particular day.

- AUDIT BACKUPTYPE(ALL)

The order is all daily backup volumes, then all spill backup volumes.

[Figure 100 on page 1287](#) is a sample list of printer output you can request when you audit a backup volume.

```
-- DFSMSHsm AUDIT -                BACKUP VOLUME -- LISTING - AT 10:13:08 ON 91/02/02
AUDITING BACKUP TAPE VOLUME, VOLSER = BATP01

ERROR          DATA SET NAME          AUDITED      BACKED UP
TYPE                                     VOLUME      TO VOLUME

**NONE         HSM40.VTOC.VPRIM01.D84033.T100500    BATP01      -NONE-
**NONE         HSM40.BACK.G834921.RRDS.H4033.T100644  BATP01      BATP01
**NONE         HSM40.BACK.G834921.RRDS.H4033.T100720  BATP01      BATP01
*ERR 14        HSM40.BACK.H952762.PSF.H4033.T100746    BATP01      -NONE-
**NONE         HSM40.BACK.H952762.PSF.H4033.T100802    BATP01      BATP01
**NONE         HSM40.BACK.H952762.PSF.H4033.T100817    BATP01      BATP01
*ERR 14        HSM40.BACK.H952762.PSF.H4033.T100833    BATP01      -NONE-
**NONE         HSM40.BACK.H952762.PSF.H4033.T100847    BATP01      BATP01
**NONE         HSM40.BACK.H952762.PSF.H4033.T100903    BATP01      BATP01
**NONE         HSM40.BACK.H952762.PSF.H4033.T100916    BATP01      BATP01
- END OF -      BACKUP VOLUME - LISTING -
```

Figure 100. Sample List of Output for AUDIT BACKUPVOLUMES

[Figure 101 on page 1288](#) is a sample terminal list you can request when you audit a backup volume.


```

AUDITING VOLUMES, VOLSER = BATP01
AUD=TV   DSN=HSM40.VTOC.VPRIM01.D84033.T100500   ERROR TYPE  **NONE
AUDITED VOL BATP01      BACKUP VOL -NONE-
AUD=TV   DSN=HSM40.BACK.G834921.RRDS.H4033.T100644   ERROR TYPE  **NONE
AUDITED VOL BATP01      BACKUP VOL BATP01
AUD=TV   DSN=HSM40.BACK.G834921.RRDS.H4033.T100720   ERROR TYPE  **NONE
AUDITED VOL BATP01      BACKUP VOL BATP01
AUD=TV   DSN=HSM40.BACK.H952762.PSF.H4033.T100746   ERROR TYPE  *ERR 14
AUDITED VOL BATP01      BACKUP VOL -NONE-
AUD=TV   DSN=HSM40.BACK.H952762.PSF.H4033.T100802   ERROR TYPE  **NONE
AUDITED VOL BATP01      BACKUP VOL BATP01
AUD=TV   DSN=HSM40.BACK.H952762.PSF.H4033.T100817   ERROR TYPE  **NONE
AUDITED VOL BATP01      BACKUP VOL BATP01
AUD=TV   DSN=HSM40.BACK.H952762.PSF.H4033.T100833   ERROR TYPE  *ERR 14
AUDITED VOL BATP01      BACKUP VOL -NONE-
AUD=TV   DSN=HSM40.BACK.H952762.PSF.H4033.T100847   ERROR TYPE  **NONE
AUDITED VOL BATP01      BACKUP VOL BATP01
AUD=TV   DSN=HSM40.BACK.H952762.PSF.H4033.T100903   ERROR TYPE  **NONE
AUDITED VOL BATP01      BACKUP VOL BATP01
AUD=TV   DSN=HSM40.BACK.H952762.PSF.H4033.T100916   ERROR TYPE  **NONE
AUDITED VOL BATP01      BACKUP VOL BATP01
ARC0802I DFSMSHsm  AUDIT  ENDING

```

Figure 101. Sample Terminal List for AUDIT BACKUPVOLUMES

Table 74 on page 1281 explains the headings and labels used.

Diagnosis and repair action for AUDIT BACKUPVOLUMES and AUDIT BACKUPTYPE

When you request AUDIT BACKUPVOLUMES or BACKUPTYPE, AUDIT attempts repair action when FIX is specified, for error message (*ERR) 14 for a tape backup volume. In each case, AUDIT issues a message that describes the type of correction attempted and whether or not the fix has been successful. When a fix is unsuccessful, the message contains the return code from the routine attempting the fix. (See [“Error codes \(*ERR\) and diagnosis”](#) on page 1295.)

Using the AUDIT MASTERCATALOG and AUDIT USERCATALOG commands

You can request the following catalog audits with the AUDIT command:

- An audit of the master catalog by specifying the MASTERCATALOG parameter
- An audit of a user catalog by specifying the USERCATALOG parameter

When you specify the FIX parameter, the AUDIT function enqueues continually on the DFSMSHsm control data sets during the entire audit, to minimize the possibility of other DFSMSHsm functions making changes to them. Because auditing a complete catalog can take considerable time, AUDIT issues the following message:

```

ARC0803A  WARNING:  AUDIT OF CATALOG MAY DEGRADE PERFORMANCE,
REPLY 'Y' TO START AUDIT OR 'N' TO CANCEL AUDIT COMMAND.

```

If the system operator does not reply with a Y, a message is sent to the system console stating that the operator canceled the audit of the catalog.

Assuming the audit is not canceled, the data sets driving the audit are those that are supported for migration and have a cluster (VSAM) or alien (non-VSAM) entry in the specified catalog. For each such entry, AUDIT makes the following checks:

If the catalog entry indicates a data set on a primary volume of a device type managed by DFSMSHsm:

- If AUDIT cannot verify (using the OBTAIN function) that the non-VSAM data set is indeed on the indicated volume (the volume may not be mounted):
 - If there is no MCDS data set record, AUDIT reports ***ERR 04**.
 - If there is an MCDS data set record indicating the data set has been recalled, AUDIT reports ***ERR 06**.
 - If there is an MCDS data set record, indicating the data set is migrated:
 1. If the data set record indicates a migration copy on tape, but the TTOC records for the indicated migration volume do not contain an entry for the data set, AUDIT reports ***ERR 16**.

2. If the data set record indicates a migration copy on tape, but the positioning information in the data set record does not agree with the information in the TTOC, or the indicated TTOC extension cannot be found, AUDIT reports ***ERR 17**.

When neither condition “1” on page 1288 nor condition “2” on page 1289 holds, AUDIT reports ***ERR 09**.

- If the data set is on the indicated volume, but the MCDS data set record indicates the data set is migrated, AUDIT reports ***ERR 18**.

If the catalog entry indicates a migrated data set:

- If there is no MCDS data set record, AUDIT reports ***ERR 10**.
- If there is an MCDS data set record indicating the data set has been recalled, AUDIT reports ***ERR 11**.

Information reported when using AUDIT MASTERCATALOG and AUDIT USERCATALOG

Figure 102 on page 1289 is a sample list of printer output you can request when you audit master catalog information.

```
-- DFSMSHsm AUDIT - CATALOG -- LISTING - AT 18:37:05 ON 91/01/24
```

ERROR TYPE	DATA SET NAME	DATA SET ON VOLUME	CATALOGED TO VOLUME	MIGRATED TO VOLUME	BACKED UP TO VOLUME
*ERR 04	TTL.IMS.STAGE2	??????	P3282B	-NONE-	
*ERR 04	TTL.ISPF.PROFILE	??????	P3282B	-NONE-	
**NONE	TTL.JCL.CNTL	P3282B	P3282B	-NONE-	
*ERR 04	TTL.SPFEDITA	??????	P3282B	-NONE-	
*ERR 04	TTL.SPFEDITB	??????	P3282B	-NONE-	
**NONE	TTL.SPFLOG1.LIST	P3282B	P3282B	-NONE-	
**NONE	TTL.SPFTEMP1.CNTL	P3282B	P3282B	-NONE-	
**NONE	VTAM.CNTL	P3282B	P3282B	-NONE-	

```
- END OF - CATALOG - LISTING -
```

Figure 102. Sample List of Output for AUDIT MASTERCATALOG

Figure 103 on page 1289 is a sample list of printer output you can request when you audit user catalog information.

```
-- DFSMSHsm AUDIT - CATALOG -- LISTING - AT 10:13:51 ON 91/02/02
```

AUDITING NEWSAM.CAT

ERROR TYPE	DATA SET NAME	DATA SET ON VOLUME	CATALOGED TO VOLUME	MIGRATED TO VOLUME	BACKED UP TO VOLUME
*ERR 10	D324711.POFB.N.F40CM012.DSET01	??????	MIGRAT	-NONE-	
*ERR 04	D324711.POFB.N.F40CM013.DSET01	??????	PRIM01	-NONE-	
*ERR 04	D324711.POFB.N.F40CM013.DSET02	??????	PRIM01	-NONE-	
*ERR 10	D324711.PSF.N.F40CM012.VSPRIMER	??????	MIGRAT	-NONE-	
*ERR 04	D324711.PSF.N.VSPRIMER	??????	PRIM01	-NONE-	
*ERR 10	D324711.PSFB.N.F40CM012.DSET02	??????	MIGRAT	-NONE-	
*ERR 04	D324711.PSFB.N.F40CM013.DSET01	??????	PRIM01	-NONE-	
*ERR 04	D324711.PSFB.N.F40CM013.DSET02	??????	PRIM01	-NONE-	
*ERR 04	D324711.PSFB.N.F40CM029.DSET04	??????	PRIM01	-NONE-	
*ERR 04	D324711.PSFB.N.F40CM029.DSET08	??????	PRIM02	-NONE-	
*ERR 04	D324711.PSFB.N.F40CM029.DSET09	??????	PRIM02	-NONE-	
*ERR 04	D324711.PSFB.N.F40CM013.DSET01	??????	PRIM01	-NONE-	
*ERR 04	D324711.PSFB.N.F40HM007.FILL.UP	??????	MIG101	-NONE-	
*ERR 04	D324711.PSFB.N.TEST	??????	PRIM01	-NONE-	
*ERR 04	D324711.PSFB.N.TEST.DSET01	??????	PRIM01	-NONE-	
*ERR 04	D324711.PSFB.R.F40FP058.DSET02	??????	PRIM01	-NONE-	
*ERR 04	D324711.PSFB.RP.F40RP023.DSET06	??????	PRIM02	-NONE-	
*ERR 04	D324711.PSFB.RP.F40RP045.DSET04	??????	PRIM01	-NONE-	
**NONE	G834921.ISPF.PROFILE	P3282B	P3282B	-NONE-	
**NONE	G834921.RRDS.N.F40EX001.CLUSTER3	PRIM01	PRIM01	-NONE-	
**NONE	G834921.RRDS.N.F40EX001.CLUSTER4	PRIM01	PRIM01	-NONE-	

```
- END OF - CATALOG - LISTING
```

Figure 103. Sample List of Output for AUDIT USERCATALOG

Table 74 on page 1281 explains the headings and labels used.

Diagnosis and repair action for AUDIT MASTERCATALOG and AUDIT USERCATALOG

When you request AUDIT MASTERCATALOG or AUDIT USERCATALOG, AUDIT does not fix any of the detected errors. You may, however, be able to use the FIXCDS command to add or change a record of a data set in the MCDS, to make the catalog and the MCDS agree. To find out how to use the FIXCDS command, refer to the [z/OS DFSMSHsm Diagnosis](#).

For any reported condition and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis”](#) on page 1295.

Using the AUDIT MIGRATIONCONTROLDATASET command

When you specify `AUDIT MIGRATIONCONTROLDATASET`, the data set records in the MCDS drive the audit. For each such record (data set), `AUDIT` makes the following checks:

1. If the data set record indicates a migration copy on tape, but the TTOC records for the indicated migration volume do not contain an entry for the data set, AUDIT reports ***ERR 16**.
2. If the data set record indicates a migration copy on tape, but the positioning information in the data set record does not agree with the information in the TTOC, or the indicated TTOC extension cannot be found, AUDIT reports ***ERR 17**.

When neither condition “1” on page 1290 nor condition “2” on page 1290 holds:

- If there is no catalog entry, but the data set record indicates a migrated copy, AUDIT reports ***ERR 03**.
- If both the catalog entry and the data set record indicate the data set is not migrated, but AUDIT cannot (using the OBTAIN function) verify the data set is on the volume indicated in the catalog, AUDIT reports ***ERR 06**.
- If a catalog entry exists, indicating the data set is not migrated, but the data set record indicates the data set *is* migrated:
 - If AUDIT cannot verify the data set is on the volume indicated in the catalog, AUDIT reports ***ERR 09**.
 - If the data set is on the volume indicated in the catalog, AUDIT reports ***ERR 18**.
- If a catalog entry exists, indicating the data set is migrated, but the data set record indicates the data set is now on a primary volume, AUDIT reports ***ERR 11**.

Information reported when using AUDIT MIGRATIONCONTROLDATASET

Figure 104 on page 1290 is a sample list of printer output that you can request when you audit the MCDS.

[illegible]

Figure 104. Sample List of Output for AUDIT MCDS

Figure 105 on page 1291 is a sample terminal list that you can request when you audit the MCDS.


```

AUD=MC      DSN=D324711.ESDS.R.F40RP123.CLUSTER2      ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP02      BACK VOL=

AUD=MC      DSN=D324711.PSF.N.F40RL016.DSET01          ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP02      BACK VOL=

AUD=MC      DSN=D324711.PSF.N.F40RL055.DSET02          ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP02      BACK VOL=

AUD=MC      DSN=D324711.PSF.R.F40RP123.DSET02          ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP02      BACK VOL=

AUD=MC      DSN=G834921.KSDS.N.F40TM479.CLUSTER1       ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP01      BACK VOL=

AUD=MC      DSN=G834921.PSF.N.F40TM479.DSET03          ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP03      BACK VOL=

AUD=MC      DSN=H952762.PSFB.F40AU001.DSET03          ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP03      BACK VOL=

AUD=MC      DSN=H952762.PSFB.F40AU001.DSET04          ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP01      BACK VOL=

AUD=MC      DSN=M059259.BDAM.N.F40RL025.DSET01         ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP02      BACK VOL=

AUD=MC      DSN=M059259.PSF.N.F40RL050.DSET02         ERROR TYPE= *ERR 06
DSN ON VOL=??????? CTLG VOL=PRIM01 MIG VOL=RECALLED     BACK VOL=

AUD=MC      DSN=M059259.PSF.N.F40RL056.DSET12         ERROR TYPE= *ERR 03
DSN ON VOL=??????? CTLG VOL=-NONE- MIG VOL=M2TP03      BACK VOL=

AUD=MC      DSN=M100222.GDG.N.F40RP123.DSET03.G0002V00 ERROR TYPE= *ERR 06
DSN ON VOL=??????? CTLG VOL=PRIM05 MIG VOL=RECALLED     BACK VOL=

AUD=MC      DSN=M100222.KSDS.R.F40RP123.CLUSTER1     ERROR TYPE= *ERR 06
DSN ON VOL=??????? CTLG VOL=PRIM05 MIG VOL=RECALLED     BACK VOL=

AUD=MC      DSN=M100222.PSFB.N.F40RP124.DSET01        ERROR TYPE= *ERR 06
DSN ON VOL=??????? CTLG VOL=PRIM03 MIG VOL=RECALLED     BACK VOL=

```

Figure 105. Sample Terminal List for AUDIT MCDS

Table 74 on page 1281 explains the headings and labels used.

Diagnosis and repair action for AUDIT MIGRATIONCONTROLDATASET

When you request AUDIT MIGRATIONCONTROLDATASET, AUDIT attempts repair action when FIX is specified, for error message (*ERR) 03. In each instance, AUDIT issues a message that describes the type of correction attempted and whether or not the fix has been successful. When a fix is unsuccessful, the message contains the return code from the routine attempting the fix.

For other reported conditions and troubleshooting hints, see “Error codes (*ERR) and diagnosis” on page 1295.

Using the AUDIT BACKUPCONTROLDATASET command

When you specify AUDIT BACKUPCONTROLDATASET, the data set records in the BCDS drive the audit. A backed up data set is audited if either:

- At least one backup version was made when the data set was cataloged, or
- An MCDS data set record exists (the data set was migrated at some time).

For each such record (data set), AUDIT makes the following checks:

1. If the MCDS data set record indicates a migration copy on tape, but the TTOC records for the indicated migration volume do not contain an entry for the data set, AUDIT reports ***ERR 16**.
2. If the MCDS data set record indicates a migration copy on tape, but the positioning information in the data set record does not agree with the information in the TTOC, or the indicated TTOC extension cannot be found, AUDIT reports ***ERR 17**.

When neither condition “1” on page 1291 nor condition “2” on page 1291 holds:

- If there is no catalog entry, but the data set record indicates a migrated copy, AUDIT reports ***ERR 03**.
- If a catalog entry exists, indicating a data set on a primary volume:
 - If AUDIT cannot (using the OBTAIN function) verify the data set is on the volume indicated in the catalog:
 - If there is no MCDS data set record, AUDIT reports ***ERR 04**.
 - If the MCDS data set record indicates the data set was migrated, then recalled, AUDIT reports ***ERR 06**.

- If the MCDS data set record indicates the data set is now migrated, AUDIT reports ***ERR 09**.
- If the data set is on the indicated volume, but the MCDS data set record indicates the data set is migrated, AUDIT reports ***ERR 18**.
- If a catalog entry exists, indicating a migrated data set:
 - If there is no MCDS data set record, AUDIT reports ***ERR 10**.
 - If the MCDS data set record indicates the data set was migrated, then recalled, AUDIT reports ***ERR 11**.

Information reported when using AUDIT BACKUPCONTROLDATASET

Figure 106 on page 1292 is a sample list of printer output that you can request when you audit the BCDS.

```
-- DFSMSHsm AUDIT - DFSMSHsm CONTROL DATASET -- LISTING - AT 18:52:05 ON 91/01/24
```

ERROR TYPE	DATA SET NAME	DATA SET ON VOLUME	CATALOGED TO VOLUME	MIGRATED TO VOLUME	BACKED UP TO VOLUME
**NONE	H952762.PSFB.F40AU001.DSET01	PRIM01	PRIM01	-NONE-	BATP01
*ERR 06	H952762.PSFB.F40AU001.DSET03	??????	PRIM04	RECALLED	BATP01
**NONE	M059259.PSF.N.F40RL035.DSET01	??????	-NONE-	RECALLED	BACK01
*ERR 03	H952762.PSFB.F40AU001.DSET04	??????	-NONE-	M2TP01	BATP02
**NONE	M059259.PSF.N.F40RL050.DSET03	??????	-NONE-	RECALLED	BACK01
**NONE	M100222.KSDS.R.F40RP125.DSET01	PRIM01	PRIM01	-NONE-	BACK01
- END OF -	CATALOG - LISTING -				

Figure 106. Sample List of Output for AUDIT BCDS

Figure 107 on page 1292 is a sample terminal list that you can request when you audit the BCDS.

AUD=BC	DSN=H952762.PSFB.F40AU001.DSET01	ERROR TYPE= **NONE
DSN ON VOL=PRIM01	CTLG VOL=PRIM01 MIG VOL= -NONE-	BACK VOL=BATP01
AUD=BC	DSN=H952762.PSFB.F40AU001.DSET03	ERROR TYPE= *ERR 06
DSN ON VOL=??????	CTLG VOL=PRIM04 MIG VOL=RECALLED	BACK VOL=BATP01
AUD=BC	DSN=H952762.PSFB.F40AU001.DSET04	ERROR TYPE= *ERR 03
DSN ON VOL=??????	CTLG VOL= -NONE- MIG VOL=M2TP01	BACK VOL=BATP02
AUD=BC	DSN=M059259.PSF.N.F40RL035.DSET01	ERROR TYPE= **NONE
DSN ON VOL=??????	CTLG VOL= -NONE- MIG VOL=RECALLED	BACK VOL=BACK01
AUD=BC	DSN=M059259.PSF.N.F40RL050.DSET03	ERROR TYPE= **NONE
DSN ON VOL=??????	CTLG VOL= -NONE- MIG VOL=RECALLED	BACK VOL=BACK01
AUD=BC	DSN=M100222.KSDS.R.F40RP125.DSET01	ERROR TYPE= **NONE
DSN ON VOL=PRIM01	CTLG VOL=PRIM01 MIG VOL= -NONE-	BACK VOL=BACK01
ARC0802I	DFSMSHsm AUDIT ENDING	

Figure 107. Sample Terminal List for AUDIT BCDS

Table 74 on page 1281 explains the headings and labels used.

Diagnosis and repair action for AUDIT BACKUPCONTROLDATASET

When you request AUDIT BACKUPCONTROLDATASET, AUDIT attempts repair action when FIX is specified, for error message (*ERR) 03. In each instance, AUDIT issues a message that describes the type of correction attempted and whether or not the fix has been successful. When a fix is unsuccessful, the message contains the return code from the routine attempting the fix.

For other reported conditions and troubleshooting hints, see [“Error codes \(*ERR\) and diagnosis” on page 1295](#).

Using the AUDIT OFFLINECONTROLDATASET command

When you specify AUDIT OFFLINECONTROLDATASET, you must also specify which tape table of contents (TTOC) records you wish to drive the audit. You can choose records for:

- A specific day in the backup cycle (DAILY(*day*)), or
- All days in the backup cycle (DAILY), or
- All spill volumes (SPILL), or
- All level 2 migration tape volumes (ML2), or
- All backup and migration tape volumes (ALL).

Errors detected when using AUDIT OFFLINECONTROLDATASET for backup

If you request any category of backup volume, AUDIT reads from the OCDS the TTOC records for all backup tapes in the categories specified.

For each valid data set entry in each TTOC, AUDIT tries to read a backup version record from the BCDS. If a backup version record exists, but none of the volumes (including any spanned volumes) indicated in that record is the same as the volume whose TTOC is driving the audit, AUDIT reports ***ERR 14**.

Errors detected when using AUDIT OFFLINECONTROLDATASET for migration

If you request the ML2 category, AUDIT reads from the OCDS the TTOC records for all migration level 2 tape volumes.

For each valid data set entry in each TTOC, AUDIT uses the data set name in that entry to find a corresponding data set record from the MCDS:

- If that name was apparently generated by DFSMSHsm for a migration copy (the second qualifier is .HMIG.), AUDIT tries to read the corresponding alias (**A**) record from the MCDS. If this record can be read, AUDIT uses it to try to read the data set record.
- If the name in the TTOC entry is not recognized as generated by DFSMSHsm, or if no alias record can be found, AUDIT uses the name in the TTOC entry to read the data set record.

For each data set, AUDIT then makes the following checks:

- If there is no catalog entry, but the data set record indicates a migrated copy, AUDIT reports ***ERR 03**.
- If a catalog entry exists, indicating the data set is on a primary volume:
 - If there is no MCDS data set record, AUDIT reports ***ERR 04**.
 - If there is an MCDS data set record, indicating the data set has been recalled, but to a volume different from the one recorded in the catalog, AUDIT reports ***ERR 08**.
 - If the data set record indicates a migration copy on tape, but the positioning information in the data set record does not agree with the information in the TTOC, or the indicated TTOC extension cannot be found, AUDIT reports ***ERR 17**.
 - If the data set record indicates the data set is now migrated to a DASD, AUDIT reports ***ERR 18**.
- If the catalog entry indicates a migrated data set:
 - If there is no MCDS data set record, AUDIT reports ***ERR 10**.
 - If there is an MCDS data set record, indicating the data set has been recalled, AUDIT reports ***ERR 11**.

Information reported when using AUDIT OFFLINECONTROLDATASET

The order of output for the different requests for auditing the OCDS is as follows:

- AUDIT OFFLINECONTROLDATASET(DAILY(*day*)):

The output is in ascending volume serial number order for tape daily backup volumes assigned to the day you specify.

- AUDIT OFFLINECONTROLDATASET(DAILY):

The output is ordered by all volumes assigned to day 1, then all volumes assigned to day 2, and so forth, until DFSMSHsm has audited all tape daily backup volumes in the OCDS.

- AUDIT OFFLINECONTROLDATASET(SPILL):

The output is in ascending volume serial number order for tape spill volumes in the OCDS.

- AUDIT OFFLINECONTROLDATASET(ML2):

The output is in ascending volume serial number order for tape migration level 2 volumes in the OCDS.

- AUDIT OFFLINECONTROLDATASET(ALL):

Using the AUDIT command

1. All tape daily backup volumes. These are ordered by day, with the volumes for each day in ascending volume serial number order.
2. All tape spill backup volumes. These are in ascending volume serial number order.
3. Unassigned tape backup volumes. These are in ascending volume serial number order.
4. Migration level 2 volumes. These are in ascending volume serial number order.

Figure 108 on page 1294 is a sample list of printer output you can request when you audit OCDS information for all tape volumes.

```
-- DFSMSHsm AUDIT -          DFSMSHsm CONTROL DATASET -- LISTING - AT 18:36:38 ON 91/01/24

AUDITING THE OFFLINE CONTROL DATASET

  ERROR          DATA SET NAME          AUDITED          BACKED UP
  TYPE                                     VOLUME           TO VOLUME

**NONE          HSM40.VT0C.VPRIM01.D84024.T180240      BATP01          -NONE-
**NONE          HSM40.BACK.H952762.PSFB.H4024.T180339    BATP01          BATP01
**NONE          HSM40.BACK.H952762.PSFB.H4024.T180412    BATP01          BATP01
**NONE          HSM40.BACK.H952762.PSFB.H4024.T180432    BATP01          BATP01
*ERR 14         HSM40.BACK.H952762.PSFB.H4024.T180458    BATP01          BATP02
AUDITING THE OFFLINE CONTROL DATASET, MIGRATION VOLUME = M2TP01

  ERROR          DATA SET NAME          AUDITED          CATALOGED          MIGRATED          DUPLICATE DSN
  TYPE                                     VOLUME           TO VOLUME         TO VOLUME         ON VOLUME

*ERR 03         H952762.PSFB.F40AU001.DSET04          M2TP01          -NONE-            M2TP01
- END OF -      CATALOG - LISTING -
```

Figure 108. Sample List of Output for AUDIT OCDS(ALL)

Figure 109 on page 1294 is a sample terminal list you can request when you audit OCDS information for all tape migration level 2 volumes.

```
AUDITING OCDS, MIGRATION VOL=M2TP01

AUD=L2          DSN=H952762.PSFB.F40AU001.DSET04          ERROR TYPE= *ERR 03
AUDITED VOL=M2TP01  CTLG VOL=-NONE-  MIG VOL= M2TP01  DUP DSN ON VOL=
ARC0802I DFSMSHsm AUDIT ENDING
```

Figure 109. Sample Terminal List for AUDIT OCDS(ML2)

Table 74 on page 1281 explains the headings and labels used.

Diagnosis and repair action for AUDIT OFFLINECONTROLDATASET

When you request AUDIT OFFLINECONTROLDATASET, AUDIT attempts repair action when FIX is specified, for error messages (*ERR) 03 and 14. In each instance, AUDIT issues a message that describes the type of correction attempted and whether or not the fix has been successful. When a fix is unsuccessful, the message contains the return code from the routine attempting the fix.

For other reported conditions and troubleshooting hints, see “Error codes (*ERR) and diagnosis” on page 1295.

Summary of errors detected by the original AUDIT commands

Table 75 on page 1294 summarizes the errors detected and corrected by the audits listed in “Using the original AUDIT command to audit DFSMSHsm data set information” on page 1280. In this table, a D signifies that the error is detected, and a C signifies that the error is corrected when you specify the FIX parameter.

Table 75. Summary of errors detected or corrected by the listed AUDIT options									
Error Number	Data Set Audit	Primary Volume Audit	Migration Volume Audit	Backup Volume Audit	Master Catalog Audit	User Catalog Audit	Migration Control Data Set Audit	Backup Control Data Set Audit	Offline Control Data Set Audit
*ERR 01	D	C	D						
*ERR 02		C							
*ERR 03	D	D	C				C	C	C
*ERR 04	D				D	D		D	D

Table 75. Summary of errors detected or corrected by the listed AUDIT options (continued)

Error Number	Data Set Audit	Primary Volume Audit	Migration Volume Audit	Backup Volume Audit	Master Catalog Audit	User Catalog Audit	Migration Control Data Set Audit	Backup Control Data Set Audit	Offline Control Data Set Audit
*ERR 05		D	D						
*ERR 06	D				D	D	D	D	
*ERR 07		D							
*ERR 08		D	D						D
*ERR 09	D				D	D	D	D	
*ERR 10	D	D			D	D		D	D
*ERR 11	D	D	D		D	D	D	D	D
*ERR 12		D							
*ERR 14				C					C
*ERR 15	D	D	D						
*ERR 16	D		D		D	D	D	D	
*ERR 17	D		D		D	D	D	D	D
*ERR 18	D	D	D		D	D	D	D	D

Error codes (*ERR) and diagnosis

This topic lists the error conditions that AUDIT can detect.

For certain reported conditions involving control information (see topics One and Two), the AUDIT command can determine an Audit Repair Action. If the FIX option is specified, the AUDIT function takes the specified action. Where the verbs "patch," "create," or "delete" appear in the table, the AUDIT function generates a FIXCDS command (or series of commands) to make the specified changes. The following column also contains additional diagnostic information for some errors.

It is recommended that you direct AUDIT output to a specific data set, and that you review that output for errors that may have occurred when AUDIT tried to repair a problem.

For those cases where AUDIT cannot repair the problem, some Troubleshooting Hints have been provided.

Note: The error messages are listed sequentially, but not every number in the sequence has been used.

Table 76. Error codes used in AUDIT reports

Description	Audit Repair Action	Troubleshooting Hints
*ERR 01		
Neither the catalog nor the MCDS has a record of the data set being audited. This condition can also be caused by the user catalog being offline. This condition may not be an error.	For AUDIT VOLUMES (primary): AUDIT tries to catalog the data set.	If you are auditing a specific data set or level: <ul style="list-style-type: none">A dsname or qualifier may have been misspelled; if so, correct the misspelling and reissue the command.The data set may be uncataloged, in which case DFSMSHsm would not have a migration copy. If there is no misspelling, the catalog in which the data set is cataloged may be offline; if so, the catalog must be brought online before reissuing the command.

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
*ERR 02		
The catalog has no record of the data set being audited, but the MCDS record indicates that the data set is on a primary volume.	For AUDIT VOLUMES (primary): If the primary volume being audited is the same as the volume serial indicated in the MCDS record, AUDIT tries to catalog the data set.	
*ERR 03		
The catalog has no entry for the data set that is being audited, but the MCDS record indicates that the data set is on a migration volume.		<p>For AUDIT DSNAMEs:</p> <p>Because there is no catalog entry for the data set with the migration copy, that copy is most likely an old one that has not yet been scratched, probably because of a previous error when the data set was being recalled or deleted. There are two methods to correct the problem: (1) issue the HSEND DELETE command for the data set, deleting the D and A records in the MCDS immediately; or (2) have the migration copy and its D record removed during the next migration cleanup by using the FIXCDS D <i>dsname</i> PATCH() command to reset the assigned flag (MCDFASN) bit to 0 (off) and to set the needs scratch flag (MCDFNSCR) bit to 1 (on).</p> <pre>FIXCDS D <i>dsname</i> PATCH(X'06' BITS(0....1.))</pre> <p>This command <i>resets</i> the assigned bit and <i>sets</i> the needs scratch bit.</p> <p>Also possible: the catalog entry was inadvertently deleted, and the migrated copy is still valid. If this is the case, rebuild the catalog entry as a non-VSAM data set with a volume serial of "MIGRAT".</p>
*ERR 03 (continued)		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
The catalog has no record of the data set being audited, but the MCDS record indicates that the data set is on a migration volume.	<p>For AUDIT VOLUMES, MCDS, BCDS, or OCDS:</p> <p>AUDIT catalogs the data set and (for VSAM) any VSAM intercept object names associated with the data set, with a volume serial of MIGRAT.</p> <p>If the data set in question is a rolled-off generation data set (GDS), no error is reported, and the GDS is not recataloged.</p>	<p>For AUDIT VOLUMES (primary):</p> <p>Although there is no catalog entry for the migrated data set, there is an uncataloged data set with the same name on the primary volume being audited.</p> <p>If you can verify that there is no relationship between these two data sets, the <i>persistence</i> of the migration copy is probably caused by a previous error when the data set was being recalled or deleted. You can have the migration copy and its D record removed during the next migration cleanup by using the FIXCDS D <i>dsname</i> PATCH() command to reset the assigned flag (MCDFASN) and to set the needs scratch flag (MCDFNSCR).</p> <pre>FIXCDS D dsname PATCH(X'06' BITS(0....1.))</pre> <p>This command <i>resets</i> the assigned bit and <i>sets</i> the needs scratch bit.</p>
*ERR 04		
The catalog record of the data set being audited indicates that the data set is on a level 0 volume, but this could not be verified. No MCDS record of the data set exists for this error condition. This may or may not be an error.		If you can verify that the data set in question is on the volume indicated by the catalog entry, there is no problem. Otherwise, you should either delete the catalog entry or alter it to indicate the correct volume serial numbers.
*ERR 05		
The data set is cataloged on a volume other than the volume being audited. No MCDS record exists for the data set.		The volume being audited apparently has an uncataloged data set with the same name as a data set (on another volume) that <i>is</i> cataloged. You may want to rename the uncataloged data set, then catalog it also under its new name.
*ERR 06		
The catalog and the MCDS records for the data set being audited indicate that the data set is on a primary volume. The data set, however, is not on the volume recorded in the catalog, and the volume where the data set does reside cannot be determined.		<p>The volume in question may not be mounted.</p> <p>If you can verify which volume contains the data set in question, alter the catalog entry to indicate the correct volume. Otherwise, if you can verify that the data set is no longer needed, delete the catalog entry.</p> <p>The existence of the MCDS record indicates that migration cleanup has not yet removed that record of a previously recalled data set.</p>
*ERR 07		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
The serial number of the volume containing the data set being audited as recorded in the catalog does not agree with the serial number of the primary volume containing the data set as recorded in the MCDS. The two conflicting volume serial numbers are on the lists under the CATALOGED TO VOLUME and MIGRATED TO VOLUME headings, respectively.		This condition can occur when a migrated data set is recalled, moved to another primary volume, and the audit is done before migration cleanup has removed the MCDS record. After migration cleanup has removed the record, the AUDIT function no longer reports the condition.
*ERR 08		
The catalog indicates that a data set of the same name resides on a volume other than the one being audited. The MCDS record indicates a primary volume.		The volume being audited apparently has an uncataloged data set with the same name as a data set (on another volume) that <i>is</i> cataloged. You may want to rename the uncataloged data set, then catalog it also under its new name.
*ERR 09		
The catalog record of the data set being audited indicates that the data set is on a volume not managed by DFSMSHsm, but the MCDS indicates that the data set is on a migration volume.		<p>The catalog may have been offline to DFSMSHsm when the catalog entry for the data set in question was replaced with one for a nonmigrated (or recalled) data set. The condition could have also been caused by a DELETE NOSCRATCH operation. The MCDS record for the data set will have the serial number of the volume from which the data set was migrated.</p> <p>If you can determine that the current catalog entry has the correct status for the data set, use the <code>FIXCDS D dsname PATCH()</code> command to reset the assigned flag (MCDFA SN) and to set the needs scratch flag (MCDFN SCR) in the MCDS record; this causes the next migration cleanup to remove the migration copy.</p> <pre>FIXCDS D dsname PATCH(X'06' BITS(0....1.))</pre> <p>This command <i>resets</i> the assigned bit and <i>sets</i> the needs scratch bit.</p> <p>If, however, you determine that the correct status for the data set is a migrated copy, then alter the data set now referred to by the catalog entry to refer to volume MIGRAT.</p>
*ERR 10		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
<p>The catalog record of the data set being audited indicates that the data set is on a migration volume, but the MCDS has no record of the data set.</p>		<p>To indicate a migrated data set, DFSMSHsm uses the volume serial MIGRAT in the catalog entry for that data set. If someone in your organization catalogs data sets on a primary volume called MIGRAT, change that volume serial number and alter all related catalog entries to avoid confusion with DFSMSHsm's indicator.</p> <p>If no such primary volume MIGRAT exists, then you need to verify whether the data set in question is <i>supposed</i> to be migrated:</p> <ul style="list-style-type: none"> • If it is and you need to recall the data set, you need to locate the migration copy. (If no previous LIST output or activity log identifies the migration volume, do a LIST VTOC [for each DASD migration volume] or LIST TTOC [for each tape migration volume].) Then reconstruct the MCDS record, using the following command: <pre data-bbox="1045 905 1469 982">FIXCDS D <i>dsname</i> ADDMIG(<i>volser</i>)</pre> <p>Another way to rebuild the data set record is to run AUDIT MEDIACONTROLS, once you have determined the volume serial number. This may be required for successful recall of the data set.</p> <ul style="list-style-type: none"> • If the data set is <i>not supposed</i> to be migrated, either delete the catalog entry, or alter it to refer to a different (primary) volume.
*ERR 11		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
The catalog record of the data set being audited indicates that the data set is on a migration volume, but the MCDS record indicates that the data set is on a primary volume.		<p>To indicate a migrated data set, DFSMSHsm uses the volume serial MIGRAT in the catalog entry for that data set. If someone in your organization catalogs data sets on a primary volume called MIGRAT, change that volume serial number and alter all related catalog entries to avoid confusion with the DFSMSHsm indicator.</p> <p>If no such primary volume MIGRAT exists, then an error has apparently occurred during an attempted recall of the data set in question. You need to verify whether the data set is <i>supposed</i> to be migrated:</p> <ul style="list-style-type: none"> • If it is and you need to be able to recall the data set, use the <code>FIXCDS D dsname DISPLAY</code> command to determine from the MCDS record what migration volume has held the migration copy, verify (with the <code>LIST</code> or <code>AUDIT</code> command) that the migration copy is still on that migration volume, and set the assigned flag (MCDFASN) in the MCDS record by using a <code>FIXCDS D dsname PATCH()</code> command. <p>This command sets the assigned bit.</p> <pre>FIXCDS D dsname PATCH(X'06' BITS(1.....))</pre>
*ERR 12		
The catalog and MCDS records of the data set being audited indicate that the data set is on a migration volume, but the mounted volume table entry indicates that the volume is not a migration volume. (The data set is apparently a very old migration copy, made before HSM Release 3.)		<p>Most likely an uncataloged data set on the volume has the same name as a (very old) cataloged migrated data set. This is not an error, but you should catalog all your data sets.</p> <p>Otherwise, if you can determine (from lists of migration volumes, and so forth) that DFSMSHsm <i>should</i> have considered the volume as a migration volume, then use the <code>FIXCDS V volser PATCH()</code> command to set the flag in the volume record that identifies the volume as a migration volume rather than a level-0 volume.</p>
*ERR 14		
The BCDS backup version record of the data set being audited specifies a serial number that does not match the serial number of the backup volume being audited. If the data set resides on a tape volume, this error is detected only if the volume is also not spanned by the data set being audited.	For a tape backup volume, AUDIT sets the invalid flag in the TTOC entry, and in the TTOC for any spanned volume.	

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
*ERR 15		
Either the catalog entry for this data set indicates that the type of data set is not supported by DFSMSHsm migration, or the entry did not indicate a volume for the data set.		<p>For AUDIT DSNAMEs: Most likely this error is a typing error in specifying a particular data set to be audited; the object cataloged by that name is not one that is supported by DFSMSHsm. Correct the spelling of the <i>dsname</i> and reissue the AUDIT command.</p> <p>For AUDIT VOLUMES: Since the catalog indicates the data set is something not supported by migration (for instance, a VSAM page space), this may not be an error. Even so, it is probably best to move the data set to another volume.</p>
*ERR 16		
<p>The TTOC records do not coincide with the catalog and the MCDS data set record. One of the following conditions may have occurred:</p> <ul style="list-style-type: none"> • The catalog and the MCDS data set record say the data set is migrated. The volume serial number of the TTOC, however, does not match the volume serial number in the MCDS data set record, or (according to that record) of any volumes the data set might span. • The catalog and the MCDS data set record indicate that the data set is on a specific primary volume. The TTOC, however, contains a valid data set entry for the data set, which means the data set has been migrated (and may have been recalled). • If the DATA SET ON VOLUME field contains “INVALID”, the TTOC indicates that the data set copy is invalid. • If the DATA SET ON VOLUME field contains “NO ENT”, one of the following conditions is true: <ul style="list-style-type: none"> – If the tape being audited is in single-file format, the data set cannot be found in the TTOC extension identified by the MCD record. – If the tape being audited is a multifile tape, the file block identifiers (FBID) in the TTOC do not coincide with the FBID in the MCD record. This situation may be resolved by recalling and again migrating the data set in error. 		<p>The TTOC entry may identify a previous migration copy. A system failure most likely occurred before DFSMSHsm was able to invalidate it. You can use the FIXCDS T command to invalidate the entry for the data set in question:</p> <pre>FIXCDS T key ENTRY(dsname) PATCH(X'34' BITS(.0.....))</pre> <p>Verify that the migration volume referred to in the data set record actually contains a migration copy of the data set. (If the volume is a tape volume, you can issue a LIST TTOC command.)</p>
*ERR 17		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
<p>Here are the possible error conditions AUDIT reports and their meanings:</p> <p>OFFLINE The catalog cannot be opened, and is probably offline.</p> <p>NO TTC The TTOC record of the migration volume or backup volume as indicated in the data set record does not exist. The missing TTOC record can be either the base TTOC record or an extension TTOC record.</p> <p>-SEQ1- The first data set file sequence number in the base TTOC record is greater than the data set file sequence number indicated in the MCDS data set record.</p> <p>-SEQ1- The last data set file sequence number in the base TTOC record is less than the data set file sequence number indicated in the MCDS data set record for a non-VSAM data set.</p> <p>-SEQ1- The last data set file sequence number in the base TTOC record plus one is less than the data set file sequence number indicated in the MCDS record for a VSAM data set.</p> <p>-SEQ2- The TTOC record shows a data set name based on the data set file sequence number found in the MCDS data set record. This data set name in the TTOC does not match the data set name in the MCDS data set record.</p>		<p>Check that the catalog is online to DFSMSHsm while the AUDIT command is running; an offline catalog can cause this error indication.</p> <p>If the catalog is <i>not</i> offline, then diagnosis depends on the tape format involved:</p> <ul style="list-style-type: none"> • If the tape is in single-file format, run the AUDIT MEDCTL command against the tape volume to determine, by checking the tape volume, whether the MCDS data set record or the TTOC record set is in error. (The FIX option will regenerate a TTOC set.) • If the tape is in multifile format, the TTOC entry most likely represents an out of date copy and can be invalidated by using the FIXCDS T command: <pre>FIXCDS T key ENTRY(dsname) PATCH(X'34' BITS(.0.....))</pre>
*ERR 18		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
The catalog record of the data set being audited indicates that the data set is on a primary volume, but the MCDS indicates that the data set is on a migration volume.		<p>The catalog may have been offline to DFSMSHsm when the catalog entry for the data set in question was replaced with one for a nonmigrated (or recalled) data set. The MCDS record for the data set will have the serial number of the volume from which the data set was migrated.</p> <p>If you can determine that the current catalog entry has the correct status for the data set, then use the FIXCDS D <i>dsname</i> PATCH() command to reset the assigned flag (MCDFASN) and to set the needs scratch flag (MCDFNSCR) in the MCDS record; this causes the next migration cleanup to remove the migration copy.</p> <pre>FIXCDS D <i>dsname</i> PATCH(X'06' BITS(0.....1.))</pre> <p>This command <i>resets</i> the assigned bit and <i>sets</i> the needs scratch bit.</p> <p>If, however, you determine that the correct status for the data set is a migrated copy, then rename and recatalog the data set now referred to by the catalog and alter the original catalog entry to refer to volume MIGRAT.</p>
*ERR 19 userdsn HAS NO D RECORD		
Data set <i>userdsn</i> is not migrated. No data set (D) record can be found.		The data set name may be spelled incorrectly.
*ERR 20 userdsn ON migvol, BUT NO V RECORD		
The catalog indicates that data set <i>userdsn</i> has been migrated and the MCDS indicates that the migration copy is on a specific migration volume (<i>migvol</i>); but there is no migration volume (V) record for that volume.	<p>If the D record is flagged to indicate the copy is in an SDSP data set, and</p> <ul style="list-style-type: none"> • If an SDSP data set can be found on the volume, an ADDVOL command is generated for the L1 volume. • If no SDSP data set can be found on the volume, but the migration copy can be found there, the D record is patched to reset the SDSP flag, and an ADDVOL command is generated for the L1 volume. <p>If the D record is not flagged for a copy in an SDSP data set, and the migration copy can be found on the volume, an ADDVOL command is generated for the L1 volume.</p>	
*ERR 21 userdsn INDICATES DEVICE TYPE X'...', V RECORD INDICATES X'!..'		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
For user data set <i>userdsn</i> , a data set (D) record exists, indicating that the migration copy is on a migration volume of device type X'...', but the volume (V) record for that volume indicates a different device type X'...'.	If the volume record indicates tape, and a TTOC record exists, the data set (D) record is patched to replace the device type. If the volume record indicates DASD and the migration copy can be found on the volume, the data set (D) record is patched to replace the device type.	
*ERR 22 <i>userdsn</i> HAS NO ALIAS RECORD		
The data set (D) record for user data set <i>userdsn</i> indicates that the migration copy is not in an SDSP data set. Therefore, there should be a MCDS alias record identifying the migration copy; but the MCDS alias record does not exist.	If the migration copy can be found on the volume, an alias (A) record is created.	
*ERR 23 <i>userdsn</i>—VSAM ASSOCIATIONS RECORD MISSING		
The MCDS data set (D) record for user data set <i>userdsn</i> indicates the existence of a MCDS VSAM association (O) record, but no such record exists.		If the VSAM data set has already been recalled, then the D record is extraneous; you can use the FIXCDS command to delete it.
*ERR 24 <i>userdsn</i>—MISSING VSAM INTERCEPT OBJECT		
The MCDS VSAM associations (O) record for the migration copy of data set <i>userdsn</i> indicates the existence of an intercept object representing one of the components of the VSAM data set. The catalog entry for this object is <i>not</i> a non-VSAM entry indicating MIGRAT as its only volume serial.	The VSAM Association (O) record is patched, to correspond to the actual components found.	
*ERR 25 <i>userdsn</i>—component MIGRATED, BUT NO ALIAS RECORD		
The VSAM associations (O) record for the migration copy of data set <i>userdsn</i> indicates the existence of an intercept object representing a <i>component</i> of the VSAM data set. This <i>component</i> has a non-ICF catalog entry with a volume serial of MIGRAT, but the MCDS alias (A) record does not exist.	The alias (A) record for the component is created.	
*ERR 26 <i>userdsn</i>—L RECORD MISSING		
The MCDS data set record for data set <i>userdsn</i> indicates that a BCDS backup migrated data set (L) record has been created, but no such L record exists.	If a BCDS data set (B) record is found that indicates a backup was made after the migration date, the MCDS data set (D) record is patched to reset the L-record flag. If no such B record is found and the D record indicates the migration copy is on L1, a BACKDS command is generated for the data set.	
*ERR 27 <i>userdsn</i>—EXTRANEIOUS VSAM ASSOCIATIONS RECORD		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
A VSAM-associations (O) record refers to an MCDS data set (D) record for data set <i>userdsn</i> . Either the data set record does not exist, or (if present) does not indicate the existence of a VSAM-associations record.	The VSAM Association (O) record is deleted.	
*ERR 28 <i>userdsn</i>—EXTRANEOUS ALIAS RECORD		
An extraneous MCDS alias (A) record is referring to an MCDS data set (D) record for data set <i>userdsn</i> , but the MCDS data set (D) record does not exist.	If there is no data set (D) record, <ul style="list-style-type: none"> • The alias (A) record is deleted; • If the A record indicates a VSAM component, and the catalog entry for that component indicates that it is migrated, the catalog entry is deleted. 	
*ERR 29 <i>userdsn</i>—NO TTOC RECORD		
An MCDS data set (D) record for a migration copy of data set <i>userdsn</i> indicates the migration copy is on a migration tape volume. Either no TTOC record exists for the migration volume, or no TTOC extension contains a valid entry for the migration copy.		You can run AUDIT MEDCTL against the migration tape in the D record, and (if FIX is specified) AUDIT will rebuild the TTOC. This error may occur if AUDIT DATASETCONTROLS is executed while migration is currently updating a given TTOC record. If this occurs, run AUDIT DATASETCONTROLS while migration is inactive.
*ERR 30 INVALID CDS RECORD TYPE		
An invalid record type has been detected in the control data set (MCDS or BCDS) being audited.	The unknown record is deleted (internally).	
*ERR 31 <i>userdsn</i> HAS NO MIGRATED DATA SET RECORD EXTENSION		
The MCD record (D) indicates that a user data set has been encrypted, but there is no MCDX record (3).	No audit repair can be made.	If the data set has already been recalled, then the D record is extraneous; you can use the FIXCDS command to delete it. If the key label is known the FIXCDS command can be used to add the MCDX record (3).
*ERR 32 <i>userdsn</i>—EXTRANEOUS MIGRATED DATA SET RECORD EXTENSION		
An extraneous MCDX record (3) is referring to a MCD record (D) that does not exist or an extraneous MCDX record (3) exists where the MCDX indicator is not enabled in the associated MCD record (D).	If there is no MCD record (D) or the MCD record (D) exists but the MCDX indicator is not enabled, the MCDX record (3) is deleted.	
*ERR 33 <i>userdsn</i>— MISMATCH NUMBER OF RETAINED MCA RECORDS, FOUND <i>num1</i>, EXPECTED <i>num2</i>		
The MCD record (D) reports <i>num1</i> 'Retained' MCA records, but found <i>num2</i>	No audit repair can be made.	Use the DFSMSHsm LIST command to validate 'Retained' MCA records. Delete invalid records, and update the 'Retained' MCA list to reflect the deletion.
*ERR 35 MCB VERSION FIELD FOR <i>dsn</i> NEEDS VERSION NUMBER <i>vernum</i>		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
<p>All MCC (backup version) records that are associated with the specified MCB (data set backup) record have been audited. During the audit of these records, the most current version found based on the MCC record timestamp has been tracked. The MCBGEN field should contain the version number of the most current MCC record, but it does not.</p> <p>If the suggested version number is 1, then in most cases this indicates that the actual version number could not be determined, so versioning is restarted. This situation occurs only if an existing version number cannot be established.</p>	A FIXCDS PATCH is issued to update the MCBGEN field to the highest version number.	None.
*ERR 36 BACKUP VERSION ENTRY <i>backupdsn</i> CONTAINS (<i>mcbver</i>), SHOULD CONTAIN (<i>mccgen</i>)		
The MCC record and associated MCB record have a discrepancy between the MCBVER and MCCGEN fields for the specified version entry. The MCBVER field needs to be modified to be the same as the MCCGEN field for the version entry.	A FIXCDS PATCH is issued to update the MCBVER field to the proper value, which is the value contained in the MCCGEN field of the associated MCC record.	None.
*ERR 37 <MCBR key> BAD SEQUENCE NUMBER		
The sequence number in the MCBR (Z) record and MCC (C) record should be identical, but they are not. Either the data set backup retained copy (Z) record or the data set backup version (C) record are in error.		<p>You can display the MCBR (Z) record using the following command:</p> <pre>FIXCDS Z <MCBR key> INDEX(n) DISPLAY</pre> <p>The default for INDEX(n) is INDEX(1), which will display the most current MCBR record for that data set. See <i>z/OS DFSMSHsm Diagnosis</i> for more information on using the FIXCDS command.</p> <p>You can display the MCC (C) records using the following command:</p> <pre>FIXCDS C <MCC key> DISPLAY</pre> <p>You can find the MCBR key in the ERR37 message text. You can find the MCC keys in the output of the FIXCDS Z DISPLAY command. All MCC records listed in the MCBR version array should contain the same sequence number. Each MCC record needs to be FIXCDS displayed to find its sequence number. You need to consider all of the sequence number information when determining which record contains the incorrect sequence number. You can use the FIXCDS PATCH command to correct the record with the correct sequence number.</p>

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
*ERR 38 <MCC OR MCB key> HAS INVALID SEQUENCE NUMBER		
The dataset backup record contains an indication of the number of dataset backup retained copy records that exist. The number is incorrect.		
*ERR 39 <i>userdsn</i> IS MISSING <i>version</i>		
The BCDS data set (B) record for <i>userdsn</i> refers to a backup <i>version</i> of that data set. However, the corresponding backup version (C) record no longer exists or exists but does not refer back to this (B) record.		<p>The time and date stamp incorporated in the version name may indicate a version old enough that it is no longer useful, and the copy referred to by that version name may no longer exist.</p> <p>If a LIST BCDS LEVEL(<i>userdsn</i>) indicates there are other available versions of <i>userdsn</i>, you may not need the missing version.</p> <p>If there is no other available version, you need to determine (for example, from a previous LIST output) what backup volume contains the version, so that you can use the "FIXCDS C <i>version</i> ..." command to regenerate the C record.</p> <p>You should discover what caused the deletion of the C record and correct that problem to avoid future occurrences.</p>
*ERR 40 <i>userdsn</i> - backup version name MISSING B RECORD		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
<p>The backup version (C) record refers to a BCDS data set record that does not exist, or the (B) record exists but does not refer to the backup version (C) record.</p> <p>In the message, <i>userdsn</i> is the original data set name in the (B) record and version is the generated backup name in the (C) record.</p> <p>Note: Each FIXCDS command generated with this error is dependent on previous FIXCDS commands. If NOFIX is used, the offsets in the output data set are invalid.</p>	<p>If the backup version (C) record indicates the version is invalid, then a BDELETE command is issued to delete the backup version (C) record.</p> <p>If the backup version (C) record indicates the backup version is for an uncataloged data set or a data set with the RACF erase on scratch option, then no action is taken – regardless of whether there is a valid data set (B) record or not.</p> <p>If the backup version (C) record references a data set (B) record that does not exist, then a new (B) record is created and the version added.</p> <p>If the data set (B) record referenced by the backup version (C) record exists but does not correspond to the specified backup version, then one of the following actions will be performed:</p> <ul style="list-style-type: none"> • If the maximum number of backup versions has been reached and the specified backup version is older than any of those described in the data set record, the backup version is deleted. • If the maximum number of backup versions has not been reached, then the data set (B) record is extended. <ul style="list-style-type: none"> – If the version is older than any of those described in the (B) record, then it is added at the end. – If the version is not the oldest, then it is inserted in date order. 	<p>The number of versions specified by SETSYS VERSIONS or in the management class is ignored and new versions might be added until the system maximum has been reached. Subsequent BACKUP, EXPIREBV, or BDELETE commands can be used to cleanup unwanted backup versions. The RECYCLE command must be issued to delete the versions from the backup volumes.</p> <p>GENERATION and VERSION numbers are not updated in the data set (B) record as backup versions are added. Different backup versions can have duplicate numbers.</p> <p>Backup versions are added to the (B) record in date order. If multiple backup versions exist for the same date, they are not specifically ordered by time.</p> <p>Cataloged backup versions take precedence over uncataloged backup versions with newer dates.</p>
*ERR 41 version INDICATED ON <i>backvol</i>, BUT NO X RECORD		
<p>The backup version (C) record for <i>version</i> indicates the version is on backup volume <i>backvol</i>, but no BCDS backup volume (X) record exists for <i>backvol</i>.</p>	<p>If the volume is found to be an L1 volume, the data set (B) record is patched to indicate L1 residence of this version.</p> <p>If the volume is the second or later of a multivolume version, an ADDVOL command is generated for volume serial <i>backvol</i>.</p>	
*ERR 42 version INDICATED ON <i>migvol</i>, BUT NO V RECORD		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
A backup version (C) record indicates <i>version</i> is on migration volume <i>migvol</i> (that is, it has been created by the (H)BACKDS command), but no MCDS volume record (V) exists for <i>migvol</i> .	If the volume is found to be a backup volume, the data set (B) record is patched to indicate the version is not on L1. Otherwise, an ADDVOL command is generated for the L1 volume.	
*ERR 43 <i>version</i> RESIDES ON <i>migvol</i>, BUT NO M RECORD		
*ERR 43 <i>migvol</i> - CONTAINS <i>version</i>, BUT NO M RECORD		
Backup version <i>version</i> exists on migration volume <i>migvol</i> , but no BCDS move backup version (M) record exists for moving the version to a backup volume.	<p>If a backup version (C) record exists for the version, and</p> <ul style="list-style-type: none"> • If a BCDS data set (B) record referring to the version exists, an M record is created. • If no such B record exists, the C record is patched for subsequent deletion and scratching of the backup version. 	
*ERR 44 <i>version</i> INDICATES DEVICE TYPE X'...', X RECORD INDICATES X'...'		
The BCDS backup version (C) record for <i>version</i> indicates the version is on a backup volume of device type X'...', but the volume (X) record for that volume indicates a different device type X'...'.	<p>If the volume record indicates tape, and a TTOC record exists, the backup version record is patched to replace the device type.</p> <p>If the volume record indicates DASD, and the backup version can be found on the volume, the backup version record is patched to replace the device type.</p>	
*ERR 45 <i>version</i> - NO TTOC ENTRY		
A BCDS backup version (C) record for <i>version</i> indicates the version is on a backup tape volume. Either no TTOC record exists for the backup volume, or no TTOC extension contains a valid entry for the backup version.		
*ERR 46 <i>userdsn</i> - BAD L RECORD		
A backup migrated data set (L) record refers to a migration copy of data set <i>userdsn</i> , but no MCDS data set (D) record exists.	The L record is deleted.	
*ERR 47 <i>version</i> INDICATED ON <i>migvol</i>, BUT CANNOT BE FOUND		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
The BCDS data set (B) record indicates that <i>version</i> is on migration volume <i>migvol</i> and a backup version (C) record exists, but either <i>migvol</i> is not mounted or it does not contain the backup version.		<p>If the version is no longer needed, no action need be taken.</p> <p>If the migration volume is mounted, but you know the version is on a different volume, you can use the FIXCDS command to update volume serial numbers in the C record for the version.</p> <p>If the migration volume is not mounted, it should be mounted to provide access to the version.</p>
*ERR 48 <i>userdsn</i> HAS NO B RECORD		
Data set <i>userdsn</i> , specified as a parameter for auditing backup controls, is not backed up (does not have a BCDS data set (B) record).		The data set name may be spelled incorrectly.
During AUDIT MEDIACONTROLS VOLUMES processing, a data set backup version for <i>userdsn</i> was found on the tape volume specified. There is no BCDS data set (B) record for <i>userdsn</i> .		If a backup version is required for data set <i>userdsn</i> , use the BACKDS command to create a current backup version.
*ERR 50 <i>volser</i> NOT A MIGRATION VOLUME [(DUMP BACKUP LEVEL-0)]		
<p>Specified volume <i>volser</i> is not a migration volume as there is no MCDS migration volume record.</p> <ul style="list-style-type: none"> If volume <i>volser</i> is a BCDS dump volume record, it will be reported as (DUMP). If volume <i>volser</i> is a BCDS backup volume record, it will be reported as (BACKUP). If volume <i>volser</i> is a level-0 volume, it will be reported as (LEVEL-0). 		<p>The volume serial specified in the command may be spelled incorrectly.</p> <p>If not, and the volume is identified as a different type of DFSMSHsm-owned or DFSMSHsm-managed volume, you cannot use this AUDIT option on the volume.</p> <p>If the volume serial is not misspelled and you have evidence that the volume was once managed by DFSMSHsm, you need to check DFSMSHsm logs to determine why the volume was deleted.</p>
*ERR 51 <i>volser</i> - V RECORD INDICATES TAPE, NO TTOC EXISTS		
A migration (V) record for volume <i>volser</i> exists, but no TTOC base record is found.	<p>If the volume record indicates</p> <ul style="list-style-type: none"> A tape <i>other than</i> one in single-file format, and An empty volume <p>an ADDVOL command is generated for <i>volser</i>.</p> <p>If the volume record indicates a tape in single-file format, the base TTOC is created, and individual data set TTOC entries are created as the tape is read.</p>	
*ERR 52 <i>volser</i> - V RECORD INDICATES NO RACF, TTOC INDICATES RACF		
*ERR 52 <i>volser</i> - X RECORD INDICATES NO RACF, TTOC INDICATES RACF		
*ERR 52 <i>volser</i> - X RECORD INDICATES RACF, TTOC INDICATES NO RACF		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
For volume <i>volser</i> , there is a conflict between either the MCDS volume (V) record or the BCDS backup volume (X) record and the TTOC. The V record or the X record for <i>volser</i> indicates a RACF protected tape, but the TTOC record does not, or vice versa.	If RACF indicates that the volume is RACF protected, either the TTOC or the volume record is patched to indicate that protection.	
*ERR 53 <i>volser</i> - RACF-INDICATED, BUT NOT IN DFSMSHsm'S TAPEVOL		
*ERR 53 <i>volser</i> - IN DFSMSHsm'S TAPEVOL, BUT NOT RACF INDICATED		
For volume <i>volser</i> , there is a conflict between either the MCDS volume (V) record or the BCDS backup volume (X) record and DFSMSHsm's TAPEVOL volume set. The V record or the X record for <i>volser</i> indicate a RACF protected tape, but DFSMSHsm's TAPEVOL volume set does not, or vice versa.	If RACF indicates that the volume is not in DFSMSHsm's tape volume set, RACF is invoked to add the volume to the set; otherwise, the volume record is patched to indicate RACF protection.	
*ERR 54 <i>volser</i> IN L2CR RECORD, NO V RECORD		
Volume <i>volser</i> , which is in the L2CR record, does not have a MCDS volume (V) record.		If the volume is not intended to be (or is no longer) a DASD level-2 migration volume, you can use the FIXCDS command to remove the entry for the volume from the L2CR record or use the DEFINE ML2 to redefine the list of volumes to be associated with the key range involved. Otherwise, issue an ADDVOL command for the volume, to reidentify it to DFSMSHsm as a DASD level-2 migration volume.
*ERR 55 <i>volser</i> IN L2CR RECORD NOT DASD L2 VOLUME		
Volume <i>volser</i> in the L2CR record has a corresponding MCDS volume (V) record that indicates a volume type other than DASD (L2).		If the V record for the volume correctly indicates that the volume is not a DASD level-2 migration volume, you can use the FIXCDS command to remove the entry for the volume from the L2CR record or use the DEFINE ML2 to redefine the list of volumes to be associated with the key range involved. Otherwise, use the FIXCDS command to correct the V record so that the volume is identified as a DASD level-2 migration volume.
*ERR 56 MC1 EXTENSION <i>nnnn</i> MISSING		
The last MC1 migration level-1 free space (1) record incorrectly indicates that there is another extension record <i>nnnn</i> .	The last MC1 extension is patched to indicate no continuation.	
*ERR 57 MC1 EXTENSION <i>nnnn</i> EXTRANEIOUS		
There is an extraneous extension record <i>nnnn</i> in the MCDS migration level-1 free space record.	The extraneous MC1 extension is deleted.	
*ERR 58 <i>orig volser</i> ALTERNATE TAPE VOLSERS MISMATCHED: TTOC <i>ttoc volser</i> {MCT <i>mct volser</i> MCV <i>mcv volser</i>}		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
The alternate volume serials for duplex volumes do not match between the TTOC <i>ttoc volser</i> and either the MCT <i>mct volser</i> or the MCV <i>mcv volser</i> .		
*ERR 70 volser NOT A BACKUP VOLUME [(DUMP MIGRATION LEVEL-0)]		
Volume <i>volser</i> is not a backup volume as there is no BCDS backup volume record. <ul style="list-style-type: none"> If volume <i>volser</i> is a BCDS dump volume, it is reported as (DUMP). If volume <i>volser</i> is a MCDS migration volume, it is reported as (MIGRATION). If volume <i>volser</i> is a level-0 volume, it is reported as (LEVEL-0). 		<p>The volume serial specified in the command may be spelled incorrectly.</p> <p>If not, and the volume is identified as a different type of DFSMSHsm-owned or DFSMSHsm-managed volume, you cannot use this AUDIT option on the volume.</p> <p>If the volume serial is not misspelled and you have evidence that the volume was once known to DFSMSHsm, you need to check DFSMSHsm logs to determine why the volume was deleted.</p>
*ERR 71 volser - TTOC EXISTS, NO X RECORD		
The TTOC base record was found for volume <i>volser</i> , but the backup volume (X) record is missing.		
*ERR 71 volser - X RECORD INDICATES TAPE, NO TTOC EXISTS		
A backup volume (X) record for volume <i>volser</i> exists, but no TTOC base record has been found.	<p>For VOLUMECONTROLS(BACKUP) If the X record indicates an empty volume, an internal ADDVOL command is generated for <i>volser</i>.</p> <p>For MEDIACONTROLS:The base TTOC is created, and individual data set TTOC entries are created as the tape is read.</p>	
*ERR 72 volser - X RECORD - {SP UN DAY nn}, BVR - {UN DAY nn SP}		
<p>A BVR is a BCDS backup cycle volume record subdivided into spill and day of backup cycle (daily).</p> <p>A BVR lists a backup volume <i>volser</i>, but there is a mismatch between the BCDS backup volume (X) record type {SP UN DAY nn} and the BVR type {UN DAY nn SP}.</p>	The BVR entry is deleted.	
*ERR 73 volser - X RECORD - {SP UN DAY nn}, TTOC - {UN DAY nn SP}		
The usage for volume <i>volser</i> in the BCDS backup volume (X) record is inconsistent with the {SP UN DAY nn} usage in the TTOC record key.		Depending on your intended backup usage for the volume, use the FIXCDS command to correct either the TTOC record set (the base and any extensions), or the X record.
*ERR 74 volser - X RECORD INDICATES TAPE, BVR ENTRY INDICATES DASD		
*ERR 74 volser - X RECORD INDICATES DASD, BVR ENTRY INDICATES TAPE		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
<p>A BVR is a BCDS backup cycle volume record subdivided into spill and day of backup cycle (daily).</p> <p>The BVR record entry does not agree with the BCDS backup volume (X) record as to whether volume <i>volser</i> is a DASD or tape volume.</p>	The BVR entry is patched to agree with the volume record.	
*ERR 75 <i>volser</i> - X RECORD INDICATES FULL, BVR ENTRY INDICATES NOT FULL		
*ERR 75 <i>volser</i> - X RECORD INDICATES NOT FULL, BVR ENTRY INDICATES FULL		
<p>A BVR is a BCDS backup cycle volume record subdivided into spill and day of backup cycle (daily).</p> <p>There is a conflict between a BVR entry and a BCDS backup volume (X) record, where the X record indicates <i>full</i> but the BVR record indicates <i>not full</i>, or vice versa.</p>	The BVR entry is patched to agree with the volume record.	
*ERR 76 <i>volser</i> - X RECORD INDICATES SINGLE-FILE TAPE, BVR IS MULTI-FILE		
*ERR 76 <i>volser</i> - X RECORD INDICATES MULTI-FILE TAPE, BVR IS SINGLE-FILE		
<p>A BVR is a BCDS backup cycle volume record subdivided into spill and day of backup cycle (daily).</p> <p>There is a conflict between a BVR entry and a BCDS backup volume (X) record, for volume <i>volser</i>: The X record indicates a multifile format whereas the BVR entry indicates a single-file format, or vice versa.</p>	The BVR entry is patched to agree with the volume record.	
*ERR 77 <i>volser</i> - X RECORD, NO BVR ENTRY		
<p>A BVR is a BCDS backup cycle volume record subdivided into spill and day of backup cycle (daily).</p> <p>A BVR entry exists for volume <i>volser</i>, but there is no BCDS backup volume (X) record for the volume, or vice versa.</p>	<p>If the BVR entry is missing, an ADDVOL command is generated for the volume, to create the missing entry.</p> <p>If the volume record is missing, the BVR entry is deleted.</p>	
*ERR 78 BVRxx EXTENSION <i>nnnn</i> MISSING		
<p>A BVR is a BCDS backup cycle volume record subdivided into spill and day of backup cycle (daily).</p> <p>In the BVR record set, the last entry indicates that there is another extension (<i>nnnn</i>), but the extension does not exist.</p>	The missing BVR extension is created.	
*ERR 79 BVRxx EXTENSION <i>nnnn</i> EXTRANEOUS		
<p>A BVR is a BCDS backup cycle volume record subdivided into spill and day of backup cycle (daily).</p> <p>There is an extraneous extension <i>nnnn</i> in the BVR record set <i>BVRxx</i>.</p>	The previous BVR extension is patched to indicate the continuation.	

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
*ERR 90 <i>volser</i> - NO V OR X RECORD		
An eligible volume (P) record indicates that volume <i>volser</i> has been backed up or dumped and is a DFSMSHsm volume, but neither a BCDS backup volume (X) record nor a MCDS migration volume (V) record exists for volume <i>volser</i> .		<p>If the P record represents a DFSMSHsm-owned volume that was backed up at some point and then deleted, the P record can be left as it is, in case you need to recover the volume.</p> <p>If the volume is not a DFSMSHsm-owned volume, you can use the FIXCDS command to reset the DFSMSHsm-owned flag in the P record.</p>
*ERR 91 <i>volser</i> - <i>vtoccopy</i> INDICATED ON <i>migvol</i>, BUT NO V RECORD		
The BCDS eligible volume (P) record for volume <i>volser</i> indicates that VTOC copy data set <i>vtoccopy</i> is on migration volume <i>migvol</i> , but no corresponding MCDS volume (V) record exists for <i>migvol</i> .	If the VTOC copy can be found on the volume, an ADDVOL command is generated for an L1 volume.	
*ERR 92 <i>volser</i> - <i>vtoccopy</i> INDICATED ON <i>backvol</i>, BUT NO X RECORD		
The BCDS eligible volume (P) record for volume <i>volser</i> indicates that VTOC copy data set <i>vtoccopy</i> is on backup volume <i>backvol</i> , but no corresponding BCDS backup volume (X) record exists for <i>backvol</i> .		If you can confirm that the VTOC copy data set still resides on the volume, then you can add the volume as a backup volume, using the ADDVOL command.
*ERR 93 <i>volser</i> - <i>vtoccopy</i> INDICATED SPANNING TO <i>backvol</i>, BUT NO X RECORD		
The BCDS eligible volume (P) record for volume <i>volser</i> indicates that VTOC copy data set <i>vtoccopy</i> spans to a second volume (<i>backvol</i>). No BCDS backup volume (X) record exists, however, for <i>backvol</i> .		The data set is probably an obsolete VTOC copy data set. If the volume indicated by qualifier “.Vvolser” has not been backed up (or dumped) since the date indicated in qualifier “.Dyyddd”, issue the BACKVOL command for the volume.
*ERR 94 <i>volser</i> - NO DGN RECORD FOR <i>generation</i>		
The BCDS eligible volume record for volume <i>volser</i> refers to dump generation <i>generation</i> , but no corresponding dump generation (G) record exists.		<p>If *ERR 103 occurs for the volume in *ERR 94, use the FIXCDS command to correct either the G record or the <i>generation</i> entry in the P record.</p> <p>If you also see an ERR 104 message referring to this generation, it will identify the dump volume that contained the missing generation.</p>
*ERR 95 <i>volser</i> - NO DVL RECORD FOR <i>dumpvol</i> IN COPY <i>n</i> OF <i>generation</i>		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
The BCDS eligible volume record for volume <i>volser</i> indicates one or more dump generation record keys. Dump generation copy <i>n</i> of dump generation <i>generation</i> describes a dump volume (<i>dumpvol</i>). No BCDS dump volume record exists for <i>dumpvol</i> .		You should use the ADDVOL command to identify the volume as a dump volume. If the generation contains other dump copies or if you have no need to restore data from that generation (for example, there is a later generation), no further action is needed. Otherwise, use the FIXCDS command to update the Y record for the volume, using the dump class and volume sequence data from the dump copy information in the G record.
*ERR 96 <i>volser</i> - INVALID DVL RECORD FOR <i>dumpvol</i> IN COPY <i>n</i> OF <i>generation</i>		
For volume <i>volser</i> , there is an invalid BCDS dump volume record for dump volume <i>dumpvol</i> in copy <i>n</i> . This is described in the dump generation (G) record for <i>generation</i> .		If you know (for example, from DFSMSHsm LIST output) that the volume does contain part of a dump copy for <i>generation</i> , use the FIXCDS command to turn on the valid flag in the Y record for the volume.
*ERR 97 <i>volser</i> - BAD CHAIN FROM <i>generation</i> TO <i>dumpvol</i>		
1) The dump generation (G) record for <i>generation</i> refers to dump volume <i>dumpvol</i> as holding part of a dump copy, but the dump volume (Y) record for <i>dumpvol</i> contains part of a different generation. 2) The position of dump <i>volser dumpvol</i> in the array of volumes for its dump copy in the (G) record is inconsistent with the position of the entry in the (Y) record which refers to that generation.		If you know (for example, from DFSMSHsm LIST output) that the volume does contain part of a dump copy for <i>generation</i> , use the FIXCDS command to update the Y record for the volume to refer to that generation. You may also see an *ERR 98 message for the dump volume.
*ERR 98 <i>volser</i> - BAD CHAIN FROM <i>dumpvol</i> TO <i>generation</i>		
The dump volume (Y) record for <i>dumpvol</i> indicates that the volume contains part of a dump copy in generation <i>generation</i> , but the dump generation (G) record for <i>generation</i> does not refer to volume <i>dumpvol</i> .		If you also see an *ERR 97 message referring to the volume, use the FIXCDS command to update the Y record for the volume to refer to the <i>generation</i> mentioned in the *ERR 97 message.
*ERR 99 <i>volser</i> - CLASS <i>class1</i> OF <i>dumpvol</i>, CLASS <i>class2</i> IN <i>generation</i>		
For dump generation <i>generation</i> of eligible volume <i>volser</i> , dump volume <i>dumpvol</i> is supposed to contain part of a dump copy. However, the CLASS (<i>class1</i>) listed in the BCDS dump volume record and the CLASS (<i>class2</i>) listed in the BCDS dump generation record are not the same.		Based on your knowledge from previous DFSMSHsm output of the intended dump class for the dump volume, use the FIXCDS command to correct the dump class name in either the Y record for the volume or the G record for <i>generation</i> .
*ERR 100 <i>volser</i> - BAD VOL SEQ FOR <i>dumpvol</i> IN COPY <i>n</i> OF <i>generation</i>		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
For eligible volume <i>volser</i> , the volume sequence number for dump volume <i>dumpvol</i> in the dump volume record does not agree with the sequence number for that volume in copy <i>n</i> in the dump generation (G) record for <i>generation</i> .		Based on your knowledge from previous DFSMSHsm output of the sequence of dump volumes in the dump copy, use the FIXCDS command to correct the volume sequence field in the Y record for the volume, or the position of the volume in the dump copy's array of volumes in the G record for the generation.
*ERR 101 <i>volser</i> - <i>dumpvtoccopy</i> INDICATED ON <i>migvol</i>, BUT NO V RECORD		
A BCDS eligible record for volume <i>volser</i> exists, indicating that dump VTOC copy data set <i>dumpvtoccopy</i> should exist on migration volume <i>migvol</i> . No MCDS migration volume (V) record exists for <i>migvol</i> .	If the DUMPVTOC copy can be found on the volume, an ADDVOL command is generated for an L1 volume.	
*ERR 102 <i>volser</i> - NO P RECORD FOR <i>generation</i>		
For volume <i>volser</i> , there is a dump generation record (<i>generation</i>) that does not have a corresponding BCDS eligible volume (P) record.		If you need to restore the dumped volume from this generation, use the FIXCDS command to recreate the P record, such that it refers to the generation.
*ERR 103 <i>volser</i> - NO INDICATION OF <i>generation</i>		
For volume <i>volser</i> there is a BCDS eligible volume (P) record, but that record has no indication of generation <i>generation</i> , even though a dump generation (G) record exists for that generation.		If *ERR 94 occurs for the volume mentioned in *ERR 103, you should use the FIXCDS command to correct either the DGN record or the <i>generation</i> entry in the P record.
*ERR 104 <i>dumpvol</i> - NO DGN RECORD FOR <i>generation</i>		
The dump volume record for <i>dumpvol</i> indicates that the volume is part of a copy in dump generation <i>generation</i> , but no dump generation (G) record exists for the generation.		If you also see an *ERR 97 message referring to the volume, you should use the FIXCDS command to update the Y record for the volume to refer to the <i>generation</i> mentioned in the *ERR 97 message. If you also see an *ERR 94 message referring to this generation, it will contain the volume serial number, which make up the first 6 bytes of the DGN record key.
*ERR 105 <i>volser</i> - NO P RECORD		
For volume <i>volser</i> , there is no BCDS eligible volume (P) record.		The volume serial specified in the command may be spelled incorrectly.
*ERR 106 <i>cpname</i> - NO F (FRB) RECORD FOUND		
There is no copy pool F FRB record for <i>cpname</i> .		If <i>cpname</i> is " *NONE* ", there are no FRB records. This indicates that a fast replication backup was not created or failed. (That is, the FRBACKUP command has not been issued for the indicated copy pool or has failed.)
*ERR 110 <i>volser</i> - NO V OR X RECORD [DUMP]		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
For volume <i>volser</i> , there is no MCDS volume (V) record or BCDS backup volume (X) record. DUMP indicates that a dump volume (Y) record exists for <i>volser</i> .		<p>The volume serial specified in the command may be spelled incorrectly.</p> <p>If not, and the volume is identified as a different type of DFSMSHsm-owned or DFSMSHsm-managed volume, you cannot use this AUDIT option on the volume.</p> <p>If the volume serial is not misspelled and you have evidence that the volume was once known to DFSMSHsm, you need to check DFSMSHsm logs to determine why the volume was deleted.</p>
*ERR 111 <i>volser</i> - MISSING TTOC EXTENSION <i>nnnn</i>		
Volume <i>volser</i> is missing one or more TTOC extensions (<i>nnnn</i>). (Gaps exist in the sequence of extension numbers.)	The missing TTOC extension is created.	
*ERR 112 <i>volser</i> - EXTRANEIOUS TTOC EXTENSION <i>nnnn</i>		
For volume <i>volser</i> , TTOC extension <i>nnnn</i> is extraneous.	The extraneous TTOC extension is deleted.	
*ERR 113 <i>volser</i> TTOC EXT. <i>nnnn</i> - NO {A D} RECORD FOR <i>ttocdsn</i>		
<p>In TTOC extension <i>nnnn</i> for volume <i>volser</i>, the data set entry for <i>ttocdsn</i> is flagged to indicate a valid copy, but one of the following conditions exists:</p> <ul style="list-style-type: none"> No MCDS data set (D) record or MCDS alias entry (A) record exists for <i>ttocdsn</i>. An MCDS alias entry (A) record exists for <i>ttocdsn</i> but the MCDS data set (D) record referred to does not. 	<p>An MCDS data set (D) record is created for the TTOC entry.</p> <p>Note: If a (D) record was not created for data set <i>ttocdsn</i>, this tape begins with a partial data set. To have a (D) record properly created and built for this data set, you need to run AUDIT MEDIACONTROLS against the tape volume <i>volser</i>.</p>	
*ERR 114 <i>volser</i> - TTOC EXT. <i>nnnn</i> - <i>ttocdsn</i> HAS 'A' REC		
In TTOC extension <i>nnnn</i> for volume <i>volser</i> , the data set entry for <i>ttocdsn</i> does not indicate a valid copy, but an MCDS alias entry (A) record exists for <i>ttocdsn</i> .	<p>If there is no corresponding D record, the alias (A) record for the TTOC entry is deleted.</p> <p>If the corresponding D record is no longer valid and the catalog does <i>not</i> indicate the data set is migrated, the alias (A) record for the TTOC entry is deleted.</p> <p>If the corresponding D record is valid and indicates a migration copy at the current location, the TTOC entry is flagged valid.</p>	
*ERR 115 <i>volser</i> - TTOC EXT. <i>nnnn</i> - <i>ttocdsn</i> IS MIGRATION COPY		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
Volume <i>volser</i> with TTOC extension <i>nnnn</i> indicates that <i>ttocdsn</i> is a valid migration copy. The catalog indicates that the data set is migrated. The corresponding MCDS data set (D) record indicates a different migration volume from volume <i>volser</i> , but that volume (according to its TTOC, for a tape volume, or its VTOC, for a DASD volume) does not contain the migration copy.	The MCDS data set (D) record is updated to refer to this TTOC entry.	
*ERR 116 <i>volser</i> - TTOC EXT. <i>nnnn</i> - <i>ttocdsn</i> HAS INVALID C RECORD		
In TTOC extension <i>nnnn</i> for volume <i>volser</i> , the data set entry for <i>ttocdsn</i> does not indicate a valid copy, but a BCDS backup version (C) record exists for <i>ttocdsn</i> .		
*ERR 116 <i>volser</i> - TTOC EXT. <i>nnnn</i> - <i>ttocdsn</i> HAS INVALID C RECORD (continued)		
In TTOC extension <i>nnnn</i> for volume <i>volser</i> , the data set entry for <i>ttocdsn</i> is flagged as a valid copy, and a BCDS backup version (C) record exists for <i>ttocdsn</i> , but a corresponding BCDS data set record does not exist, or does not refer to that version.	All associated backup version records are deleted from DFSMSHsm's control dataset, and the TTOC entry is patched invalid.	
*ERR 117 <i>volser</i> - TTOC EXT. <i>nnnn</i> - NO P RECORD FOR <i>vtoccopy</i>		
*ERR 117 <i>volser</i> - TTOC EXT. <i>nnnn</i> - NO P RECORD FOR <i>vcatcopy</i>		
For volume <i>volser</i> with TTOC extension <i>nnnn</i> , no BCDS eligible volume (P) record exists for VTOC copy data set <i>vtoccopy</i> or <i>vcatcopy</i> .		The data set is probably an obsolete VTOC (or VCAT) copy data set created when the volume indicated in the qualifier ".Vvolser" was backed up. Because no P record now exists for that volume, or if you need the backup, you should issue the BACKVOL command for the volume to create a backup and to regenerate the P record.
*ERR 118 <i>volser</i> - TTOC EXT. <i>nnnn</i> - <i>vtoccopy</i> NOT IN P RECORD FOR <i>srcvol</i>		
*ERR 118 <i>volser</i> - TTOC EXT. <i>nnnn</i> - <i>vcatcopy</i> NOT IN P RECORD FOR <i>srcvol</i>		
For volume <i>volser</i> , TTOC extension <i>nnnn</i> indicates a VTOC copy (<i>vtoccopy</i>) or a VCAT copy (<i>vcatcopy</i>), but <i>vtoccopy</i> or <i>vcatcopy</i> is not referred to in the corresponding BCDS eligible volume (P) record for <i>srcvol</i> .	The current TTOC entry is patched invalid.	
*ERR 119 <i>volser</i> - NO PREDECESSOR TTOC FOR <i>prevol</i>		
The TTOC for volume <i>volser</i> indicates that the first data set (migration copy or backup version) on the tape volume is continued (spans) from volume <i>prevol</i> . However, no corresponding (migration or backup) TTOC base record exists for tape volume <i>prevol</i> .		
*ERR 120 <i>volser</i> - NOT A SUCCESSOR FROM TTOC FOR <i>prevol</i>		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
The TTOC for volume <i>volser</i> indicates that the first data set (migration copy or backup version) on the tape volume is continued (spans) from volume <i>prevol</i> . The TTOC, however, for <i>prevol</i> does not indicate volume <i>volser</i> as a successor.		
*ERR 121 <i>volser</i> - <i>ttocdsn</i> NOT SPANNED FROM TTOC FOR <i>prevol</i>		
The TTOC for volume <i>volser</i> indicates that the first data set (<i>ttocdsn</i>) on the tape volume is continued (spans) from the volume <i>prevol</i> , and the TTOC for <i>prevol</i> indicates <i>volser</i> as a successor volume; but the last TTOC entry for volume <i>prevol</i> has a different data set name from the first TTOC entry for the current volume <i>volser</i> .		
*ERR 122 <i>volser</i> - NO SUCCESSOR TTOC FOR <i>sucvol</i>		
The TTOC for volume <i>volser</i> indicates that the last data set on the tape volume continues on (spans to) volume <i>sucvol</i> . There is, however, no corresponding (migration or backup) TTOC base record for tape volume <i>sucvol</i> .		
*ERR 123 <i>volser</i> - NOT A PREDECESSOR TO TTOC FOR <i>sucvol</i>		
The TTOC for volume <i>volser</i> indicates that the last data set (migration copy or backup version) on the tape volume continues on (spans to) volume <i>sucvol</i> . The TTOC, however, for <i>sucvol</i> does not indicate volume <i>volser</i> as a predecessor.		
*ERR 124 <i>volser</i> - <i>ttocdsn</i> NOT SPANNED TO TTOC FOR <i>sucvol</i>		
The TTOC for volume <i>volser</i> indicates that the last data set (<i>ttocdsn</i>) on the tape volume continues on (spans to) volume <i>sucvol</i> . A TTOC record set exists for volume <i>sucvol</i> , and it indicates <i>volser</i> as a predecessor volume; but the first TTOC entry for volume <i>sucvol</i> has a different data set name from the last TTOC entry for the current volume <i>volser</i> .		
*ERR 125 <i>volser1</i> SPANS {TO FROM} <i>volser2</i>, BUT <i>ttcdsn</i> THAT SPANNED IS NO LONGER VALID		
<i>volser1</i> is the volume being audited. If TO is the insert, <i>volser2</i> is the successor volume and the last data set (<i>ttcdsn</i>) on <i>volser1</i> continues on (spans to) <i>volser2</i> . If FROM is the insert, <i>volser2</i> is the predecessor volume and the first data set on <i>volser1</i> spans from <i>volser2</i> . The TTOC for <i>volser1</i> indicates that <i>volser2</i> is also needed for <i>ttcdsn</i> , but this is incorrect since <i>ttcdsn</i> is no longer valid.	If a TTOC exists for <i>volser2</i> and it indicates a data set spans to or from <i>volser1</i> and the spanning data set names are the same on both TTOCs, the TTOC for <i>volser2</i> is corrected by patching the appropriate field, TTCSUCVL or TTCPREVL, to blanks. The TTOC for <i>volser1</i> is corrected by patching the appropriate field, TTCSUCVL or TTCPREVL, to blanks.	
*ERR 130 <i>volser</i> - NOT SINGLE-FILE-FORMAT 3480 TAPE VOLUME		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
For volume <i>volser</i> , either a MCDS volume record or a BCDS backup volume record exists, but does not indicate a cartridge type tape written in single-file format.		The MEDIACONTROLS option cannot be used with this volume, because the volume is not written in single-file format.
*ERR 131 <i>volser</i> - CANNOT BE MOUNTED.		
For volume <i>volser</i> , either a MCDS volume record or a BCDS backup volume record exists, but the volume cannot be allocated or mounted.		The volume cannot be mounted, or the operator has chosen not to mount the tape volume. Determine that the volume is accessible to your operator before reissuing this AUDIT command for this volume.
*ERR 132 <i>volser</i> - <i>userdsn</i> NO LONGER MIGRATED		
A migration copy of <i>userdsn</i> exists on migration volume <i>volser</i> , but no catalog entry exists for <i>userdsn</i> , or the catalog indicates that <i>userdsn</i> is on a level-0 volume.	<p>If there is no MCDS data set (D) record and the current TTOC entry is flagged valid, the TTOC entry is patched invalid, and an existing alias (A) record is deleted;</p> <p>If a data set (D) record refers to the current copy on this tape, the TTOC entry is patched valid until the problem can be resolved;</p> <p>If the D record refers to another copy and the current TTOC entry is flagged valid, the TTOC entry is patched invalid.</p>	
*ERR 133 <i>volser</i> - <i>userdsn</i> (MIGRATED) NEEDS VALID D RECORD		
For volume <i>volser</i> , the catalog indicates that user data set name <i>userdsn</i> is migrated (<i>volser</i> is MIGRAT), but the MCDS data set (D) record either does not exist or is not flagged as valid.	The current TTOC entry is patched valid. A valid MCDS data set (D) record and (if needed) an alias (A) record are created.	If the data set in question is a VSAM component, see ERR 169 for additional diagnostic data and handling. If Audit rebuilt the MCD record for a VSAM data set and used the data component name for the key, further action is required to change the key to the cluster name. See ERR 169 for details and instructions.
*ERR 134 <i>volser</i> - CANNOT IDENTIFY ANY DFSMSHsm DATA SET		
The MCDS volume (V) record indicates that volume <i>volser</i> is a migration volume, but there is no common data set descriptor (CDD) record found on the tape, thus no DFSMSHsm migration copy or backup version can be identified.	The first data set TTOC entry is patched invalid.	
*ERR 135 <i>volser</i> - TTOC ENTRY FOR <i>ttocdsn</i> - BLOCK COUNT SHOULD BE <i>nnnn</i>		
In volume <i>volser</i> , the block count indicated for TTOC entry <i>ttocdsn</i> does not agree with the actual block count. The TTOC block count should be <i>nnnn</i> .	The TTOC entry is patched with the correct block count.	
*ERR 136 <i>volser</i> - TTOC ENTRY FOR <i>ttocdsn</i> - SHOULD BE VALID		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
For volume <i>volser</i> , both the MCDS data set (D) record and the catalog indicate that the TTOC entry for <i>ttocdsn</i> should be flagged as valid, but it is flagged as invalid.	The TTOC entry is patched valid, and (if needed) an alias (A) record is created.	
*ERR 137 <i>volser</i> - TTOC BASE NEEDS VALID-BLOCK COUNT <i>nnnn</i>		
For volume <i>volser</i> , a block count discrepancy exists between the number of blocks of valid data and the count of valid blocks in the TTOC base record. The TTOC base record needs to be updated with the valid block count of <i>nnnn</i> .	The base TTOC is patched with the valid block count.	
*ERR 138 <i>volser</i> - TTOC BASE NEEDS TOTAL-BLOCK COUNT <i>nnnn</i>		
For volume <i>volser</i> , a block count discrepancy exists between the number of blocks of data and the count of blocks in the TTOC base record. The TTOC base record needs to be updated with the block count of <i>nnnn</i> .	The base TTOC is patched with the total block count.	
*ERR 140 <i>migvol</i> - <i>userdsn</i> HAS NO ALIAS RECORD		
*ERR 140 <i>migvol</i> - <i>cdddsn</i> HAS NO ALIAS RECORD		
For migration volume <i>migvol</i> , no MCDS alias entry (A) record exists for one of the following conditions: <ul style="list-style-type: none"> • Data set name <i>userdsn</i> in the VTOC has the form of a DFSMSHsm-generated migration copy name. • Data set name <i>cdddsn</i>, in what appears to be a common data set descriptor (CDD) record, indicates a migration copy. (The catalog entry for <i>cdddsn</i> indicates volume serial MIGRAT; an MCDS data set (D) record for <i>cdddsn</i> refers to an alias (A) record for the migration copy.) 	An alias (A) record is created, referring to this migration copy.	
*ERR 141 <i>migvol</i> - SCRATCHED EMPTY DATA SET {<i>migdsn</i> <i>version</i>}		
For migration volume <i>migvol</i> , migration copy <i>migdsn</i> or backup version <i>version</i> is empty.	The empty data set is scratched and uncataloged.	
*ERR 142 <i>migvol</i> - <i>version</i> HAS NO C RECORD		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
<p>For migration volume <i>migvol</i>, one of the following conditions exists for the data set named <i>version</i> in the VTOC:</p> <ul style="list-style-type: none"> The name has the form of a DFSMSHsm-generated backup version name, but no corresponding BCDS backup version (C) record exists. The first record of data set <i>version</i> appears to be a common data set descriptor (CDD) record, but the index of versions in the B record does not refer to <i>version</i>. 	The data set is scratched.	
*ERR 143 <i>migvol</i> - NO P RECORD FOR <i>srcvol</i> OF <i>vtoccopy</i>		
*ERR 143 <i>migvol</i> - NO P RECORD FOR <i>srcvol</i> OF <i>vcacopy</i>		
Migration volume <i>migvol</i> contains either the VTOC copy <i>vtoccopy</i> or the VCAT copy <i>vcacopy</i> made during a backup of volume <i>srcvol</i> . However, no BCDS eligible volume (P) record exists for source volume <i>srcvol</i> .	The copy data set is scratched and uncataloged.	
*ERR 144 <i>migvol</i> - <i>vtoccopy</i> NOT IN P RECORD FOR <i>srcvol</i>		
*ERR 144 <i>migvol</i> - <i>vcacopy</i> NOT IN P RECORD FOR <i>srcvol</i>		
Migration volume <i>migvol</i> contains either the VTOC copy <i>vtoccopy</i> or the VCAT copy <i>vcacopy</i> made during a backup of volume <i>srcvol</i> . The BCDS eligible volume (P) record does not refer to either <i>vtoccopy</i> or <i>vcacopy</i> .	The copy data set is scratched and uncataloged.	
*ERR 145 <i>migvol</i> - NO P RECORD FOR <i>srcvol</i> OF <i>dumpvtoccopy</i>		
Migration volume <i>migvol</i> contains the dump VTOC copy <i>dumpvtoccopy</i> made during a volume dump of <i>srcvol</i> . However, no BCDS eligible volume (P) record exists for source volume <i>srcvol</i> .	The copy data set is scratched and uncataloged.	
*ERR 146 <i>migvol</i> - <i>dumpvtoccopy</i> NOT IN ANY DGN RECORD FOR <i>srcvol</i>		
Migration volume <i>migvol</i> contains the dump VTOC copy <i>dumpvtoccopy</i> made during a volume dump of <i>srcvol</i> . However, no dump generation record (DGN) exists that refers to <i>dumpvtoccopy</i> .	The copy data set is scratched and uncataloged.	
*ERR 147 <i>migvol</i> - <i>userdsn</i> NEEDS D RECORD		
<p>User data set <i>userdsn</i> is on migration volume <i>migvol</i>. The catalog indicates that this data set is migrated. The MCDS data set (D) record for <i>userdsn</i> is either missing or indicates some other migration volume, but the VTOC or TTOC for that other migration volume does not refer to <i>userdsn</i>.</p> <p>The <i>userdsn</i> is not recognized as a DFSMSHsm-generated name.</p>	The D record is created, or patched, to refer to the copy on the migration volume being audited.	

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
*ERR 148 migvol - userdsn MAY BE OLD MIGRATION COPY		
An MCD record was found for the migrated data set on the volume, or in the SDSP data set being processed. The catalog indicates that this data set is migrated. A discrepancy was found between the MCD record and the volume being audited.		<p>For AUDIT MEDIACONTROLS(SDSP) VOLUME(volser):</p> <p>If the MCD record indicates this data set is on tape, the AUDIT output will recommend AUDIT MEDIACONTROLS VOLUME(<i>tapevolser</i>).</p> <p>If the data set is determined to be properly migrated to tape and you want the SDSP entry removed to prevent further ERR148s, then</p> <ol style="list-style-type: none"> 1. Recall the migrated copy from the tape volume. 2. Create an MCD record for the data set in the SDSP as follows: <pre>FIXCDS D userdsn ADDMIG(sdspvol)</pre> <p>Note: If the MCD create fails due to an already existing MCD record, delete the existing MCD record and create a new MCD record for the SDSP data set as suggested. The command to delete the MCD record is:</p> <pre>FIXCDS D userdsn DELETE</pre> <ol style="list-style-type: none"> 3. Delete the MCD entry from the MCDS and remove the entry from the SDSP by issuing the following command: <pre>HSEND DELETE userdsn</pre> <p>The MCD record indicates it is not in an SDSP data set, and the issuance of an OBTAIN for the data set on the volume indicated in the MCD record resulted in "volume not mounted".</p> <p>The MCD record indicates an SDSP data set, but the volume indicated in the MCD record does not match the volume being audited. The volume indicated in the MCD record cannot be checked because we could not allocate the SDSP data set on that volume, or the volume was not mounted.</p>
*ERR 148 migvol - userdsn MAY BE OLD MIGRATION COPY (continued)		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
An MCD record was found for the migrated data set on the volume, or in the SDSP data set being processed. The catalog indicates that this data set is migrated. A discrepancy was found between the MCD record and the volume being audited.		<p>For AUDIT DIRECTORYCONTROLS VOLUMES(<i>dasdvol</i>), the MCD record indicates the migration copy was created before HSM Release 3, and</p> <ul style="list-style-type: none"> The MCD record indicates the data set is in an SDSP data set. However, AUDIT found the migrated data set on the DASD volume being audited. AUDIT will recommend running AUDIT MEDIACONTROLS(SDSP) VOLUMES(<i>sdspvol</i>) to assist in diagnosing the error. The MCD record says it is on a DASD volume other than the one currently being audited. AUDIT tried to OBTAIN the data set from that other volume, but the volume was not mounted. AUDIT will recommend AUDIT MEDIACONTROLS VOLUMES(<i>volser</i>) of the volume indicated in the MCD record.
*ERR 149 <i>migvol</i> - <i>migdsn</i> IS OLD MIGRATION COPY		
<p>User data set <i>migdsn</i> is on migration volume <i>migvol</i>. The catalog indicates that <i>migdsn</i> is migrated. The MCDS data set (D) record for <i>migdsn</i> refers to a migration copy on a different volume. The VTOC or TTOC for that other migration volume indicates that a migration copy of <i>migdsn</i> does exist on that volume.</p> <p>The <i>migdsn</i> is not recognized as a DFSMSHsm-generated name.</p>	The data set is scratched. If an alias (A) record exists, it is deleted.	
*ERR 150 <i>migvol</i> - <i>version</i> HAS NO CDD RECORD		
On migration volume <i>migvol</i> , the first record in (what appears to be, by its name) migration copy <i>migdsn</i> or in (what appears to be, by its name) backup version <i>version</i> , cannot be identified as a common data set descriptor (CDD) record.		<p>The data set, if named like a DFSMSHsm backup version, may nevertheless not be a DFSMSHsm data set. If it is not, and because the data set is on a migration volume, not a backup volume, there should be no problem.</p> <p>If the data set is a backup version, it is not possible to recover it without a CDD as the first record, and there is no point in having DFSMSHsm move the version to a backup volume.</p> <ul style="list-style-type: none"> If no C record exists, the version can be scratched from the volume. If a C record exists for the version, issue the BDELETE command to delete the version.
*ERR 151 <i>migvol</i> - <i>cdddsn</i> NOT CATALOGED, HAS COPY <i>migdsn</i>		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
On migration volume <i>migvol</i> , the first record in (what appears to be, by its name) migration copy <i>migdsn</i> appears to be a common data set descriptor (CDD) record created from data set <i>cdddsn</i> . There is, however, no catalog entry for <i>cdddsn</i> .	The MCDS data set (D) record is patched as invalid, and patched to indicate that the migration copy is to be scratched. The record will automatically be cleaned up the next time DFSMSHsm migration runs (secondary space management processing).	
*ERR 152 <i>migvol</i> - <i>cdddsn</i> NOT MIGRATED, HAS COPY <i>migdsn</i>		
On migration volume <i>migvol</i> , the first record in (what appears to be, by its name) migration copy <i>migdsn</i> appears to be a Common Data set Descriptor (CDD) record created from data set <i>cdddsn</i> . The catalog entry, however, for <i>cdddsn</i> indicates that the data set is not migrated.	The MCDS data set (D) record is patched as invalid, and patched to indicate that the migration copy is to be scratched. The record will automatically be cleaned up the next time DFSMSHsm migration runs (secondary space management processing).	
*ERR 160 <i>migvol</i> - DOES NOT HAVE SDSP DATA SET		
For migration volume <i>migvol</i> , no SDSP data set listed in the VTOC exists. The name of such a data set must be <i>uid.SMALLDS.Vmigvol</i> , where <i>uid</i> is the authorized userid for the current startup of DFSMSHsm.		The volume serial specified in the command may be spelled incorrectly. If not, your installation may have changed the UID value in the DFSMSHsm startup procedure since the SDSP data set on the volume was created. You may need to rename the data set, changing the first qualifier to the current UID value.
*ERR 161 <i>migvol</i> - <i>userdsn</i> NOT CATALOGED, HAS {VALID INVALID} COPY IN SDSP		
The small data set packing (SDSP) data set on migration volume <i>migvol</i> contains a migration copy of <i>userdsn</i> , but that data set is not cataloged. The message text indicates whether the data set appears to be valid or invalid.	<p>If there is no data set (D) record, and no catalog entry, the migration copy is deleted (internally) from the SDSP.</p> <p>If there is a D record, but no catalog entry, and the copy has gaps in its records, the migration copy is deleted (internally) from the SDSP, and the D record is patched invalid.</p> <p>If the D record identifies the migration copy as valid (that is, recalled) and the data set is not cataloged, and the D record does not indicate that the migration copy needs to be scratched, a FIXCDS PATCH command is generated by AUDIT to turn on the “needs scratch” indicator.</p>	<p>If there is a D record, but no catalog entry, and the D record appears to represent a valid migration copy, you should determine one of the following:</p> <ul style="list-style-type: none"> • If the data set is invalid and you want it deleted, you should issue an HSEND DELETE command, to have associated records deleted and have the data set removed from the SDSP. Following is an example of the HSEND DELETE command: <div style="background-color: #f0f0f0; padding: 10px; margin: 10px 0;"> <pre>HSEND DELETE userdsn</pre> </div> • If the data set is valid, you should catalog it to a volume serial of ‘MIGRAT’.
*ERR 162 <i>migvol</i> - <i>userdsn</i> NOT MIGRATED, HAS COPY IN SDSP		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
The small data set packing (SDSP) data set on migration volume <i>migvol</i> contains a migration copy of <i>userdsn</i> , but the catalog entry for that data set indicates that the data set is not migrated.	<p>If there is no data set (D) record, the migration copy is deleted (internally) from the SDSP.</p> <p>If there is a D record, but the catalog does not indicate MIGRAT, and the copy has gaps in its records, the migration copy is deleted (internally) from the SDSP, and the D record is patched invalid.</p> <p>If there is a D record, but the catalog does not indicate MIGRAT, and there are no record gaps in the copy, and the D record is flagged other than “invalid, needs scratch,” the D record is patched as invalid, and patched to indicate that the migration copy be scratched.</p>	
*ERR 163 migvol - userdsn MISSING ONE OR MORE RECORDS IN SDSP		
The small data set packing (SDSP) data set on migration volume <i>migvol</i> contains a migration copy of <i>userdsn</i> , but the migration copy is missing one or more records.		<p>If the data set is no longer needed, you can issue a DELETE command for the data set.</p> <p>If you can recall the data set successfully, then the record gap is not significant.</p> <p>If the data set cannot be recalled successfully, but a backup copy exists, then delete the data set (to remove the catalog entry and D record), and issue the RECOVER command for the data set.</p>
*ERR 164 TTOC FOR VOLUME <i>volser</i> INDICATES IN-STORAGE RECORDS EXIST		
The TTOC for the tape volume indicates that in storage records existed that may not have been written to the OCDS on DASD. After this AUDIT has completed, all data sets on the tape will have entries in the existing TTOC record for this volume.	The TTCFAIST bit in the base TTOC is patched off.	
*ERR 165 NO DATA FILES ARE INDICATED FOR ABR RECORD KEY = <i>abrrecordkey</i>		
The ABR record with key <i>abrrecordkey</i> indicates that neither a DFSMSdss data file nor an internal data file is included with the aggregate backup.		The ABR record specified is invalid and should be deleted.
*ERR 166 ABARSCONTROLS SPECIFIED USING AGGREGATE GROUP = <i>aggrouppname</i>. A MATCHING AGGREGATE GROUP NAME COULD NOT BE FOUND IN THE BCDS.		
The aggregate group name specified with the AUDIT command has no ABR records (has not been backed up).		Aggregate group name <i>agname</i> was probably spelled incorrectly.
*ERR 167 ABARSCONTROLS SPECIFIED, BUT NO ABR RECORDS WERE FOUND IN THE BCDS.		
The BCDS has no record of aggregate backup activity.		No ABR records exist, therefore no aggregate backups have successfully completed.
*ERR 168 [CONTROL DFSMSdss DATA INTERNAL DATA INSTRUCTION/ACTIVITY LOG] FILE NAME = <i>filename</i>, INDICATED IN THE ABR RECORD, BUT IS NOT CATALOGED. ABR RECORD KEY = <i>abrrecordkey</i>		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
The aggregate group backup refers to a particular file, but the file cannot be found in the catalog.		You should catalog the file indicated by <i>filename</i> .
*ERR 169 MCD RECORD CREATED FOR VSAM DATA SET <i>dsname</i> NEEDS NEW KEY		
In diagnosing error 133 for a migrated VSAM data set, AUDIT cannot determine its base cluster name and has used data component name <i>dsname</i> to recreate the MCDS data set record.		<p>You need to determine the base cluster name of the VSAM data set, and issue a FIXCDS command to change the key of the MCD record:</p> <pre>FIXCDS D <i>dsname</i> NEWKEY(<i>base cluster_name</i>)</pre> <p>Then, if AUDIT re-created an alias (A) record, you can either</p> <ul style="list-style-type: none"> recall and remigrate the VSAM data set; or issue a FIXCDS command to patch the alias record: <p>where <i>generated_name</i> can be found in the AUDIT output that created the alias record, or at offset X'9C' (field MCDMCANM) in the MCD record.</p> <pre>FIXCDS A <i>generated_name</i> PATCH(X'4'<i>base cluster_name</i>)</pre>
*ERR 170 SPECIFIED COPY POOL <i>cpname</i> DOES NOT EXIST		
*ERR 171 F RECORD INDICATES <i>x</i> {DUMP TOTAL DASD SMS SPECIFIED} VERSIONS, ONLY <i>y</i> VERSIONS WERE FOUND		
Number of (FRD) Dump Version records found is not equal to the number of dumped versions indicated in the (FRB) copy pool record.		<ul style="list-style-type: none"> If DUMP, the number of (FRD) Dump Version records found is not equal to the number of dumped versions indicated in the (FRB) copy pool record. If TOTAL, the number of versions found is not equal to the number of versions indicated in the FRB record. If DASD, the number of all DASD versions found is not equal to the number indicated in the FRB record. If SMS SPECIFIED, the number of non-rolloff versions found is not equal to the number indicated in the FRB record.
*ERR 172 H RECORD INDICATES <i>x</i> DASD VERSIONS, ONLY <i>y</i> VERSIONS WERE FOUND		
*ERR 173 COPY POOL <i>cpname</i> F RECORD INDICATES <i>x</i> H RECORDS FOR VERSION <i>z</i>, ONLY <i>y</i> RECORDS WERE FOUND		
(FRB) Copy pool record field and the number of (FRVP) volume pair records found do not match.		This error may be resolved by other audit discoveries. Some audit actions delete FRVP records which may result in these counts be correct.
*ERR 174 COPY POOL <i>cpname</i> F RECORD WAS NOT FOUND, ORPHANED H RECORDS EXIST		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
A volume pair record has been found, but the copy pool record to which it belongs does not exist.	The volume pair record is deleted. Note: Repair actions are only performed when FIX is specified on the audit command.	Error only found when no <i>cpname</i> was specified on the AUDIT COPYPOOLCONTROLS command.
*ERR 175 H (FRVP) RECORD INDICATES IT IS A 'PREPARE' RECORD, THE CORRESPONDING F (FRB) RECORD DOES NOT		
		<ul style="list-style-type: none"> You need to determine which indicator is correct. Use the FIXCDS PATCH command to correct the problem. You can use the LIST COPYPOOL command to help you determine which indicator is correct. Either the 'Prepare' flag should be on in the F (FRB) record or the FRVP records are orphan entries and should be deleted using the FIXCDS DELETE command.
*ERR 176 H RECORD FOR VERSION <i>x</i> EXISTS, BUT VERSION CANNOT BE FOUND IN CORRESPONDING F RECORD FOR COPY POOL <i>cpname</i>		
The copy pool record for this volume pair record does not contain a corresponding version entry.	The volume pair record is deleted. Note: Repair actions are only performed when FIX is specified on the audit command.	This volume pair record is not needed. This error is only detected when no <i>cpname</i> is specified on the AUDIT COPYPOOLCONTROLS command.
*ERR 177 I (FRTV) RECORD WAS EXPECTED FOR VOLUME <i>volser</i> AND WAS NOT FOUND		
A F (FRB) record indicates this is a valid version. The volume represented by this FRTV record needs to be made available and an FRTV record should be created to represent it or the fast replication backup of this copy pool will fail.		The FIXCDS CREATE command can be used to recreate the FRTV record.
*ERR 178 I (FRTV) RECORD FOR TARGET VOLUME <i>targetvolser</i>, EXPECTS FRSV (J) RECORD FOR SOURCE VOLUME <i>sourcevolser</i>, WHICH WAS NOT FOUND		
		<p>AUDIT has detected that the FRSV (J) record for source volume <i>sourcevolser</i> does not exist. This record should exist unless this version is a PREPARED version.</p> <p>Note: The ERR178 may be incorrectly issued when the copy pool name is not specified on the AUDIT command. Use the LIST COPYPOOL(<i>cpname</i>) ALLVOLS(ALLVERS) command to identify which copy pool these target volumes belong to. Then reissue the AUDIT COPYPOOLCONTROLS(<i>cpname</i>) command, specifying the copy pool name the <i>sourcevolser</i> and <i>targetvolser</i> belong to.</p> <p>If the ERR178 message is persistent, record errors exist.</p>

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
*ERR 179 <i>cpname</i> VERSION <i>versionnumber</i>, H RECORD HAS WRONG TARGET VOLUME <i>targetvolser</i>, OR I RECORD HAS WRONG SOURCE VOLUME <i>sourcevolser</i>		
		<p>Determine which record is incorrect and correct the record using the FIXCDS PATCH command.</p> <p>Then use the FRDELETE command to delete the copy pool version (<i>cpname</i>) identified in the message.</p> <p>Example: FRDELETE COPYPOOL(<i>cpname</i>) ALL BOTH</p>
*ERR 180 J RECORD NOT FOUND FOR VOLUME <i>volser</i>, IT BELONGS TO COPY POOL <i>cpname</i>, VER= <i>vernum</i>		
A volume pairs record was found indicating source volume <i>volser</i> , but the source volume record does not exist.		This error will be issued if the source volume has been removed from the copy pool, and a successful FRBACKUP has taken place since its removal.
*ERR 181 F (FRB) RECORD FOR <i>cpname</i> NOT FOUND, POSSIBLY ORPHANED J (FRSV) RECORD FOR VOLUME <i>volser</i>		
		<ul style="list-style-type: none"> Remove <i>cpname</i> from the FRSV <i>cpname</i> array. If there are no <i>cpnames</i> left in the array, delete the FRSV record for VOLUME <i>volser</i>.
*ERR 182 NO P (MCP) RECORD FOUND FOR VOLUME <i>volser</i>, IS THIS AN ERROR?		
*ERR 183 VOLUME <i>volser</i> FROM THE J (FRSV) RECORD L0 FIELD COULD NOT BE FOUND IN A CORRESPONDING H (FRVP) RECORD		
		H (FRVP) record was not found or <i>volser</i> was not found in the corresponding H (FRVP) records.
*ERR 184 ORPHANED J RECORD DETECTED FOR VOLUME <i>srcvol</i>		
No <i>cpname</i> copy pool record (FRB) was found for any cypool this source volume belongs to. A source volume can belong to many copy pools.	<p>The FRSV source volume record is deleted.</p> <p>Note: Repair actions are only performed when FIX is specified on the audit command.</p>	Previously issued ERR181 message(s) may be help to identify the copy pool name this record was known to have belonged to. This error is only detected when no <i>cpname</i> is specified on the AUDIT COPYPOOLCONTROLS command.
*ERR 185 COPY POOL <i>cpname</i> COULD NOT BE FOUND IN THE J (FRSV) RECORD COPY POOL ARRAY FOR VOLUME <i>volser</i>		
		You need to use the FIXCDS PATCH command to patch the FRSV record with the <i>cpname</i> , or if the FRSV record contains no other copy pool names, deleted it with the FIXCDS DELETE command.
*ERR 186 ORPHANED K RECORD FOUND, COPY POOL <i>cpname</i> NO LONGER EXISTS		
The copy pool record for this copy pool dump record does not exist. This record is no longer needed.	<p>The FRD copy pool dump record is deleted.</p> <p>Note: Repair actions are only performed when FIX is specified on the audit command.</p>	This error is only detected when no <i>cpname</i> is specified on the AUDIT COPYPOOLCONTROLS command.

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
*ERR 187 K RECORD EXISTS, COPY POOL <i>cpname</i> F RECORD INDICATES NO DUMP COPY		
The FRB copy pool record indicates there is no dump copy for this copy pool version. An FRD copy pool dump version record exists for this version.	The corresponding version entry in the FRB copy pool record is patched to indicate the dump version exists. Note: Repair actions are only performed when FIX is specified on the audit command.	This error is only detected when no <i>cpname</i> is specified on the AUDIT COPYPOOLCONTROLS command.
*ERR 188 K (FRD) RECORD NOT FOUND, F (FRB) RECORD FOR COPY POOL <i>cpname</i> INDICATES THAT IT DOES EXIST		
The FRB record indicates a dump for this copy pool exists, but the record representing the dump cannot be found.		
*ERR 189 H (FRVP) RECORD NOT FOUND COPY POOL <i>cpname</i>		
During the AUDIT VOLUMECONTROLS (RECOVERABLE), an error was detected.		You should consider running AUDIT COPYPOOLCONTROLS (<i>cpname</i>) to learn more regarding this discrepancy.
*ERR 190 VOLUME <i>volser</i> COULD NOT BE FOUND IN ANY ASSOCIATED H (FRVP) RECORD FOR COPY POOL <i>cpname</i>		
		You should consider running AUDIT COPYPOOLCONTROLS (<i>cpname</i>) to learn more regarding this discrepancy.
*ERR 191 P RECORD FOUND FOR VOLUME <i>volser</i>, IS NOT INDICATED AS A COPY POOL VOLUME		
*ERR 192 F RECORD FOUND FOR COPYPOOL <i>cpname</i>, WHICH VOLUME <i>volser</i> BELONGS TO		
If detected during AUDIT VOLCTL(RECOVERABLE), this is detected and flagged so that you can confirm that the source volume described is recoverable.	If detected during AUDIT VOLCTL(RECOVERABLE) FIX, the fix is to build the proper AUDIT COPYPOOLCONTROLS command and execute to confirm that the volume referenced is in a recoverable state. AUDIT does not issue any patch commands to correct any records. Its intent is to allow you the opportunity to confirm that this volume is a valid recoverable volume.	Audit the specified copy pool to determine the recoverability of this volume.
*ERR 193 J RECORD NOT FOUND FOR VOLUME <i>volser</i>		
		<ul style="list-style-type: none"> The MCP or DGN record for volume <i>volser</i> indicates that an FRSV record should exist for this volume. Expect this error to be issued for each dumped version of this volume. The MCPF_COPYPOOL bit should be patched off if it is determined that this is no longer a cpool volume, as is indicated by the missing FRSV record. FIXCDS P <i>volser</i> PATCH('9' BITS(.....0))
*ERR 194 CATALOG INFORMATION DATA SET <i>dsname</i> FOR COPY POOL <i>cpname</i>, VERSION <i>version</i> WAS NOT FOUND		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
The copy pool record indicates that the catalog information was captured but a data set containing this information was not found by AUDIT.		Remove the indication of catalog information from the FRB and reset the number sequence field to zero.
*ERR 195 COPY POOL <i>cpname</i>, VERSION <i>version</i>, INDICATES CATALOG INFORMATION DATA SET <i>dsname</i> BUT THE DATA SET CORRESPONDS TO A DIFFERENT COPY POOL VERSION		
The copy pool record indicates that the catalog information was captured but the corresponding catalog information data set indicates that it contains catalog information for a different copy pool backup version.		Run AUDIT DIRECTORYCONTROLS against the ML1 volume where the catalog information data set resides. If necessary, remove the indication of catalog information from the FRB and reset the number sequence field.
*ERR 196 CATALOG INFORMATION DATA SET <i>dsname</i>, IS NOT CATALOGED		
The catalog information data set was found on an ML1 volume but the data set is not cataloged.		Catalog the data set.
*ERR 197 CATALOG INFORMATION DATA SET <i>dsn</i> INDICATES COPY POOL <i>cpname</i> BUT A CORRESPONDING BACKUP WAS NOT FOUND		
There is no corresponding Fast Replication backup version for the catalog information data set.		Determine if the catalog information data set corresponds to a valid fast replication backup version. If it does, patch the FRB record for the fast replication backup version to indicate that the catalog information data set exists. If it does not, delete the catalog information data set.
*ERR 198 CATALOG INFORMATION DATA SET <i>dsname</i>, CORRESPONDS TO COPY POOL <i>cpname</i> , VERSION <i>version</i> BUT THE COPY POOL BACKUP VERSION DOES NOT INDICATE CATALOG INFORMATION WAS CAPTURED		
The copy pool backup version specified by the catalog information data set indicates that the catalog information was not captured. This catalog information data set does not correspond to a valid existing copy pool backup version.		Delete the catalog information data set.
*ERR 199 CATALOG INFORMATION DATA SET <i>dsname</i>, CORRESPONDS TO COPY POOL <i>cpname</i> , VERSION <i>version</i> BUT THE COPY POOL BACKUP VERSION INDICATES A DIFFERENT CATALOG INFORMATION DATA SET		
The copy pool version that the catalog information data set indicates that it is associated to is not the catalog information data set that the copy pool version record indicates.		Update the entry within the FRB record which corresponds to this copy pool backup version to indicate that the catalog information is available and to reflect the correct number sequence of the data set.
*ERR 200 ERROR ENCOUNTERED WHEN PROCESSING CATALOG INFORMATION DATA SET <i>dsn</i> , FOR COPY POOL <i>cpname</i> VERSION <i>vers</i>		
There was an unexpected error while trying to process the catalog information data set. This message should be preceded by an ARC1812I message.		Refer to the ARC1812I message.
*ERR 201 CATALOG INFORMATION DATA SET <i>dsn</i>, IS CATALOGED TO A DIFFERENT VOLUME		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
The catalog locate for the catalog information data set indicates that this data set is cataloged on a different volume.		Determine if the catalog is inaccurate or if this is a duplicate uncataloged version of the catalog information data set. Scratch the uncataloged copy if appropriate. If the uncataloged copy is valid, determine if the cataloged copy should be deleted and the uncataloged copy should be cataloged.
*ERR 202 ORPHAN I (FRTV) RECORD FOUND FOR TARGET VOLUME <i>tgtvol</i>, SOURCE VOLUME <i>srcvol</i>		
The source volume record was not found for this target volume record and the target volume record was not validated by the volume pair record validation process. This record is no longer needed.	The target volume record is deleted. Note: Repair actions are only performed when FIX is specified on the audit command.	This error is only detected when no <i>cpname</i> is specified on the AUDIT COPYPOOLCONTROLS command.
*ERR 203 VALID MCD FOR DATA SET <i>dsn</i> HAS INCORRECT VOLSER, SHOULD BE VOL= <i>volser</i>		
During AUDIT MEDIACONTROLS, both a valid CDD record and corresponding valid MCD record have been identified. The MCD record does not indicate that it is for the tape volume being audited. Since the file block id and the timestamp in these two records are identical, this MCD record must belong to this migrated data set.	The migrated data set record (D) is patched with the volsr of the tape that is being audited.	
*ERR 204 I (FRTV) RECORD FOR VOLUME <i>targetvolser</i> HAS MULTIPLE H (FRVP) RECORD ENTRY REFERENCES		
Multiple FRVP record entry references were found for the target volume <i>tgtvolser</i> . A target volume can have a single valid FRVP record entry reference only. The target volume is "overused."		*ERR 179 messages are issue to assist determining the copy pools and versions that are using the target volume. The existing copy pool backups identified during auditing are not reliable backups and should be deleted using the FRDELETE command. You should then issue the FRBACKUP command to create a valid backup of each copy pool.
*ERR 205 K RECORD EXISTS, COPY POOL <i>cpname</i> F RECORD DOES NOT HAVE THIS VERSION		
A dump version record was found that has no version entry in the associated FRB record for the copy pool.	The copy pool dump version record is deleted. Note: Repair actions are only performed when FIX is specified on the audit command.	This error is only detected when no <i>cpname</i> is specified on the AUDIT COPYPOOLCONTROLS command.
*ERR 206 MCD INDICATES <i>dsn</i> IS IN CLOUD <i>cloud_network_connection_name</i> BUT NO CONTAINER <i>conname</i> OR OBJECTS WITH THE PREFIX <i>prefix</i> EXISTS		
MCD indicates that the data set is migrated to cloud but no container or no objects with the prefix that is stored in MCD exists.		If the data set is invalid and you want it deleted, you should issue an HSEND DELETE command, to have associated records deleted.
*ERR 207 <i>dsn</i> NOT CATALOGED, HAS {VALID INVALID} COPY IN CLOUD IN CONTAINER <i>container</i>		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
The data set has a migration copy of user <i>dsn</i> in cloud, but that data set is not cataloged. The message text indicates whether the data set appears to be valid or invalid.	<p>If there is no data set D record, and no catalog entry, the migration copy(object) is deleted from cloud.</p> <p>If the D record identifies the data set is recalled and the data set is not cataloged, and the D record does not indicate that the migration copy needs to be scratched, a FIXCDS PATCH command is generated by AUDIT to turn on the "needs scratch" indicator.</p>	<p>If there is a D record, but no catalog entry, and the D record appears to represent a valid migration copy, then determine one of the following:</p> <p>If data set is invalid and you want to delete it, you can issue FIXCDS D <i>dsn</i> DELETE to delete MCD record and run AUDIT MEDIACONTROLS(CLOUD(<i>cloud_network_connection_name</i>)) command with FIX option to remove migration copy from cloud.</p> <p>DFSMSdss can be used to restore the data set directly.</p>
*ERR 208 <i>dsn</i> NOT MIGRATED, HAS COPY IN CLOUD IN CONTAINER container		
The data set has a migration copy of user <i>dsn</i> in cloud, but catalog indicates that the data set is not migrated.	<p>If there is no data set D record and data set not cataloged as MIGRAT (recalled), the migration copy is deleted from cloud.</p> <p>If there is a D record, but the catalog does not indicate MIGRAT, and the D record is flagged other than "needs scratch," the D record is patched to indicate that the migration copy be scratched</p>	DFSMSdss can be used to restore the data set directly.
*ERR 209 NUMBER OF OBJECTS FOR <i>dsn</i> IN CLOUD IS NOT EQUAL TO THE VALUE THAT IS STORED IN MCD		
Real number of objects for data set <i>dsn</i> migrated to cloud is not equal to the number of objects value that is stored in MCD. Migration copy can be invalid.		Contact IBM support.
*ERR 210 CDD IS NOT FOUND FOR PREFIX <i>prefix</i> IN CONTAINER container		
Objects were found for pseudo-directory <i>prefix</i> but an APPMETA object could not be found to further examine the migration copy.		<p>Use the LIST CLOUD command to list what objects exist for that prefix. It may be possible to use the DFSMSdss RESTORE command to recover the data set. If FIX specified, the orphan objects are deleted and MCDS may retained. To delete the MCDS records follow the below steps.</p> <p>a. If Catalog entry indicates it is migrated, issue HSEND DELETE <i>usrdsn</i> to have associated records deleted.</p> <p>b. If no catalog entry or cataloged to LO, issue an AUDIT DATASETCONTROLS(MIGRATION) command to have associated records deleted.</p> <p>Following is an example of the HSEND AUDIT command: HSEND AUDIT DATASET CONTROLS(MIGRATION)</p> <p>HSEND DELETE <i>usrdsn</i></p>

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
*ERR 211 prefix IS OLD MIGRATION COPY OF dsname IN CONTAINER containername		
A migration copy for <i>dsname</i> was found in CONTAINER container, but it is not the most recent version.	The migration copy is deleted.	You can use the DFSMSdss RESTORE command to recover the old migration copy.
*ERR 212 file HAS UFN RECORD BUT DOES NOT HAVE B RECORD		
A UFN record was found for file <i>file</i> , but MCB record does not exist.		
*ERR 213 file MISSING UFN RECORD		
UFN record is not found for file <i>file</i> or file path is incorrect. <i>file</i> contains the file path or hash value. If file contains hexadecimal hash value then this is the only information that is available for the file.		
*ERR 214 file – MISSING UFN RECORD FOR [(PARENT CHILD SIBLING LAST CHILD)]		
UFN record contains valid hash value for parent, child, sibling or last child but MCB record is missing.		Maybe not all requested MCB's were processed if AUDIT FILCONTROLS(BACKUP) FILELEVEL() is issued. Use AUDIT FILCONTROLS(BACKUP) for full processing.
*ERR 215 THE DGN RECORD DGNkey IS FOR A CLOUD DUMP BUT CONTAINS INVALID VALUES		
Invalid values might include: <ol style="list-style-type: none"> 1. Cloud dump flag, DGNF_CLOUD is 0. 2. DGNCOPY# is greater than maximum number of allowed in a dump generation. 3. DGNNVSN is not 0. 4. DGN_CLOUD_NAME_LEN and DGN_CLOUD_NAME are inconsistent. 5. DGN TCT compression ratio is invalid. 6. Dump class name in DGNDCL is incorrect. 7. Container name in DGN record is incorrect. 8. More than one DGN cloud provider type indicators are set to 1. For example, both DGNF_CLD_CDA_TCT and DGNF_CLD_CDA_DIRECT flags are set to 1 		Use the FIXCDS G <i>dgnkey</i> DISPLAY <i>offset</i> command to identify invalid values, and then use FIXCDS G <i>dgnkey</i> PATCH <i>offset</i> command to change the fields with expected values. Or, contact IBM support.
*ERR 216 DGN IS NOT FOUND FOR PREFIX prefix IN CONTAINER container		
The corresponding DGN record is not found for the cloud dump prefix.	The orphaned cloud dump copy is deleted. Note: Repair actions are only performed when FIX is specified on the AUDIT command.	Determine if the orphaned dump copy is a result of an error during automatic dump expiration or deletion. Specify FIX on the AUDIT command to delete the orphaned cloud dump copy. Or, contact IBM support.
*ERR 217 THE NUMBER OF OBJECTS WITH PREFIX prefix IN THE CONTAINER container IS NOT CORRECT		

Table 76. Error codes used in AUDIT reports (continued)		
Description	Audit Repair Action	Troubleshooting Hints
The number of objects for dump copy prefix offloaded to cloud object store does not match the number of objects stored in the DGN record.		Contact IBM support.
*ERR 218 THE DUMP DEVICE CATEGORY OF DUMP CLASS <i>dumpclass</i> IS CLOUD AND NOT CONSISTENT AS OTHER COPIES IN DGN RECORD		
Mixed CLOUD and TAPE dump classes (copies) are not allowed in a dump generation.		For more information, use LIST VOLUME ALLDUMPS BCDS.
*ERR 219 DGN INDICATES <i>volser</i> IS IN CLOUD <i>cloud_network_connection</i> BUT NO CONTAINER <i>container</i> OR OBJECTS WITH THE PREFIX <i>prefix</i> EXIST		
DGN indicates that the volume is dumped to cloud object store but no container or objects with the prefix stored in the DGN exists.		Contact IBM support.
*ERR 220 MISSING DCL RECORD FOR DUMP CLASS <i>dumpclass</i>		
The DGN record contains information for dump class <i>dumpclass</i> , but the DCL record for <i>dumpclass</i> does not exist.		Use DEFINE DUMPCLASS command to redefine the dump class, if needed.
*ERR 221 F (FRB) RECORD FOR <i>cpname</i> NOT FOUND, POSSIBLY ORPHANED G (DGN) RECORD FOR VOLUME <i>volser</i>		
DGN record is found for copy pool <i>cpname</i> , but FRB record does not exist.		Consider running AUDIT COPYPOOLCONTROLS (<i>cpname</i>) to learn more about this discrepancy.
*ERR 222 P (MCP) RECORD FOR VOLUME <i>volser</i> WITH DUMP DATE <i>dump_date</i> AND DUMP TIME <i>dump_time</i> NOT FOUND, POSSIBLY ORPHANED G (DGN) RECORD		
DGN record exists with primary volume <i>volser</i> , but either MCP record for <i>volser</i> does not exist or the volume no longer has a valid dump generation associated with the DGN record.		For more information, use LIST PRIMARYVOLUME(<i>srcvol</i>) ALLDUMPS BCDS.
*ERR 223 FIXCDS 2 x'[UFN KEY]' PATCH(X'FC' x'[PREVIOUS SIBLING KEY]') PREVIOUS SIBLING KEY MISSING		
Previous sibling key is not updated on the UFN record being audited.	<p>If there are a very few such records, the above FIXCDS command can be a quick fix. If the count is too many to fix using FIXCDS commands, run the AUDIT FILECONTROLS function with FIX option.</p> <p>The recommended option is running AUDIT FILECONTROLS with FIX option.</p>	<p>User is running the AUDIT with NOFIX option. A limit of 1000 terminates the Audit and issues the message *ERR 224. The recommended way to fix then is to run AUDIT FILECONTROLS with FIX option.</p> <p>If these messages are recurring, this may indicate that the Unix file backups were done by turning off the PATCH bit MCVTF_CTL_FIXUFN.</p>
*ERR 224 ERROR LIMIT HAS STOPPED THE AUDIT, RERUN THE AUDIT WITH FIX OPTION		
ERR 223 or ERR 225 have reached the preset limit of 1000.	Run the AUDIT FILECONTROLS with FIX option.	User is running the AUDIT with NOFIX option. Check if the PATCH bit MCVTF_CTL_FIXUFN is turned OFF which is recommended to be ON.
*ERR 225 x'[UFN KEY]' UFN NODE MARKED SOFT DELETE FOUND		

Table 76. Error codes used in AUDIT reports (continued)

Description	Audit Repair Action	Troubleshooting Hints
Indicates this UFN node on BCDS is soft deleted (UFN_INVALID=ON).	Run the AUDIT FILECONTROLS with FIX option.	User is running the AUDIT with NOFIX option. A limit of 1000 terminates the Audit and issues the message *ERR 224. The recommended way to fix then is to run AUDIT FILECONTROLS with FIX option. If user wants to retain the UFN node, then back-up the associated file once again. That is going to turn OFF the soft delete flag (UFN_INVALID) and the new node will be added appropriately.
*ERR226 x'[UFN KEY]' PREVIOUS SIBLING KEY UPDATED IN THE UFN RECORD IS INCORRECTLY REFERENCING A DIFFERENT NODE.		
The previous sibling of the record being audited is not referencing back to the node being audited.	If the number of messages exceeds 50, audit starts printing ERR 227 indicating exceeding the allowed limit for ERR 226. Audit continues. Contact IBM support when ERR 226 message is seen.	Contact IBM support for further investigation.
*ERR 227 MAX LIMIT OF ERR 226 MESSAGES REACHED		
The number of ERR 226 messages have reached a maximum limit of 50.	User has run the AUDIT FILECONTROLS with NOFIX option. It is an unusual case of too many logical errors. Contact IBM support for further investigation.	Contact IBM support for further investigation.
*ERR 228 INFO: PREVIOUS SIBLING REFERENCE FIXED FOR THE UFN RECORD [64-character UFN Key]		
When AUDIT FILECONTROLS is run with FIX option, it has updates the previous sibling reference (UFN_PSIB) for the UFN record mentioned in the message text.	Not required. This message is informational.	Not required. This message is informational.
*ERR 229 INFO: PREVIOUS SIBLING REFERENCE FIXED FOR THE UFN RECORD [64-character UFN Key]		
When AUDIT FILECONTROLS is run with FIX option, it has updated the previous sibling reference (UFN_PSIB) for the UFN record mentioned in the message text.	Not required. This message is informational.	Not required. This message is informational.
*ERR 230 CDD IS NOT FOUND FOR MIGRATED <i>dsn</i> IN CONTAINER <i>container</i>		
A migration copy for <i>dsname</i> was found in CONTAINER container without a valid CDD		Issue a HSEND DELETE command to have associated records deleted Following is an example of the HSEND DELETE command: HSEND DELETE <i>userdsn</i>
*ERR 231 THE MCD RECORD <i>MCDkey</i> IS FOR A CLOUD MIGRATION COPY BUT CONTAINS INVALID VALUES		
More than one MCD cloud provider type indicators are set to 1. For example, both MCDF_CLD_CDA_TCT and MCDF_CLD_DIRECT are both ON.		Use the FIXCDS D <i>mckey</i> DISPLAY offset command to identify invalid values and then use FIXCDS D <i>mckey</i> PATCH <i>offset</i> command to change the field with expected values. Or contact IBM support.

Unexpected errors during an audit

Unexpected errors during an audit are errors that do not relate directly to the purpose of the audit, which is to find and repair problems with DFSMSHsm data sets and controls. For example, if the AUDIT function attempts to fix a problem and the command used to complete the fix fails, the AUDIT function provides you with an error message.

If such an error occurs, the AUDIT function writes a brief message to an output data set in the form of MSG 9xx. The AUDIT function will also write an informational message (MSG 998 or 999) to describe the AUDIT function running when the error occurs.

In addition, the DFSMSHsm command activity log contains any associated ARCxxxxx messages that describe the actual problem.

Unexpected error messages

The unexpected error messages that might be issued during an audit are explained in this topic.

MSG 900 ERROR ON ADDVOL OF VOLUME(*volser*)

Explanation

An error has been encountered in performing the ADDVOL command for the VOLUME indicated.

MSG 901 ERROR ON FIXCDS COMMAND

Explanation

An error has been encountered in performing the FIXCDS command.

MSG 902 GENCB ERROR

Explanation

An error has been encountered in performing the GENCB macro.

MSG 903 ERROR LOCKING CDS

Explanation

An error has been encountered while RESERVING/ENQUEUEING or RELEASING/DEQUEUEING the DFSMSHsm CDSs.

MSG 904 ERROR BUILDING MVT FOR VOLUME(*volser*)

Explanation

An error has been encountered during the building of a volume table entry for the VOLUME indicated.

MSG 905 [ALLOCATION | DEALLOCATION] ERROR FOR [(*dsname*) | VOLUME(*volser*)]

Explanation

An error has been encountered during ALLOCATION DEALLOCATION processing for the DATASET indicated or the VOLUME indicated.

MSG 906 LOCATE ERROR ON VOLUME(*volume*)

Explanation

An error has been encountered in the performing of the LOCATE macro for the VOLUME indicated.

MSG 907 [GETMAIN | FREEMAIN] ERROR FOR SUBPOOL(*subpool*)

Explanation

An error has been encountered in the performing of either the GETMAIN or the FREEMAIN macro.

MSG 908 ERROR CLOSING DSN(*dsname*) [VOLUME(*volser*) | SDSP FOR VOLUME(*volser*)].

Explanation

An error has been encountered during the closing of the data set *dsname*. The data set is on the volume *volser*.

MSG 909 ERROR OPENING DSN(*dsname*) [VOLUME(*volser*) | SDSP FOR VOLUME(*volser*)]

Explanation

An error has been encountered during the opening of the data set *dsname*. The data set is on the volume *volser*.

MSG 910 ERROR READING [DSN | VTOC] FOR VOLSER(*volser*)

Explanation

An error has been encountered during the reading of the data set indicated. The data set is on either the SDSP VOLUME indicated or the VOLUME indicated.

MSG 911 **ERROR DELETING CATALOG ENTRY FOR {record_key}**

Explanation

An error has been encountered during the DELETING of the indicated CATALOG entry.

MSG 912 **ERROR [DELETING | READING | WRITING] CDS RECORD, KEY = {record_key}**

Explanation

An error has been encountered during either DELETING, READING, or WRITING a record in DFSMSHsm's CDS records.

MSG 913 **ERROR CONVERTING MCP RECORD, KEY = {record_key}**

Explanation

An error has been encountered that prevented the indicated MCP record from being converted to the release 3 format.

MSG 914 **ERROR ON BDELETE COMMAND**

Explanation

An error has been encountered in performing the BDELETE command.

MSG 915 **VOLUME *volser* HAS A DISASTER ALTERNATE VOLUME**

Explanation

The volume *volser* has a disaster alternate volume that is currently being used. AUDIT MEDIACONTROLS of volumes with disaster alternate volumes is not permitted.

MSG 916 **I/O ERROR ON VOLUME *volser***

Explanation

An I/O error has been encountered reading the tape during AUDIT processing.

MSG 917 **AUDIT SKIPPING EXISTING TTOC RECORDS FOR VOLUME *volser***

Explanation

AUDIT has detected in-storage TTOC records for this volume that were not written to the OCDS. This allows AUDIT to fast forward to an indicated point on this tape volume, reducing the time required to audit this tape and restore the OCDS records.

MSG 919 **AUDIT MEDCTL CANNOT RESUME ON VOLUME *volser*.**

Explanation

AUDIT has detected an invalid parameter or the tape is in FAILEDCREATE status. ARC0876I was issued to the user and the command activity log explaining why audit could not resume.

MSG 920 **ORIGINAL AUDIT COMMANDS WILL NOT PROVIDE COMPLETE RESULTS. IBM RECOMMENDS USE OF THE PREFERRED ENHANCED AUDIT COMMANDS.**

Explanation

You have issued an ORIGINAL DFSMSHsm AUDIT command option. DFSMSHsm's Original AUDIT commands are no longer being updated. It is recommended that you identify an equivalent ENHANCED DFSMSHsm AUDIT command option to audit your DFSMSHsm Control Data Set Information. To assist you in determining which ENHANCED AUDIT command would serve you best, refer to the table 'Comparison of AUDIT options'

MSG 922 **COPY POOL=*cpname* VERSION=*version* IS NOT SUPPORTED ON THIS RELEASE OF DFSMSHSM. IBM RECOMMENDS RE-RUNNING THE AUDIT ON A HOST WHICH SUPPORTS THIS COPY POOL BACKUP VERSION**

Explanation

The audited copy pool backup version has attributes which are not supported by the release of the DFSMSHsm host on which the audit was performed. Issue the AUDIT command again on a DFSMSHsm host running on a higher release of z/OS to ensure that the copy pool version is properly audited.

For example, DFSMSHsm running on z/OS V1R12 supports auditing a copy pool backup version with an incomplete fast reverse restore. If the audit command is issued on a DFSMSHsm host running on z/OS V1R11, the incomplete fast reverse restore attribute is not supported and cannot be audited. This message is issued.

**MSG 924 UNEXPECTED ERROR OCCURRED
 DURING CALLS TO THE CLOUD**

Explanation

AUDIT has encountered an unexpected error when attempting to access CLOUD. See preceding error messages for more information about the cloud services error.

MSG 925 SHA 256 HASH IS NOT AVAILABLE

Explanation

AUDIT has encountered a HASH error during FILECONTROL processing.

**MSG 926 ERROR IN DETERMINING DGN
 KEY FROM CLOUD DUMP PREFIX
 prefix_name.**

Explanation

AUDIT has encountered an error when attempting to determine the dump generation record (DGN) key from the cloud dump copy name *prefix_name*.

**MSG 998 AUDIT CONTINUING, COMMAND
 ACTIVITY LOG MAY CONTAIN
 ADDITIONAL INFORMATION**

Explanation

AUDIT has encountered an unexpected error. This message is preceded by a 9xx message that indicates the type of error and related information. Additional information may be found in the DFSMSHsm command activity log. The AUDIT command continues with either the next data set or volume.

**MSG 999 AUDIT ENDING EARLY, SEE
 COMMAND ACTIVITY LOG**

Explanation

The AUDIT function has encountered an unexpected error. This message is preceded by a 9xx message that indicates the type of error and related information. Additional information can be found in the DFSMSHsm command activity log. The AUDIT command is ENDING because of the error encountered.

Errors encountered during AUDIT VOLUMES, AUDIT BACKUPVOLUMES, or AUDIT BACKUPTYPE

AUDIT can encounter certain conditions, as follows, when requested to audit data set information on a primary, migration, or backup volume. The conditions are listed in the order in which they might be encountered.

Also, if an error occurs when AUDIT tries to read the job file control block (JFCB) when opening the VTOC, the AUDIT command sends the identification of the module with the error to the DFSMSHsm log and to the operator and abnormally ends auditing with a dump.

**ARC0500I CANNOT ALLOCATE VOLUME
 volser, DARC=*reascode***

Explanation

The volume cannot be allocated. The error message contains the volume serial number and the return code from dynamic allocation.

**ARC0525I *dsname* SCRATCH FAILED ON
 VOLUME=*volser*,RC=*retcode***

Explanation

In attempting to scratch (or scratch and uncatalog) a utility data set, either the scratch or uncatalog operation failed.

**ARC0805I I/O ERROR OCCURRED DURING
 READING OF VTOC ON VOLUME
 *volser***

Explanation

When reading data set control blocks, AUDIT encountered an I/O error.

**ARC0806I OPEN OF VTOC FOR VOLUME *volser*
 FAILED**

Explanation

AUDIT was not able to open the volume table of contents (VTOC) to read the data set control blocks.

**ARC0807I INSUFFICIENT SPACE FOR
 READING VTOC OF VOLUME *volser***

Explanation

AUDIT could not get buffer space to read the data set control blocks.

ARC0808I UNABLE TO AUDIT VOLUME *volser*

Explanation

The volume has not been previously identified with the ADDVOL command, or is not mounted.

Other considerations when using the AUDIT command

There might be other considerations when you use the AUDIT command, such as with using multiple DFSMSHsm hosts, printing audit information, and encountering errors.

Multiple DFSMSHsm host considerations when using the AUDIT command

For the greatest accuracy in a multiple-host HSMplex, with any AUDIT command other than AUDIT COMMONQUEUE, issue the command from one host after you have stopped DFSMSHsm in the other hosts. For most users, this is unrealistic even though AUDIT may report discrepancies due to the concurrent operation of multiple hosts. If you cannot stop DFSMSHsm in the other hosts, use one of the following approaches when you issue the AUDIT command:

- Run the AUDIT NOFIX command once, and then run it again several days later. Those errors common to both executions are likely to be true discrepancies in the CDSs.
- Execute the AUDIT command from one host immediately after restoring a CDS from backups, before other hosts are started.
- For AUDIT COMMONQUEUE, it is best to issue the command when all hosts are active and connected.

Note: If AUDIT is running and backup of the DFSMSHsm control data sets has started (using BACKVOL CDS command or AUTOBACKUP), all DFSMSHsm functions on the host that started this backup are halted until the AUDIT function and the backup of the DFSMSHsm CDSs have completed.

Printing information from the AUDIT command

When you have decided what to audit, you can specify whether you want to audit information only for data sets with error conditions or all data sets. You can also specify if you want the AUDIT command to try to fix the error condition automatically.

When you issue the AUDIT command, printer output is always produced and sent to the SYSOUT data set or to an alternative output data set. When you issue AUDIT &ellip; TERMINAL, output is printed at the terminal from which the command is issued in addition to the printer output.

If you have the output sent to an alternative output data set, you identify the data set name with the OUTDATASET parameter. It is recommended that you always direct your AUDIT output to an alternative data set, and that you thoroughly review this output for content, accuracy, and understanding of corrections. At times, the output will contain further diagnostic and troubleshooting recommendations for errors that were encountered. Always investigate any reported MSGxxx error, and you should understand and accept any failures, and not ignore them.

Note: This is not applicable for AUDIT COMMONQUEUE.

Chapter 76. Using the LIST command

You use the LIST command to determine the status of DFSMSShsm and SMS-managed volumes and data sets. When you list primary volume, migration volume, or data set information, you can decide whether DFSMSShsm lists the information from the MCDS, the BCDS, or both. You can also decide to let DFSMSShsm print the output from the LIST command at your terminal, send it to a SYSOUT data set, or send it to an alternative output data set. This topic describes the categories of information you can list, explains the output of the LIST command, and contains sample printer output and sample terminal output for all categories of information.

All references to primary volumes in this topic include the SMS-managed volumes that have been internally ADDVOLED to DFSMSShsm, thereby becoming eligible for DFSMSShsm automatic processing.

You can list the following categories of information:

- [Aggregate backup and recovery activity](#)
- [Backup volumes](#)
- [Cloud storage](#)
- [Copy pools](#)
- [Copy pool backup storage groups](#)
- [Data sets](#)
- [Dump class](#)
- [Dump volumes](#)
- [Migration or primary volume from MCDS or BCDS](#)
- [DFSMSShsm host serialization \(HOSTID\)](#)
- [Tape volume from TTOC](#)
- [User authorization](#)

Requesting information for aggregate backup and recovery activity information

To get a list of aggregate backup and recovery information for all the aggregate groups that DFSMSShsm manages, specify LIST AGGREGATE(*). This command orders the output in alphanumeric sequence by aggregate group name. To get a list of the information for all the active versions of a specific aggregate group that DFSMSShsm manages, specify LIST AGGREGATE(*agname*).

Table 77 on page 1341 explains the output labels for any aggregate volume.

Table 77. Output labels when you request information for aggregate backup and recovery		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
ABR RECORD KEY	ABRKEY	This field contains the key of the ABR record.
AGGREGATE GROUP NAME	AGNAME	This field contains the name of the aggregate group processed.
DFSMSShsm VERSION	HSMVERS	This field contains the DFSMSShsm version and release number.
AGGREGATE ACCOUNT CODE	AGGREGATE ACCOUNT CODE	This field contains an installation-defined accounting code for the aggregate.
ABACKUP DATE	ABKDATE	This field contains the date when ABACK was performed.

Table 77. Output labels when you request information for aggregate backup and recovery (continued)

Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
ABACKUP TIME	ABKTIME	This field contains the time when ABACK was performed.
VERSION	VERSION	This field contains the version number for this ABACKUP.
COPY NUMBER	COPYNUM	This field contains the copy number to which this ABR record pertains.
UNITNAME	UNIT	This field contains the unitname.
NUMBER OF COPIES	NUMCOPIES	This field contains the number of copies created for this version of the ABACKUP output.
SOURCE SYSTEM	SRCSYS	This field contains the source system name.
MANAGEMENT CLASS	MGMTCLS	This field contains the name of the management class that is associated with the aggregate group that was backed up.
REMOTE DESTINATIONS	REM-DEST	This field is reserved for future function.
CONTROL FILE NAME	CF	This field contains the data set name of the control file.
DFSMSdss DATA FILE NAME	DFLO	This field contains the data set name of the DFSMSdss data file.
INTERNAL I/O DATA FILE NAME	DFINT	This field contains the data set name of the Internal/IO data file.
ACTIVITY LOG/ INSTRUCTION FILE NAME	INS/LOGF	This field contains the data set name of the instruction/activity log file. ** ERROR ** at the end of the data set name field indicates an error occurred during the ABACKUP. The file is not recoverable.
ABACKUP ACTIVITY LOG DATA SET NAME	ABKLOGDSN	This field contains the data set name of the ABACKUP activity log.
INSTRUCTION DATA SET NAME	INSTDSN	This field contains the data set name of the instruction data set.
FILTER OUTPUT DATA SET NAME	FODSN	This field contains the data set name of the FILTER OUTPUT data set.
CONTROL FILE VOLSERS	CFVOLS	This field contains a list of the volume serial numbers that contain the control file and associated library names.
DFSMSdss DATA FILE VOLSERS	DFLOVOLS	This field contains a list of the volume serial numbers that contain the DFSMSdss data file and associated library names.
INTERNAL I/O DATA FILE VOLSERS	DFINTVOLS	This field contains a list of the volume serial numbers that contain the I/O data file and associated library names.
INSTRUCTION / ACTIVITY LOG FILE VOLSERS	INS/LOGFVOLS	This field contains a list of the volume serial numbers that contain the instruction/activity log file and associated library names.
TAPE EXPIRATION DATE	TPEXPDATE	This field contains the expiration date for the ABARS output files.
NUMBER OF USER TAPES	DSTPCNT	This field contains the number of tapes.
NUMBER OF ACCOMPANY TAPES	ACCTPCNT	This field contains the number of tape volumes containing data sets from the ACCOMPANY list.
STORAGE USED AT ABACKUP SITE LO, ML1, ML2, TOTAL	ABKLOSTO, ABKML1STO, ABKML2STO, ABKTOTSTO	This field contains the amount of data that was backed up from the various hierarchy levels. LO=Level 0 DASD, ML2=ML2 tape/DASD, ML1=ML1 DASD, TOTAL=total amount of data backed up.

Table 77. Output labels when you request information for aggregate backup and recovery (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
ABACKUP CPU TIME	ABACKUP CPU TIME	This field contains the CPU time that was used for the ABACKUP of this aggregate group.
TAPE STACKING	TAPE STACKING	This field indicates if the ABACKUP output files have been stacked or not stacked.
DATE	ARVDATE-TIME	This field contains the date when the last ARECOVER was performed.
RETCODE	ARVRC	This field contains the return code of the most recent ARECOVER.
OPTION	ARVOPT	This field contains the performed ARECOVER option: EXECUTE/VERIFY/PREPARE
XMIT=Y/N	XMIT	This field indicates if the XMIT option was requested.
STORAGE REQUIREMENTS LO, ML1, ML2, TOTAL	ABKLOSTO, ABKML1STO, ABKML2STO, ABKTOTSTO	This field contains the storage required to complete recovery.
ARECOVER ACTIVITY LOG DATA SET NAME	ARVLOGDSN	This field contains the data set name of the ARECOVER activity log that is associated with the latest ARECOVER.
CONFLICT RESOLUTION DATA SET NAME	CONDSN	This field contains the data set name of the CONFLICT RESOLUTION data set that is associated with the latest ARECOVER.
RESTART DATA SET NAME	RSTDSN	This field contains the data set name of the RESTART data set that is associated with the latest ARECOVER.
ARECOVER CPU TIME	ARECOVER CPU TIME	This field contains the CPU time used for the ARECOVER of this aggregate group.
TAPE STACKING	TAPE STACKING	This field indicates if the ABACKUP output files have been stacked or not stacked.
TOTAL BACKUP VERSIONS	TOTAL BACKUP VERSIONS	This field contains the total number of backup versions for each data set. This total includes all active and retained backup versions.

Figure 110 on page 1344 is a sample of a terminal list when you have specified AGGREGATE(PAY1) VERSION(0002), and TERMINAL parameters.

Aggregate backup and recovery activity

```
ARC0300I **OPER** ISSUED===>LIST AGGREGATE(AGR1) TERM
ABRKEY=AGR1.1996290000101
HSMVERS=V1.4.0 AGNAME=AGR1
AGGREGATE ACCOUNT CODE=TESTACCTCODEUPT032CHARS
VERSION=0001 COPYNUM=01 NUMCOPIES=01
ABKDATE-TIME=1996/10/16, 15:48 UNIT=3590-1 SRCSYS=3090
MGMTCLS=NONE
REM-DEST= DESTID TRANS-OK
      NONE
CF=AGR1.C.C01V0001
DFLO=AGR1.D.C01V0001
DFINT=AGR1.O.C01V0001
INS/LOGF=AGR1.I.C01V0001
ABKLOGDSN=HSMACT.H1.ABACKUP.AGR1.D96290.T154833
INSTDSN=HSMATH0.TEST.INSTRUCT
FODSN=NONE
CFVOLS=
VOLS= A00058
LIBS= *NO_LIB*
DFLOVOLS=
VOLS= A00058
LIBS= *NO_LIB*
DFINTVOLS=
VOLS= A00058
LIBS= *NO_LIB*
INS/LOGFVOLS=
VOLS= A00058
LIBS= *NO_LIB*
TPEXPDATE=1999/12/31 DSTPCNT=0000 ACCTPCNT=0000
ABKL0STO=0000001141K ABKML1STO=0000000000K
ABKML2STO=0000000000K ABKTOTSTO=0000001141K
ABACKUP CPU TIME=00:00:00.32
TAPE STACKING = YES
ARC0140I LIST COMPLETED, 32 LINE(S) OF DATA OUTPUT
```

Figure 110. Sample Terminal List when You Specify AGGREGATE(PAY1), VERSION(002) and TERMINAL.

Figure 111 on page 1344 is a sample of a printer list when you have specified AGGREGATE(PAY1) VERSION(0001), and ODS.

```
--- DFSMSHSM CONTROL DATASET AGGREGATE BACKUP AND RECOVERY VERSION LISTING ----
AT 15:28:29 ON 1996/10/16 FOR SYSTEM=3090

ABR RECORD KEY = AGR1.1996290000101
AGGREGATE GROUP NAME = AGR1
AGGREGATE ACCOUNT CODE = TESTACCTCODEUPT032CHARS
ABACKUP DATE = 1996/10/16 ABACKUP TIME = 15:25:59
VERSION = 0001 COPY NUMBER = 01
UNIT NAME = 3590-1 NUMBER OF COPIES = 01
SOURCE SYSTEM = 3090
MANAGEMENT CLASS = NONE
REMOTE DESTINATIONS = DESTINATION ID TRANS. OK
      NONE

CONTROL FILE NAME = AGR1.C.C01V0001
DFSMSDSS DATA FILE NAME = AGR1.D.C01V0001
INTERNAL I/O DATA FILE NAME = AGR1.O.C01V0001
ACTIVITY LOG/INSTRUCTION FILE NAME = AGR1.I.C01V0001

ABACKUP ACTIVITY LOG DATA SET NAME = HSMACT.H1.ABACKUP.AGR1.D96290.T152559
INSTRUCTION DATA SET NAME = HSMATH0.TEST.INSTRUCT
FILTER OUTPUT DATA SET NAME = NONE

CONTROL FILE VOLSERS =
VOLS= A00048
LIBS= *NO_LIB*
DFSMSDSS DATA FILE VOLSERS =
VOLS= A00048
LIBS= *NO_LIB*
INTERNAL I/O DATA FILE VOLSERS =
VOLS= A00048
LIBS= *NO_LIB*
INSTRUCTION/ACTIVITY LOG FILE VOLSERS =
VOLS= A00048
LIBS= *NO_LIB*
TAPE EXPIRATION DATE = 1999/12/31

NUMBER OF USER TAPES = 0000
NUMBER OF ACCOMPANY TAPES = 0000

STORAGE USED AT ABACKUP SITE :
      L0 ML1 ML2 TOTAL
0000001141K 0000000000K 0000000000K 0000001141K

ABACKUP CPU TIME = 00:00:00.30
TAPE STACKING = YES

ARECOVER INFORMATION: NONE

----- END OF - AGGREGATE BACKUP AND RECOVERY VERSION - LISTING -----
```

Figure 111. Sample Printer List when You Specify AGGREGATE(PAY1), VERSION(001) and ODS.

Requesting information for backup volumes

You specify the LIST command with the BACKUPVOLUME parameter to get a list of the information for all the backup volumes that DFSMSHsm manages. The command lists the volumes in alphanumeric sequence by volume serial number. You specify LIST BACKUPVOLUME(volser) to get a list of the information for a specific backup volume that DFSMSHsm manages. [Table 78 on page 1345](#) explains the output labels for any backup volume.

Table 78. Output labels when you request information for backup volumes		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
VOLSER	VOL	This field contains the volume serial number of the backup volume.
DEVICE TYPE	DEVTYPE	This field contains the name of the unit where this volume can be allocated.
BACKUP TYPE	BACKTYP	DAILY or SPILL indicates the DFSMSHsm volume category of the backup volume. AVAIL indicates that the backup category has not been assigned and the volume is available as a daily or spill backup volume when DFSMSHsm uses the volume for the first time.
VOL FULL	FULL	For a DASD backup volume, a YES indicates that DFSMSHsm has failed to back up a data set because the backup volume does not have enough space. For a tape backup volume, a YES indicates that while DFSMSHsm is writing to the tape, the end-of-tape marker is reached or a data-movement error has occurred. The volume has been marked full to prevent further use.
TOTAL TRACKS	TOTAL TRKS	This field contains the total track capacity of the DASD volume. This field does not apply to tape.
FREE TRACKS	FREE TRKS	This field contains the number of tracks on the DASD volume available for data sets. This field does not apply to tape.
THRESH	THRESH	This field contains the threshold of occupancy for the DASD backup volume. This field does not apply to tape.
LAST BACKUP DATE	LAST BACKUP DATE	This field contains the date when the volume was most recently used as a target volume for backup, spill, or recycle processing.
PSWD	PSWD	A YES or Y indicates that the tape volume is password protected. This field does not apply to DASD.
EXP	EXP	A YES or Y indicates that the tape volume is protected by an expiration date. This field does not apply to DASD.
RACF	RACF	A YES or Y indicates that the tape volume is RACF protected. This field does not apply to DASD.
EMPTY	EMPTY	A YES or Y indicates that the tape volume is empty. This field does not apply to DASD.
IDRC	IDRC	Y = Volume contains data in the Improved Data Recording Capability format. N = Volume does not contain data in the Improved Data Recording Capability format. ** = Volume is empty—not assigned the Improved Data Recording Capability format.

Table 78. Output labels when you request information for backup volumes (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DUPLEX ALT	DUPLEX ALT	<p><i>volser</i> indicates the volume serial number of the duplexed alternate</p> <p>*PEND* indicates that an exception condition and an internal TAPECOPY is pending</p> <p>*NONE* indicates that the volume was not created in a duplexing environment</p> <p>NA indicates that the original volumes are DASD rather than tape</p>
PCT FULL	PCT FULL	This field contains the percentage of the backup volume used for a tape written using 3590 architecture. A value of **** means the percentage is unknown.

Figure 112 on page 1346 is a sample printer list of all the backup volumes when you specify the BACKUPVOLUME parameter. If you requested information for only one volume, the list has only one entry.

Note: If **** appears in the FREE TRACKS field for a DASD backup volume, DFSMSHsm has not yet calculated—and therefore does not know—the amount of free space on this volume.

```

1- DFSMSHSM CONTROL DATASET - BACKUP VOLUME-- LISTING ----- AT 10:55:46 ON 01/04/03 FOR SYSTEM=3090
VOLSER  DEVICE  BACKUP  VOL  TOTAL  FREE  THRESH  LAST BACKUP  PSWD  EXP  RACF  EMPTY  IDRC  DUPLEX  PCT
TYPE    TYPE    TYPE    FULL TRACKS TRACKS          DATE                                     ALT    FULL
A00002   3480   DAILY    NO   ***** *****   ***   00/09/05     YES  NO   NO   NO   N   *NONE*   ****
A00017   3480   DAILY    YES   ***** *****   ***   00/08/25     YES  NO   NO   NO   N   *NONE*   ****
A00018   3480   DAILY    NO   ***** *****   ***   00/10/31     YES  NO   NO   NO   N   *NONE*   ****
BACK01   3390   DAILY    NO   00000150 00000021 090   01/04/02     ***  ***  ***  ***  NA   NA       NA
BACK02   3390   DAILY    NO   00000150 00000023 090   01/04/03     ***  ***  ***  ***  NA   NA       NA
BACK03   3390   SPILL     NO   00000150 *****   090   00/00/00     ***  ***  ***  ***  NA   NA       NA
----- END OF - BACKUP VOLUME - LISTING -----

```

Figure 112. Sample Printer List of All the Backup Volumes when You Specify BACKUPVOLUME

Requesting information for cloud storage

You specify the LIST CLOUD (*cloud_network_connection_name*) to get a list of the contents of cloud storage at various levels. The first level, LIST CLOUD(*cloud_network_connection_name*), returns a list of the DFSMSHsm containers that are found in the specified cloud. The next level, LIST CLOUD(*cloud_network_connection_name*) CONTAINER(*containername*), returns a list of the pseudo-directories found within the container. The pseudo-directory name is a prefix for all of the objects within a particular migration copy. The last level, LIST CLOUD(*cloud_network_connection_name*) CONTAINER(*containername*) PREFIX(*objectprefixname*), returns a list of all of the objects that make up that migration or full volume dump copy.

Table 79. Output labels for List of Cloud Storage Contents		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
CLOUD NAME	CLOUD NAME	The <i>cloud_network_connection_name</i> from which the listing was retrieved
CONTAINER NAME	CONTAINER NAME	Name of container

Table 79. Output labels for List of Cloud Storage Contents (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
PREFIX NAME	OBJECT PREFIXES	Pseudo-directory name for the migration copy or the full-volume dump copy residing in cloud. It is a common unique prefix for all objects that make up a migration copy or the full-volume dump copy.
CLOUD INFORMATION FOR DATASET	DATASET NAME	For migration copy residing in cloud, this field contains the name of the original data set that was migrated, when an APPMETA object is found with valid DFSMSHsm information. This field does not apply to full volume dump copies residing in cloud object store. 'CDD NOT FOUND' is displayed.
OBJECT NAME	OBJECT NAMES	The full object name

Figure 113 on page 1347 is a sample printer list of a specific cloud when you specify the CLOUD(*cloud_network_connection_name*) parameter:

```
- DFSMSHSM CONTROL DATASET - CLOUD INFO    -%CLD--- LISTING ----- AT 12:53:43 ON 22/01/10 FOR SYSTEM=3090

CLOUD NAME                CONTAINER NAME
PRODCLOUD01               SYSZARC.ARCPLEX0.DMP.2021185.6646C392
PRODCLOUD01               SYSZARC.ARCPLEX0.DMP.2021277.43ABF490
PRODCLOUD01               SYSZARC.ARCPLEX0.DMP.2021277.E6CF5013
PRODCLOUD01               SYSZARC.ARCPLEX0.DMP.2021277.FA0D34F1
PRODCLOUD01               SYSZARC.ARCPLEX0.DMP.2022001.252DB985
PRODCLOUD01               SYSZARC.ARCPLEX0.MIG.2021330.907B3C81
----- END OF - CLOUD INFO - LISTING -----

- DFSMSHSM CONTROL DATASET - CLOUD INFO    -%CLD--- LISTING ----- AT 10:30:11 ON 22/01/10 FOR SYSTEM=3090

CLOUD NAME                CONTAINER NAME
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021281.59C1CB82
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021295.5002E05B
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021302.549C8CA8
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021260.6905F63F
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021274.3BA65EA2
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021274.93DC5088
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021302.BB177F12
PRODCLOUD02               SYSZARC.ARCPLEX0.DMP.2021277.CC841FA0
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021309.F67A6D7D
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021260.92075591
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021309.835A6488
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021267.7E5ABC22
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021267.E71454E1
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021274.ECA25C71
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021295.21E35318
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021288.30178B40
PRODCLOUD02               SYSZARC.ARCPLEX0.MIG.2021267.C3E07ECF
----- END OF - CLOUD INFO - LISTING -----
```

Figure 113. Sample Printer List of a specific cloud when you specify the CLOUD(*cloud_network_connection_name*)

Figure 114 on page 1348 are sample printer lists of specific objects when you specify the CLOUD(*cloud_network_connection_name*) CONTAINER(*container_name*) parameter:

Copy pools

```
- DFSMSHSM CONTROL DATASET - CLOUD INFO      -%CLD--- LISTING ----- AT 10:47:13 ON 21/09/22 FOR SYSTEM=3090
CLOUD NAME                                CONTAINER NAME                                PREFIX NAME
PRODCLOUD02                               SYSZARC.ARCPLEXY.MIG.2021185.A33258A5          DFHSM.HMIG.T513210.HSMATH1.SC11.B1265
                                                DFHSM.HMIG.U513210.HSMATH2.SC11.B1265
                                                DFHSM.HMIG.V513210.HSMATH2.SC11.B1265
----- END OF - CLOUD INFO - LISTING -----
- DFSMSHSM CONTROL DATASET - CLOUD INFO      -%CLD--- LISTING ----- AT 16:17:28 ON 21/09/17 FOR SYSTEM=3090
CLOUD NAME                                CONTAINER NAME                                PREFIX NAME
PRODCLOUD02                               SYSZARC.ARCPLEXY.DMP.2021185.F80DCC3A          DFHSM.DMP.DCL11S.VSMS062.D21260.T525314
                                                DFHSM.DMP.DCL12S.VSMS064.D21260.T315514
                                                DFHSM.DMP.DCL12S.VSMS065.D21260.T335514
                                                DFHSM.DMP.DCL9S.VSMS063.D21260.T581116
----- END OF - CLOUD INFO - LISTING -----
```

Figure 114. Sample Printer List of specific objects when you specify the CLOUD(*cloud_network_connection_name*) CONTAINER(*container_name*)

Figure 115 on page 1348 is a sample printer list of a specific objects when you specify the CLOUD(*cloud_network_connection_name*) CONTAINER(*container_name*) PREFIX(*prefix_name*) parameters:

```
- DFSMSHSM CONTROL DATASET - CLOUD INFO      -%CLD--- LISTING ----- AT 21:06:45 ON 22/01/10 FOR SYSTEM=3090
CLOUD INFORMATION FOR DATASET: CDD NOT FOUND
CLOUD NAME                                CONTAINER NAME                                OBJECT NAME
PRODCLOUD01                               SYSZARC.ARCPLEX0.DMP.2022001.252DB985          DFHSM.DMP.DCL9.VSMS061.D22010.T304512/HDR
                                                DFHSM.DMP.DCL9.VSMS061.D22010.T304512/SMS061
                                                /DATA/EXTENTS000000001
                                                DFHSM.DMP.DCL9.VSMS061.D22010.T304512/SMS061
                                                /DATA/EXTENTS000000001/CMPMETA
                                                DFHSM.DMP.DCL9.VSMS061.D22010.T304512/SMS061
                                                /META/DTPBTMC
                                                DFHSM.DMP.DCL9.VSMS061.D22010.T304512/SMS061
                                                /META/DTPBTMT
                                                DFHSM.DMP.DCL9.VSMS061.D22010.T304512/SMS061
                                                /META/DTPTRK0
                                                DFHSM.DMP.DCL9.VSMS061.D22010.T304512/SMS061
                                                /META/DTPVTOC
                                                DFHSM.DMP.DCL9.VSMS061.D22010.T304512/SMS061
                                                /META/DTPVVD5
                                                DFHSM.DMP.DCL9.VSMS061.D22010.T304512/SMS061
                                                /META/DTPVVD5_chunked_part_0
----- END OF - CLOUD INFO - LISTING -----
```

Figure 115. Sample Printer List of specific objects when you specify the CLOUD(*cloud_network_connection_name*) CONTAINER(*container_name*) PREFIX(*prefix_name*)

Requesting information for copy pools

You specify the LIST command with the COPYPOOL parameter to get a list of information for each version of each copy pool that has at least one fast replication backup version. The command lists the copy pools in alphanumeric sequence by copy pool name. You specify LIST COPYPOOL(*cpname*) to get a list of the detailed information for each version of the specified copy pool. The information includes source volume to target volume pairings. Table 80 on page 1348 explains the output labels for copy pools.

Table 80. Output labels when you request information for copy pools		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
CATINFO	CATINFO	This field indicates whether data set catalog information was captured for this version. CATINFO=Y/N/P (YES/NO/PARTIAL)
COPYPOOL	COPYPOOL	This field contains the copy pool name.
*DATA SET NAME	Not Supported	Data sets backed up in this copy pool version.
DATE	DATE	This field contains the creation date of the copy pool version.
DCLASS	DCLASS	This fields contains the changes to existing output formatting.

Table 80. Output labels when you request information for copy pools (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DUMPCCLASS	DUMPCCLASS	This field indicates if the dump class associated with the copy pool is required or optional.
DUMPDEVCAT	DUMPDEVCAT	This field indicates the dump device category as follows: Value Meaning CLOUD Indicates the copy pool dump version resides in the cloud object store. TAPE Indicates the copy pool dump version resides on tapes.
DUMPSTATE	DUMPSTATE	This field indicates the dump state of volumes in the copy pool.
FASTREPLICATIONSTATE	FASTREPLICATIONSTATE	This field indicates the fast replication state of the copy pool.
FCFRR	FCFRR	This field indicates whether the copy is defined allowing fast reverse restore (FCFRR=Y) or not (FCFRR=N).
INCREMENTAL	INCREMENTAL	This field indicates whether the corresponding version is an Incremental version.
RECOVERYINCOMPLETE	RECOVERYINCOMPLETE	This field indicates whether the copy pool version was recovered completely.
*SGNAME	SGNAME	This field contains the storage group name.
*SOURCE	SOURCE	This field lists source volumes in the storage group.
*TARGET	TARGET	This field lists target volumes in the storage group.
TIME	TIME	This field contains the creation time of the copy pool version.
TOKEN (C)	TOKEN (C)	This field contains the token in alphanumeric format.
TOKEN (H)	TOKEN (H)	This field contains the token in hexadecimal format.
TOTAL NUM OF VOLUMES	TOTAL NUM OF VOLUMES	This field lists the total number of volumes backed up for this copy pool version.
VERSION	VERSION	This field contains the copy pool version number.
VALID	VALID	This field indicates whether the copy pool version is valid, as follows: Value Meaning Y Indicates that the copy pool version contains a valid DASD backup copy. N Indicates that the copy pool version does not contain a valid DASD backup copy. * Indicates that the copy pool version is in a state defined in a z/OS V1R8 or later environment, and thus, cannot be used for recovery on this host system.
VTOCENQ	VTOCENQ	This field indicates whether DFSMSshm serialized on the VTOCs of the volumes that were processed.

Note: * Appears only if you specify COPYPOOL(*cpname*).

Figure 116 on page 1350 is a sample printer list of all the copy pools when you specify the COPYPOOL parameter.

```
-- DFSMSHsm CONTROL DATASET --COPY POOL--LISTING ----- AT 00:56:24 ON 09/10/12 FOR SYSTEM=1234

COPYPOOL=CP1
ALLOWPPRCP FRB=PN FRR=PN

VERSION  VTOCENQ  DATE      TIME      FASTREPLICATIONSTATE  DUMPSTATE
002      Y       2009/10/12  00:44:36  RECOVERABLE           ALLCOMPLETE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=00002, INCREMENTAL=N, CATINFO=N, FCFRR=Y, RECOVERYINCOMPLETE=N

VERSION  VTOCENQ  DATE      TIME      FASTREPLICATIONSTATE  DUMPSTATE
001      Y       2009/10/12  00:20:17  RECOVERABLE           ALLCOMPLETE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=00002, INCREMENTAL=N, CATINFO=N, FCFRR=Y, RECOVERYINCOMPLETE=N

COPYPOOL=CP2
ALLOWPPRCP FRB=NO FRR=NO

VERSION  VTOCENQ  DATE      TIME      FASTREPLICATIONSTATE  DUMPSTATE
001      Y       2009/10/12  00:54:13  RECOVERABLE           NONE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=00002, INCREMENTAL=N, CATINFO=N, FCFRR=Y, RECOVERYINCOMPLETE=N

COPYPOOL=CP4
ALLOWPPRCP FRB=NO FRR=NO

VERSION  VTOCENQ  DATE      TIME      FASTREPLICATIONSTATE  DUMPSTATE
001      Y       2009/10/12  00:55:08  RECOVERABLE           NONE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=00002, INCREMENTAL=N, CATINFO=N, FCFRR=Y, RECOVERYINCOMPLETE=N

----- END OF -- COPY POOL -- LISTING -----
```

Figure 116. Sample Printer List of All the Copy Pools when You Specify COPYPOOL

Figure 117 on page 1350 is a sample printer list of a specific copy pool when you specify the COPYPOOL(*cpname*) parameter.

```
-- DFSMSHsm CONTROL DATASET --COPY POOL--LISTING ----- AT 01:05:02 ON 21/10/12 FOR SYSTEM=3090

COPYPOOL=CP9
ALLOWPPRCP FRB=NO FRR=NO

VERSION  VTOCENQ  DATE      TIME      FASTREPLICATIONSTATE  DUMPSTATE
002      Y       2021/10/12  00:44:36  RECOVERABLE           ALLCOMPLETE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=00002, INCREMENTAL=N, CATINFO=N, FCFRR=Y, RECOVERYINCOMPLETE=N, FCCG=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SGRP9    SRC01B - TGT03B  SRC02B - TGT04B

DUMPCCLASS  REQUIRED  DUMPSTATE  VOLSSUC  EXPDATE  AVAILABLE  DUMPDEVCAT
DCL1        Y       COMPLETE   00002    2021/11/11  Y          TAPE

          HWCOMP  ENCRYPT  ENCTYPE  RSAKEY/KPWD
          NO      NONE      *****  *****

SOURCE  DUMPVOLS                                DEVICE TYPE
SRC01B  A00732                                3490
FILE SEQ=001, DSN=DFHSM.DMP.DCL1.VSRC01B.D21285.T364400
SRC02B  A00731                                3490
FILE SEQ=002, DSN=DFHSM.DMP.DCL1.VSRC02B.D21285.T364400

VERSION  VTOCENQ  DATE      TIME      FASTREPLICATIONSTATE  DUMPSTATE
001      Y       2021/10/12  00:20:17  RECOVERABLE           ALLCOMPLETE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=00002, INCREMENTAL=N, CATINFO=N, FCFRR=Y, RECOVERYINCOMPLETE=N, FCCG=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SGRP9    SRC01B - TGT01B  SRC02B - TGT02B

DUMPCCLASS  REQUIRED  DUMPSTATE  VOLSSUC  EXPDATE  AVAILABLE  DUMPDEVCAT
DCL1        Y       COMPLETE   00002    2021/11/11  Y          TAPE

          HWCOMP  ENCRYPT  ENCTYPE  RSAKEY/KPWD
          NO      NONE      *****  *****

SOURCE  DUMPVOLS                                DEVICE TYPE
SRC01B  A00731                                3490
FILE SEQ=001, DSN=DFHSM.DMP.DCL1.VSRC01B.D21285.T172000
SRC02B  A00730                                3490
FILE SEQ=002, DSN=DFHSM.DMP.DCL1.VSRC02B.D21285.T172000

----- END OF -- COPY POOL -- LISTING -----
```

Figure 117. Sample Printer List of a Specific Copy Pool when You Specify COPYPOOL(*cpname*)

Figure 118 on page 1351 shows a sample output for the LIST command with DATASETS option. You can use this command to find data sets that were backed up but have been deleted. Data sets are listed in alphabetical order by catalog.

```
-- DFSMSHsm CONTROL DATA SET -- COPY POOL -- LISTING -- AT 13:36:18 ON 09/03/23 FOR SYSTEM=3090
COPYPOOL=CP1 ,VER=001,GEN=000,CATINFO=Y
CATALOG INFORMATION DATA SET NAME=DFHSM.HSMCIDS.D09082.T103126.C001
TKN(C)=C' '
TKN(H)=X' '
CATALOG NAME = CP00L1Z.USERCAT
DATA SET NAME
CP00L1.FRE.N01.KSDS
CP00L1.FRE.N02.ESDS
CP00L1.FRE.N03.RRDS
CP00L1.FRE.N04.PSFB
CP00L1.FRE.N05.POFB
CP00L1.FRE.N07.POU
CP00L1.FRE.N08.DAF
CP00L1.FRE.N09.DAU
TOTAL NUMBER OF DATA SETS = 8
```

Figure 118. Sample Output for the LIST COPYPOOL(cpname) DATASETS command

Figure 118 on page 1351 is a sample printer list of a specific copy pool when you specify the COPYPOOL(cpname) parameter and one of the dump versions resides on cloud object storage:

```
-- DFSMSHsm CONTROL DATASET --COPY POOL--LISTING ----- AT 12:47:31 ON 22/01/10 FOR SYSTEM=3090
COPYPOOL=CP9
ALLOWPPRCP FRB=NO FRR=NO

VERSION  VTOCENQ   DATE       TIME       FASTREPLICATIONSTATE  DUMPSTATE
002      Y        2022/01/10    12:45:30    RECOVERABLE           ALLCOMPLETE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N,CATINFO=N,FCFRR=N,RECOVERYINCOMPLETE=N,FCCG=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SGRP9    SMS061 - SMST67  SMS063 - SMST66

DUMPCCLASS  REQUIRED  DUMPSTATE  VOLSSUC  EXPDATE  AVAILABLE  DUMPDEVCAT
DCL9        Y        COMPLETE   00002    2022/01/15    Y          CLOUD

          HWCMP  ENCRYPT  ENCTYPE  RSAKEY/KPWD
          NO     NONE    *****

SOURCE  DUMPVOLS                                DEVICE TYPE
SMS061  %CLOUD                                PRODCLOUD01
CONTAINER=SYSZARC.ARCPLX0.DMP.2022001.252DB985
FILE SEQ=001, DSN=DFHSM.DMP.DCL9.VSMS061.D22010.T304512
SMS063  %CLOUD                                PRODCLOUD01
CONTAINER=SYSZARC.ARCPLX0.DMP.2022001.252DB985
FILE SEQ=001, DSN=DFHSM.DMP.DCL9.VSMS063.D22010.T304512

VERSION  VTOCENQ   DATE       TIME       FASTREPLICATIONSTATE  DUMPSTATE
001      Y        2022/01/10    11:56:05    NONE                 ALLCOMPLETE
TOKEN(C)=C' '
TOKEN(H)=X' '
TOTAL NUM OF VOLUMES=00002,INCREMENTAL=N,CATINFO=N,FCFRR=N,RECOVERYINCOMPLETE=N,FCCG=N

SGNAME   SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET  SOURCE - TARGET
SGRP9    SMS061 - *****  SMS063 - *****

DUMPCCLASS  REQUIRED  DUMPSTATE  VOLSSUC  EXPDATE  AVAILABLE  DUMPDEVCAT
DCL1        Y        COMPLETE   00002    2022/02/09    Y          TAPE

          HWCMP  ENCRYPT  ENCTYPE  RSAKEY/KPWD
          NO     NONE    *****

SOURCE  DUMPVOLS                                DEVICE TYPE
SMS061  A07550                                3490
FILE SEQ=001, DSN=DFHSM.DMP.DCL1.VSMS061.D22010.T055611
SMS063  A07550                                3490
FILE SEQ=002, DSN=DFHSM.DMP.DCL1.VSMS063.D22010.T055611

----- END OF -- COPY POOL -- LISTING -----
```

Figure 119. Sample printer list of a specific copy pool when you specify the COPYPOOL(cpname) parameter and one of the dump versions resides on cloud object storage

Requesting information for copy pool backup storage groups

You specify the COPYPOOLBACKUPSTORAGEGROUP(*cpbsgname*) parameter on the LIST command to get a list of volumes in the copy pool backup storage group that DFSMSHsm has assigned as a target volume. The corresponding source volumes are also listed. The command lists the specified copy pool storage group's target volumes in alphanumeric sequence by volume serial number.

Figure 120 on page 1352 is a sample printer list of a specific copy pool backup storage group when you specify the COPYPOOLBACKUPSTORAGEGROUP(*cpbsgname*) parameter.

```
-- DFSMSHsm CONTROL DATASET ----CPBSG-----LISTING ----- AT 16:48:16 ON 02/09/12 FOR SYSTEM=3090
COPYPOOLBACKUP STORAGE GROUP =TARG1

SOURCE  TARGET
SRC006  TAR003
SRC007  TAR004
SRC006  TAR005
SRC007  TAR007
SRC011  TAR008
SRC012  TAR010
SRC023  TAR011
SRC018  TAR012
SRC011  TAR013
SRC019  TAR016
SRC012  TAR018
SRC023  TAR022
SRC018  TAR023
SRC025  TAR024
SRC034  TAR025
SRC036  TAR026
SRC019  TAR027
SRC025  TAR028
SRC037  TAR029
SRC034  TAR030
SRC036  TAR032
SRC037  TAR033

----- END OF -- CPBSG ----- LISTING -----
```

Figure 120. Sample Printer List for COPYPOOLBACKUPSTORAGEGROUP

Requesting information for data sets

To get a list of the information for all migrated data sets with the same first qualifier, specify the LIST command with the LEVEL(*qualifier*) parameter. The command lists the data sets in alphanumeric sequence by data set name.

To get a list of the information for *all* migrated data sets, specify the LIST command with the DATASETNAME parameter. The command lists the data sets in alphanumeric sequence by data set name followed by a summary list. To get a list of the information for a specific data set, specify the LIST command with a specific DATASETNAME(*dsname*) parameter.

You can specify one or more of the subparameters of the SELECT parameter to restrict the list to data sets that meet certain criteria.

When you request information for data sets, you can also request whether information should be listed from the migration control data set (MCDS), backup control data set (BCDS), or both control data sets.

Requesting data set information from the MCDS

Table 81 on page 1352 explains the output labels for a list of all data sets, all data sets that have the same first qualifier, or a specific data set.

Table 81. Output labels when you request MCDS information for data sets		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DATASET NAME	DSN	This field contains the name of the user data set.
MIG VOLUME	MIGVOL	This field contains the volume serial number of the migration volume the data set is on if the data set is currently migrated. If the data set is on more than one tape migration level 2 volume, this field contains the volume serial number of the first volume the data set is on. The field contains ONLINE if the data set has been recalled. If the data set has been migrated to cloud storage, this field contains %CLOUD.
LAST REF DATE	LAST REF	This field contains the date of the most recent reference of the data set.

Table 81. Output labels when you request MCDS information for data sets (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
MIGRATED DATE	MIG	This field contains the date when the data set was last migrated.
TRKS ALLOC	TRKS	If the data set is currently migrated, this field contains the number of tracks allocated for the data set on the primary volume from which the data set migrated. If the data set is recalled, this field is the number of tracks allocated for the recalled data set.
QTY 2K BLKS	2K BLKS	This field contains the size, in 2 KB blocks, of the data set on the migration volume. This field does not apply to tape. If the quantity exceeds 99,999,999, it is represented in units of K (1000 bytes) instead.
TIMES MIG	TIMES MIG	This field contains the number of times DFSMSHsm has migrated the data set. If migration cleanup has deleted the data set record, this is the number of times DFSMSHsm has migrated the data set since the record was deleted.
DS ORG	DSO	This field contains the type of data set organization: PE (partitioned data set extended) PS (physical sequential), PO (partitioned), DA (BDAM), VS (VSAM), PSL (large format sequential) or *** if the data set organization is unknown.
SDSP DS	SDSP	A YES under this heading indicates that the data set resides in a small-data-set-packing data set.
QTY 16K BLOCKS	16K BLOCKS	This field contains the size, in 16 KB blocks, of the data set on the tape migration level 2 volume. If the data set is recalled, but might be eligible for reconnection during a subsequent migration, this value contains the size, in 16 KB blocks, that the data set would be on the tape migration level 2 volume. This field does not apply to DASD migration volumes, or cloud..
LAST MIG VOLUME	LAST MIGVOL	This field contains the volume serial number of the last tape migration level 2 volume if the data set spans more than one tape volume. *NONE* indicates that the data set does not span more than one tape volume. This field does not apply to DASD.
	CLOUD	SMS Cloud construct name or CDA provider file for target cloud storage.
	CONTAINER	Name of container where the migration copy was placed.
	OBJECT PREFIX	Pseudo-directory name for the migration copy. It is a common unique prefix for all objects that make up a migration copy.
ENCRYPTION TYPE	ENCRYPTION TYPE	This data set is encrypted using the type: '0100'X - AES-256 XTS protected key. It will not be displayed if the data set is not encrypted and the field will contain only ****...
KEYLABEL	KEYLABEL	This data is encrypted with the key label displayed. It will not be displayed if the data set is not encrypted and the field will contain only ****...
A field containing only **** is not applicable (see individual field descriptions in the sample lists).		

Figure 121 on page 1354 is a sample of a printer list from the MCDS for all data sets. If you request information for a specific data set, the list contains only the entry for that data set.

Data sets

```

-----DFSMSHsm CONTROL DATASET - MIGRATED DATA SET-- LISTING ----- AT 10:24:20 ON 15/06/02 FOR SYSTEM=381A

DATASET NAME                                MIG  LAST REF MIGRATED  TRKS    QTY  TIMES DS SDSP  QTY  LAST MIG
VOLUME      DATE      DATE      ALLOC   2K BLKS MIG  ORG DS  16K BLKS VOLUME

G834921.RRDS.N.F40EX001.CLUSTER3           MIG101 15/04/02 11/02/02 00000020 00000006 00001 VS NO *****
ENCRYPTION TYPE = '0100'X KEYLABEL = 123456789012345678921234567893123456789512345678961234
G834921.RRDS.N.F40EX001.CLUSTER4           MIG101 15/04/02 11/02/02 00000020 00000006 00001 VS NO *****
ENCRYPTION TYPE = ***** KEYLABEL = *****

```

Figure 121. Sample Printer List when You Specify DATASETNAME and MIGRATIONCONTROLDATASET

To get a list of the VSAM data sets from the MCDS, specify the LIST command with the DATASETNAME SELECT(VSAM) parameters. The list of VSAM data sets contains the same header information and a list of object names by which you can automatically recall the data set.

Figure 122 on page 1354 is a sample printer list from the MCDS for VSAM data sets when you specify the DATASETNAME SELECT(VSAM) and MIGRATIONCONTROLDATASET parameters.

```

----- DFSMSHsm CONTROL DATASET - MIGRATED DATA SET-- LISTING ----- AT 10:22:08 ON 15/05/02 FOR SYSTEM=381A

DATASET NAME                                MIG  LAST REF MIGRATED  TRKS    QTY  TIMES DS SDSP  QTY  LAST MIG
VOLUME      DATE      DATE      ALLOC   2K BLKS MIG  ORG DS  16K BLKS VOLUME

G834921.KSDS.N.F40EX001.CLUSTER3           MIG101 15/02/02 15/03/02 00000020 00000006 00001 VS NO *****
ENCRYPTION TYPE = '0100'X KEYLABEL = 123456789012345678921234567893123456789512345678961234
BASE D OBJECT NAME = G834921.KSDS.N.F40EX001.DATA3
BASE I OBJECT NAME = G834921.KSDS.N.F40EX001.INDX3

G834921.KSDS.N.F40EX001.CLUSTER4           MIG101 15/02/02 15/03/02 00000020 00000006 00001 VS NO *****
ENCRYPTION TYPE = ***** KEYLABEL = *****
BASE D OBJECT NAME = G834921.KSDS.N.F40EX001.DATA4
BASE I OBJECT NAME = G834921.KSDS.N.F40EX001.INDX4

```

Figure 122. Sample Printer List of VSAM Data Sets when You Specify DATASETNAME, SELECT(VSAM), and MIGRATIONCONTROLDATASET

Table 82 on page 1354 explains the output labels when you specify the DATASETNAME or LEVEL(*qualifier*) parameter. If you want only a summary list, specify the SUMMARY parameter of the LIST command.

Table 82. Output labels when you request a summary of the data sets in the MCDS		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
MIGRATED DATA SETS	MIGRATED DATA SETS	This field contains the number of data sets listed.
TRACKS MIGRATED	TRACKS	This field contains the sum of the tracks allocated for all the data sets listed (see TRKS ALLOC in Table 81 on page 1352).
BYTES MIGRATED	BYTES	This field contains the number of user bytes of data that migrated. It can be represented in units of K (1 024 bytes), M (1 048 576 bytes), G (1 073 741 824 bytes), or T (1 099 511 627 776 bytes).

Figure 123 on page 1354 is a sample printer list of data sets when you specify DATASETNAME and SUMMARY parameters.

```

- DFSMSHSM CONTROL DATASET - SUMMARY-- LISTING ----- AT 12:49:13 ON 01/04/15 FOR SYSTEM=3090

MIGRATED          TRACKS          BYTES
DATA SETS        MIGRATED        MIGRATED

0000000012K      0000000218K      00001429K

----- END OF - MIGRATED DATASET - LISTING -----

```

Figure 123. Sample Printer List when You Specify DATASETNAME and SUMMARY

Requesting data set information from the BCDS

Table 83 on page 1355 explains the output labels for a list of all data sets, all data sets that have the same first qualifier, a specific data set, or list of any copy pools that a volume is a part of.

Table 83. Output labels when you request data set information from the BCDS		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DSNAME	DSN	This field indicates the data set name of the data set that was backed up.
BACKUP FREQ	BACK FREQ	This field indicates the number of days that must elapse after DFSMSHsm backs up a data set before DFSMSHsm can automatically back up the data set again. "****" indicates that the data set is SMS-managed.
MAX ACTIVE BACKUP VERSIONS	MAX VERS	This field indicates the maximum number of backup versions of the data set that DFSMSHsm is to keep. If more than one data set has backup versions with the same data set name, DFSMSHsm can list more than the maximum number of backup versions. "****" indicates that the data set is SMS-managed.
BACKUP VERSION DATA SET NAME	BDSN	This field indicates the data set name of the backup version.
BACKUP VOLUME	BACKVOL	This field indicates the volume serial number of the volume the backup version is on. The volume can be a backup volume or a migration level 1 volume.
FROM VOLUME	FRVOL	This field indicates the volume serial number of the primary volume where the data set is when DFSMSHsm makes the backup version. If the data set is migrated at the time of backup, this field contains the volume serial number of the primary volume from which the data set migrates.
BACKUP DATE	BACKDATE	This field indicates the date the backup version was created.
BACKUP TIME	BACKTIME	This field indicates the time the backup version was created.
SYS CAT	CAT	This field indicates (YES or NO) that the backup version was made from a cataloged data set.
GEN NMBR	GEN	This field indicates the generation number of the backup version. The most recent backup version is 0, the next most recent is 1, and so forth. "****" indicates that the version is a retained backup copy.
VER NMBR	VER	This field indicates the version number of the backup version. This is an absolute number that uniquely identifies this backup version during its life span. The numbering begins at 1 for the first backup version of the data set. "****" indicates that the version is a retained backup copy.

Table 83. Output labels when you request data set information from the BCDS (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
UNS/ RET	UNS/RET	<p>This field indicates special or exception conditions.</p> <p>Value Meaning</p> <p>UNS indicates that the data set was unserialized when backed up.</p> <p>RET indicates that the version listed is a retired version.</p> <p>U/R indicates an unserialized, retired version.</p> <p>NO indicates a version that is neither unserialized nor retired.</p>
RET DAYS	RETDAYS	<p>This field indicates the RETAINDAYS keyword was active at the time of the backup. The value represents the minimum number of days that DFSMSHsm maintains the backup copy. "*****" indicates that RETAINDAYS was not specified.</p>
	RACF IND	<p>This field indicates (YES or NO) if the RACF indicator was on at the time of backup.</p> <p>Note: The RACF IND field is displayed only when the output destination is TERMINAL.</p>
BACKUP PROF	BACK PROF	<p>This field indicates (YES or NO) if a RACF discrete backup profile exists.</p>

Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description								
NEW NAME		<p>This field consists of three single-character placeholders. The single-character placeholders are explained:</p> <p>First character indicates (Y or N) if NEWNAME was specified at the time of backup.</p> <p>Second character The values of the second character are:</p> <p>Y indicates the data set is a VSAM data set with an associated AIX/PATH and with SPHERE(NO) specified.</p> <p>N indicates the data set is a VSAM data set with an associated AIX/PATH and without SPHERE(NO) specified.</p> <p>* indicates the data set is a non-VSAM or VSAM data set with no associated AIX/PATH.</p> <p>Third character The values of the second character are:</p> <p>Y indicates the data set is a VSAM data set, is uncataloged or migrated, and the backup was processed with the GVCN(YES) option.</p> <p>N indicates the data set is a VSAM data set, is uncataloged or migrated, and the backup was processed with the GVCN(NO) option.</p> <p>* indicates the data set is a non-VSAM or VSAM data set and is cataloged, but not migrated.</p> <p>Note: The NEWNAME field is displayed only when the output destination is OUTDATASET or SYSOUT.</p>								
	NEWNM	<p>This field indicates (YES or NO) if NEWNAME was specified at the time of backup.</p> <p>Note: The NEWNM field is displayed only when the output destination is TERMINAL.</p>								
	NOSPH	<p>This field indicates:</p> <table><thead><tr><th>Value</th><th>Meaning</th></tr></thead><tbody><tr><td>YES</td><td>if the data set is a VSAM data set with an associated AIX/PATH and with SPHERE(NO) specified.</td></tr><tr><td>NO</td><td>if the data set is a VSAM data set with an associated AIX/PATH and without SPHERE(NO) specified.</td></tr><tr><td>***</td><td>if the data set is a non-VSAM or VSAM data set with no associated AIX/PATH.</td></tr></tbody></table> <p>Note: The NOSPH field is displayed only when the output destination is TERMINAL.</p>	Value	Meaning	YES	if the data set is a VSAM data set with an associated AIX/PATH and with SPHERE(NO) specified.	NO	if the data set is a VSAM data set with an associated AIX/PATH and without SPHERE(NO) specified.	***	if the data set is a non-VSAM or VSAM data set with no associated AIX/PATH.
Value	Meaning									
YES	if the data set is a VSAM data set with an associated AIX/PATH and with SPHERE(NO) specified.									
NO	if the data set is a VSAM data set with an associated AIX/PATH and without SPHERE(NO) specified.									
***	if the data set is a non-VSAM or VSAM data set with no associated AIX/PATH.									

Table 83. Output labels when you request data set information from the BCDS (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
	GVCN	<p>This field indicates:</p> <p>Value Meaning</p> <p>YES if the data set is a VSAM data set, is uncataloged or migrated, and the backup was processed with the GVCN(YES) option.</p> <p>NO if the data set is a VSAM data set, is uncataloged or migrated, and the backup was processed with the GVCN(NO) option.</p> <p>*** if the data set is a non-VSAM or VSAM data set and is cataloged, but not migrated.</p> <p>Note: The GVCN field is displayed only when the output destination is TERMINAL.</p>
TOTAL BACKUP VERSIONS	TOTAL BACKUP VERSIONS	This field indicates the total number of data set backup versions.
ENCRYPTION TYPE	ENCRYPTION TYPE	This data set is encrypted using the type: '0100'X - AES-256 XTS protected key. It will not be displayed if the data set is not encrypted.
KEYLABEL	KEYLABEL	This data is encrypted with the key label displayed. It will not be displayed if the data set is not encrypted.

Figure 124 on page 1358 is sample printer list output of information for a specific data set when you issue the LIST DATASETNAME(*dsname*) BACKUPCONTROLDATASET command. If you do not specify a data set name, the list will contain information for all data sets.

```
- DFSMSHSM CONTROL DATASET - BACKUP DATASET-- LISTING ----- AT 14:32:30 ON 10/02/25 FOR SYSTEM=3090

DSNAME = 0AYYYYY.SAM.S.DS1                BACKUP FREQ = 000, MAX ACTIVE BACKUP VERSIONS = 002
BACKUP VERSION DATA SET NAME                BACKUP FROM   BACKUP   BACKUP   SYS GEN  VER  UNS/  RET  BACKUP  NEW
VOLUME VOLUME DATE      TIME      CAT  NMBR NMBR RET  DAYS PROF  NAME
DFHSM.BACK.T312914.0AYYYYY.SAM.A0056      A07335 PRIM01 10/02/25 14:29:31 YES 000 002  NO 00200 NO  N**
ENCRYPTION TYPE = '0100'X KEYLABEL = 123456789012345678931234567894123456789512345678961234
DFHSM.BACK.T142814.0AYYYYY.SAM.A0056      A07335 PRIM01 10/02/25 14:28:14 YES 001 001  NO ***** NO  N**
ENCRYPTION TYPE = '0100'X KEYLABEL = 1234567890123456789212345678931234567894123456789512345678961234
TOTAL BACKUP VERSIONS = 0000000002

----- END OF - BACKUP DATASET - LISTING -----
```

Figure 124. Sample Printer List Output for the LIST DATASETNAME(*dsname*) BACKUPCONTROLDATASET Command

Figure 125 on page 1358 is sample terminal output of information for a specific data set when you issue the LIST DATASETNAME(*dsname*) BACKUPCONTROLDATASET command.

```
DSN=0AYYYYY.SAM.S.DS1                BACK FREQ = 000 MAX
ACTIVE BACKUP VERSIONS = 002
BDSN=DFHSM.BACK.T312914.0AYYYYY.SAM.A0056      BACKVOL=A07335
FRVOL=PRIM01
BACKDATE=10/02/25 BACKTIME=14:29:31 CAT=YES GEN=000 VER=002
UNS/RET= NO
RACF IND=NO BACK PROF=NO NEWNM=NO NOSPH=*** GVCN=*** RETDAYS=00200
ENCRYPTION TYPE = '0100'X KEYLABEL = 123456789012345678931234567894123456789512345678961234
BDSN=DFHSM.BACK.T142814.0AYYYYY.SAM.A0056      BACKVOL=A07335
FRVOL=PRIM01
BACKDATE=10/02/25 BACKTIME=14:28:14 CAT=YES GEN=001 VER=001
UNS/RET= NO
RACF IND=NO BACK PROF=NO NEWNM=NO NOSPH=*** GVCN=*** RETDAYS=*****
TOTAL BACKUP VERSIONS = 0000000002
```

Figure 125. Sample Terminal Output for the LIST DATASETNAME(*dsname*) BACKUPCONTROLDATASET Command

Figure 126 on page 1359 is sample printer list output for all data sets with the same first qualifier when you issue the LIST LEVEL(*qualifier*) BACKUPCONTROLDATASET command.


```
- DFSMSHSM CONTROL DATASET - BACKUP DATASET-- LISTING ----- AT 14:43:34 ON 10/02/25 FOR SYSTEM=3090

DSNAME = 0AYYYYY.SAM.S.DS1                BACKUP FREQ = 000, MAX ACTIVE BACKUP VERSIONS = 002
BACKUP VERSION DATA SET NAME              BACKUP FROM  BACKUP  BACKUP  SYS GEN  VER  UNS/ RET  BACKUP NEW
VOLUME VOLUME DATE    TIME    CAT NMBR NMBR RET  DAYS PROF  NAME
DFHSM.BACK.T312914.0AYYYYY.SAM.A0056      A07335 PRIM01 10/02/25 14:29:31 YES 000 002  NO 00200 NO  N**
DFHSM.BACK.T142814.0AYYYYY.SAM.A0056      A07335 PRIM01 10/02/25 14:28:14 YES 001 001  NO ***** NO  N**
TOTAL BACKUP VERSIONS = 000000002

DSNAME = 0AYYYYY.SAM.S.DS2                BACKUP FREQ = 000, MAX ACTIVE BACKUP VERSIONS = 002
BACKUP VERSION DATA SET NAME              BACKUP FROM  BACKUP  BACKUP  SYS GEN  VER  UNS/ RET  BACKUP NEW
VOLUME VOLUME DATE    TIME    CAT NMBR NMBR RET  DAYS PROF  NAME
DFHSM.BACK.T143014.0AYYYYY.SAM.A0056      A07335 PRIM01 10/02/25 14:30:14 YES 000 001  NO 00200 NO  N**
TOTAL BACKUP VERSIONS = 000000001

DSNAME = 0AYYYYY.SAM.S.DS3                BACKUP FREQ = 000, MAX ACTIVE BACKUP VERSIONS = 002
BACKUP VERSION DATA SET NAME              BACKUP FROM  BACKUP  BACKUP  SYS GEN  VER  UNS/ RET  BACKUP NEW
VOLUME VOLUME DATE    TIME    CAT NMBR NMBR RET  DAYS PROF  NAME
DFHSM.BACK.T403014.0AYYYYY.SAM.A0056      A07335 PRIM01 10/02/25 14:30:40 YES 000 001  NO 00100 NO  N**
TOTAL BACKUP VERSIONS = 000000001

----- END OF - BACKUP DATASET - LISTING -----
```

Figure 126. Sample Printer List Output for the LIST LEVEL(qualifier) BACKUPCONTROLDATASET Command

Figure 127 on page 1359 is sample terminal output for all data sets with the same first qualifier when you issue the LIST LEVEL(qualifier) BACKUPCONTROLDATASET command.

```
DSN=0AYYYYY.SAM.S.DS1                BACK FREQ = 000 MAX
ACTIVE BACKUP VERSIONS = 002

  BDSN=DFHSM.BACK.T312914.0AYYYYY.SAM.A0056    BACKVOL=A07335
FRVOL=PRIM01
BACKDATE=10/02/25 BACKTIME=14:29:31 CAT=YES GEN=000 VER=002
UNS/RET= NO
RACF IND=NO BACK PROF=NO NEWNM=NO NOSPH=*** GVCN=*** RETDAYS=00200

  BDSN=DFHSM.BACK.T142814.0AYYYYY.SAM.A0056    BACKVOL=A07335
FRVOL=PRIM01
BACKDATE=10/02/25 BACKTIME=14:28:14 CAT=YES GEN=001 VER=001
UNS/RET= NO
RACF IND=NO BACK PROF=NO NEWNM=NO NOSPH=*** GVCN=*** RETDAYS=*****
TOTAL BACKUP VERSIONS = 000000002

DSN=0AYYYYY.SAM.S.DS2                BACK FREQ = 000 MAX
ACTIVE BACKUP VERSIONS = 002

  BDSN=DFHSM.BACK.T143014.0AYYYYY.SAM.A0056    BACKVOL=A07335
FRVOL=PRIM01
BACKDATE=10/02/25 BACKTIME=14:30:14 CAT=YES GEN=000 VER=001
UNS/RET= NO
RACF IND=NO BACK PROF=NO NEWNM=NO NOSPH=*** GVCN=*** RETDAYS=00200
TOTAL BACKUP VERSIONS = 000000001

DSN=0AYYYYY.SAM.S.DS3                BACK FREQ = 000 MAX
ACTIVE BACKUP VERSIONS = 002

  BDSN=DFHSM.BACK.T403014.0AYYYYY.SAM.A0056    BACKVOL=A07335
FRVOL=PRIM01
BACKDATE=10/02/25 BACKTIME=14:30:40 CAT=YES GEN=000 VER=001
UNS/RET= NO
RACF IND=NO BACK PROF=NO NEWNM=NO NOSPH=*** GVCN=*** RETDAYS=00100
TOTAL BACKUP VERSIONS = 000000001
```

Figure 127. Sample Terminal Output for the LIST LEVEL(qualifier) BACKUPCONTROLDATASET Command

Requesting data set information from the MCDS and BCDS

To get a list of data set information from the MCDS and BCDS, specify the LIST command with the DATASETNAME BOTH parameters. To get a list of the information from the MCDS and BCDS for a specific data set specify the LIST command with the DATASETNAME(dsname) BOTH parameters.

The information is provided in separate lists. In each list, the data sets are in alphanumeric sequence by data set name.

Figure 128 on page 1360 is a sample printer list of all data sets when you specify the DATASETNAME, BOTH, and INCLUDEPRIMARY parameters. If you request information for a specific data set, the list contains only the entries for that data set.

Data sets

```

- DFSMSHSM CONTROL DATASET - MIGRATED DATASET-- LISTING ----- AT 12:49: 13 ON 01/04/05 FOR SYSTEM=3090

```

DATASET NAME	MIGRATED ON VOLUME	LAST REF DATE	MIGRATED DATE	TRKS ALLOC	QTY 2K BLKS	TIMES MIG	DS ORG	SDSP DS	QTY 16K BLKS	LAST MIG VOLUME
D324711.ESDS.R.F40RP123.CLUSTER2	A00015	00/08/27	00/08/27	000016	*****	002	VS	NO	000000	*NONE*
HSMATH0.GDGDS.N10.GDG.G0001V00	ONLINE	91/01/16	91/01/16	000019	0000000	005	PS	NO	*****	*****
HSMATH0.GDGDS.N10.GDG.G0002V00	ONLINE	91/01/18	91/01/18	000019	0000000	001	PS	NO	*****	*****
HSMATH0.GDG5.G0001V00	ONLINE	91/01/20	91/01/20	000002	0000000	004	PS	YES	*****	*****
HSMATH0.GDG5.G0002V00	ONLINE	00/00/00	00/00/00	000000	0000000	001	PS	NO	*****	*****
MRECN.DS1	ONLINE	91/12/30	90/12/30	000001	0000000	004	PS	NO	*****	*****
OW46279.NONVSAM	ONLINE	91/01/01	90/01/01	000001	0000000	004	PS	NO	*****	*****
OW46279.PO	A00001	00/10/09	00/10/10	000015	*****	002	P0	NO	000000	*NONE*
TEST.BVR	WORK01	00/08/18	01/04/02	000001	0000001	001	PS	NO	*****	*****
TEST.SMS	A00005	00/10/31	00/10/31	000015	*****	001	PS	NO	000014	*NONE*

```

- DFSMSHSM CONTROL DATASET - SUMMARY-- LISTING -----AT 12:49:13 ON 01/04/05 FOR SYSTEM=3090

```

MIGRATED DATA SETS	TRACKS MIGRATED	BYTES MIGRATED
000000000K	000000001K	00000429K

```

-----END OF - MIGRATED DATASET - LISTING -----

```

```

DSNAME = H952762.PSFB.F40LI001.DSET01

```

BACKUP VERSION DATA SET NAME	BACKUP VOLUME	FROM VOLUME	BACKUP DATE	BACKUP TIME	SYS CAT	GEN NMBR	VER NMBR	UNS/ RET	RACF IND	BACKUP PROF	NEW NAME
HSM40.BACK.T155911.H952762.PSFB.I1024	MIG101	PRIM01	01/01/24	11:59:15	YES	000	002	U/R	NO	NO	NO
HSM40.BACK.T594911.H952762.PSFB.I1024	BATP01	PRIM01	01/01/24	11:49:59	YES	001	001	NO	NO	NO	NO

```

DSNAME = M059259.PSFB.N.F40RL022.DSET02

```

BACKUP VERSION DATA SET NAME	BACKUP VOLUME	FROM VOLUME	BACKUP DATE	BACKUP TIME	SYS CAT	GEN NMBR	VER NMBR	UNS/ RET	RACF IND	BACKUP PROF	NEW NAME
HSM40.BACK.T034511.M059259.PSFB.I1333	BACK01	PRIM02	01/11/29	11:45:02	NO	000	001	NO	NO	NO	NO

```

DSNAME = M100222.KSDS.R.F40RP125.DSET01

```

BACKUP VERSION DATA SET NAME	BACKUP VOLUME	FROM VOLUME	BACKUP DATE	BACKUP TIME	SYS CAT	GEN NMBR	VER NMBR	UNS/ RET	RACF IND	BACKUP PROF	NEW NAME
HSM40.BACK.T033011.M100222.KSDS.I1025	BACK01	PRIM01	01/01/25	11:30:03	YES	000	003	NO	NO	NO	NO
HSM40.BACK.T593011.M100222.KSDS.I1024	BACK01	PRIM01	01/01/24	11:30:59	YES	001	002	NO	NO	NO	NO
HSM40.BACK.T312911.M100222.KSDS.I1023	BACK01	PRIM01	01/01/23	11:29:32	YES	002	001	UNS	NO	NO	NO

```

-----END OF - BACK-UP DATASET - LISTING -----

```

Figure 128. Sample Printer List of All Data Sets when You Specify DATASETNAME, BOTH, and INCLUDEPRIMARY

Requesting data set information for the latest backup version

Table 84 on page 1360 explains the output labels contained in a list of cataloged data sets whose latest backup version is on the specified backup volume.

Table 84. Output labels when you request data set information for the latest backup version		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DATA SET NAME	DSN	This field contains the original data set name of the backup version.
FROM VOLUME	FRVOL	This field contains the volume serial number of the primary volume where the data set is when DFSMSHsm makes the backup version. If the data set is migrated at the time of backup, this field contains the volume serial number of the primary volume from which the data set migrates.
BACKUP DATE	BACKDATE	This field contains the date the backup version has been created.
BACKUP TIME	BACKTIME	This field contains the time the backup version has been created.
UNS/ RET	UNS/RET	This field indicates special or exception conditions. UNS indicates that the data set was unserialized when backed up. RET indicates that the version listed is a retired version. U/R indicates an unserialized, retired version. NO indicates a version that is neither unserialized nor retired.

Table 84. Output labels when you request data set information for the latest backup version (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
RET DAYS	RD	This field contains the RETAINDAYS parameter specified at the time of backup. The value represents the minimum number of days that DFSMSHsm maintains the backup copy. *** indicates that RETAINDAYS was not specified.

Figure 129 on page 1361 is a sample of a printer list having specified the LIST command with the DATASETNAME and SELECT(VOLUME(volser)) parameters.

```
- DFSMSHSM CONTROL DATASET - LATEST VERSION BACK01-- LISTING ----- AT 12:51:21 ON 09/01/22 FOR SYSTEM=3090

DATA SET NAME                                FROM   BACKUP   BACKUP   UNS/   RET
VOLUME                                         DATE    TIME    RET    DAYS

HSMATH0.SMSDS4.N01.KSDS                      SMS001  09/01/22  12:49:32 NO     00999
HSMATH0.SMSDS4.N04.PSFB                      SMS001  09/01/22  12:49:58 NO     00034

----- END OF - LATEST VERSION BACK01 - LISTING -----
```

Figure 129. Sample Printer List when you specify DATASETNAME and SELECT(VOLUME(volser))

Figure 130 on page 1361 is a sample of a list having specified the LIST command with the DATASETNAME, TERMINAL, and SELECT(VOLUME(volser)) parameters.

```
DSN= HSMATH0.SMSDS4.N04.PSFB                      BACKVOL=BACK01
FRVOL=SMS001 BACKDATE=09/01/22 BACKTIME=12:49:58 UNS/RET= NO RETDAYS=00034

ARC0140I LIST COMPLETED,4 LINE(S) OF DATA
ARC0140I (CONT.) OUTPUT
```

Figure 130. Sample Terminal List when you specify DATASETNAME and SELECT(VOLUME(volser))

Requesting information for dump classes

To get a list of the dump class information for a specific type of volume managed by DFSMSHsm, specify the LIST command with the DUMPCLASS(class) parameter. Table 85 on page 1361 explains the output labels when requesting dump class information.

Table 85. Output labels when you request dump class information		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DUMP CLASS	CLASS	This field indicates the type of dump class for the unit type.
UNIT TYPE	UNIT	This field indicates the type of unit.
AUTO REUSE	REUSE	This field indicates (YES or Y, NO or N) whether the volume for this dump class is automatically available for reuse.
DATASET RESTORE	RESTORE	This field indicates (YES or Y, NO or N) whether to allow a physical data set restore from a full-volume dump copy for this dump class.
RESET CHANGE	RESET	This field indicates (YES or Y, NO or N) whether the change bit for each data set is to be reset by DFSMSDss following a successful full-volume dump.
CLASS DISABLE	DISABLE	This field indicates (YES or Y, NO or N) whether the dump class is disabled.

Table 85. Output labels when you request dump class information (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
CP REQ	REQD	For a copy pool related dump, this field indicates whether the dump class is required or optional. A value of "***" (or "*" in terminal output) indicates that a value was not specified. This field has no significance for a noncopy pool dump.
FRR AVA	AVAILABLE	For a copy pool recovery, this field indicates (YES or Y, NO or N) whether the tapes related to this dump class are available to be mounted. This field has no significance for noncopy pool recoveries.
DAY	DAY	This field indicates the day in the dump cycle. This can be "***".
FREQ	FREQ	This field indicates the minimum number of days that must elapse between the last volume dump to this class and the next automatic dump.
RETPD	RETPD	This field indicates the number of days dump copies for this dump class must be retained. This can be *NOLIM.
TAPE EXPDT	TAPEEXPDT	This field indicates the expiration date for the dump copies created in this dump class. This can be "*****".
VTOC COPIES	VTOCOPIES	This field indicates the number of VTOC copies.
DISPOSITION	DISP	This field indicates the disposition of the dump volume of this unit type. This can be "*****".
HWCMP	HWCMP	This field indicates (YES or NO) whether the data will be compressed before it is encrypted.
ENCRYPT	ENCRYPT	This field indicates the type of encryption to be performed (NONE, RSA, KEYPASSWORD, or ICOUNT).
ENCTYPE	ENCTYPE	This field indicates the type of encryption to be performed (CLRAES128, CLRTDES, or ENCTDES).
ICOUNT	ICOUNT	This field indicates the number of hash iterations to perform on the password to generate the key.
RSAKEY/KPWD	ENC_KEY	This field indicates either the RSA key label, or the password used to encrypt the dump.
RECOVER RESET	RECRST	<p>This field indicates whether or not DFSMSdss resets (sets to OFF) the data-set-changed indicator (DS1DSCHA) in the VTOC entry for all data sets that are restored during full volume recover processing.</p> <p>Value Meaning</p> <p>YES The indicator is reset.</p> <p>NO The indicator is not reset and retains the value the data sets had when dumped.</p> <p>DUMP If the RESET keyword was specified in the dump class at the time the dump was created, the indicator is reset. If either RESET was not specified, NORESET was specified or the dump copy was created prior to z/OS DFSMSHsm V2R1, the indicator is not reset and retains the value it had when the dumped was created.</p>

Table 85. Output labels when you request dump class information (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
MINSTACK	MINSTACK	This field indicates the minimum number of dump copies that should be stacked onto a tape before starting another dump task with this dump class. If not specified, the MAXSTACK specified or defaulted value will be used for MINSTACK.
STACK	STACK	This field indicates the maximum number of dump copies to be written to a dump volume in this dump class. The value is set from the STACK or its alias MAXSTACK keyword on the DEFINE DUMPCLASS command.
DUMP DEVCAT	DUDEVCAT	Value Meaning CLOUD Indicates the dumps in the dump class reside in the cloud object storage. TAPE Indicates the dumps in the dump class reside on tapes.
CLOUD	CLOUD	This field contains the name of the SMS cloud network connection construct or the CDA provider file.

Figure 131 on page 1363 is a sample printer list of the dump classes when you specify the DUMPCLASS parameter. If you specify the parameter DUMPCLASS(class), only dump volumes of class are listed.

```

---- DFSMSHsm CONTROL DATASET -DUMP CLASS-BCDS--- LISTING ----- AT 10:14:52 ON 14/03/19 FOR SYSTEM=3090

DUMP   UNIT   AUTO   DATASET  RESET  CLASS  CP  FRR   TAPE  VTOC
CLASS  TYPE    REUSE  RESTORE  CHANGE DISABLE REQ AVA DAY  FREQ  RETPD  EXPDT  COPIES  DISPOSITION
0DCLAS10 3480    NO     NO       NO      NO    ** YES ** 007  000030  *****  ***  *****

      HWCOMP ENCRYPT ENCTYPE  ICOUNT  RSAKEY/KPWD
      NO     NONE  *****  *****  *****

      RECOVER
      RESET  ZCOMP MINSTACK  STACK
      YES    NO    100      255
----- END OF - DUMP CLASS - LISTING -----

```

Figure 131. Sample Printer List of Dump Classes of Dump Volumes when You Specify DUMPCLASS

Figure 132 on page 1363 is a sample printer list of dump classes when you specify the DUMPCLASS parameter.

```

--- DFSMSHSM CONTROL DATASET -DUMP CLASS-BCDS--- LISTING --- AT 15:55:25 ON 18/10/12 FOR SYSTEM=3090

DUMP   UNIT   AUTO   DATASET  RESET  CLASS  CP  FRR   TAPE  VTOC
CLASS  TYPE    REUSE  RESTORE  CHANGE DISABLE REQ AVA DAY  FREQ  RETPD  EXPDT  COPIES  DISPOSITION
DCL0   3490    NO     YES      NO      NO    YES YES ** 000  000030  *****  002  *****

      HWCOMP ENCRYPT ENCTYPE  ICOUNT  RSAKEY/KPWD
      NO     NONE  *****  *****  *****

      RECOVER
      RESET  ZCOMP MINSTACK  STACK  DUMP
      YES    NO    001      001  DEVCAT  CLOUD
                        *****

----- END OF - DUMP CLASS - LISTING -----

```

Figure 132. Sample Printer List of a specific dump class when you specify the DUMPCLASS parameter

Dump volumes

```

--- DFSMSHSM CONTROL DATASET -DUMP CLASS-BCDS--- LISTING --- AT 15:57:45 ON 18/10/12 FOR SYSTEM=3090

DUMP    UNIT    AUTO    DATASET  RESET  CLASS  CP  FRR    TAPE  VTOC
CLASS   TYPE     REUSE   RESTORE  CHANGE DISABLE REQ AVA DAY FREQ  RETPD EXPDT COPIES DISPOSITION
DCL1    ****      NO      NO       NO      NO    YES YES  **  000  000030  *****  000  *****

HWCOMP  ENCRYPT  ENCTYPE  ICOUNT  RSAKEY/KPWD
NO      NONE    *****  *****  *****

RECOVER  RESET  ZCOMP  MINSTACK  STACK  DUMP  CLOUD
NO       NO   NO    001       001   CLOUD CLOUD
                                PRODCL001

----- END OF - DUMP CLASS - LISTING -----

```

Figure 133.

Requesting information for dump volumes

To get a list of the information for a specific dump volume managed by DFSMSHsm, specify the LIST command with the DUMPVOLUME(*volser*) parameter. To get a list of the information for the dump volumes of the requested status managed by DFSMSHsm, specify the LIST command with the DUMPVOLUME parameter without the volume serial number and with a status parameter (AVAILABLE, UNAVAILABLE, EXPIRED, UNEXPIRED, or NORETENTIONLIMIT). The command lists the volumes in alphanumeric sequence by volume serial number.

Table 86 on page 1364 explains the output labels for any dump volume. These fields are listed for each dump volume and for each dump copy stacked on the volume.

Table 86. Output labels when you request information for dump volumes		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DUMP VOLSER	DUMPVOL	This field contains the volume serial number of the dump volume.
VOL STATUS	STAT	<p>This field contains the status of the dump volume. Possible values are:</p> <p>Value Meaning</p> <p>AVAIL An available volume is one that has been added to DFSMSHsm's inventory but never used or that has been used but has since been invalidated and remains under DFSMSHsm ownership.</p> <p>UNAVA An unavailable volume is one that has had its contents invalidated, but was not eligible to be automatically reused.</p> <p>EXPIR An expired volume is one that contains parts of one or more valid dump copies, and has reached or passed its expiration date.</p> <p>UNEXP An unexpired volume is one that contains parts of one or more valid dump copies, and has not reached its expiration date.</p> <p>NORET No retention limit.</p>
UNIT TYPE	UNIT	This field contains the unit type of the dump volume.

Table 86. Output labels when you request information for dump volumes (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
SEQ	FILESEQ	This field contains the number of the file containing the dump copy. If the field contains "****", then the file shown is continued from another dump tape.
SOURCE VOLSER	SOURCEVOL	This field contains the volume serial number of the source for this dump copy.
SMS	SMS	This field indicates whether a volume is SMS-managed. Value Meaning Yes Identifies the volume as SMS-managed No Identifies the volume as non-SMS-managed.
CLASS	CLASS	This field contains the dump class of the dump volume.
DUMPED DATE	DUMPED DATE	This field contains the date of the dump copy.
DUMPED TIME	DUMPED TIME	This field contains the time of the dump copy.
EXP DATE	EXPDATE	This field contains the expiration date of the dump volume.
IDRC	IDRC	Value Meaning Y Volume contains data in the Improved Data Recording Capability format. N Volume does not contain data in the Improved Data Recording Capability format. ** Volume is empty - not assigned the Improved Data Recording Capability format.
LIBRARY	LIBRARY	This field contains the tape library information for the volume. The field can be: Value Meaning library name the library with which the tape volume is associated. *NO LIB* the tape volume is not associated with a tape library. *ERROR** an error occurs when the library information is retrieved. Check the command activity log for additional messages.
PCT FULL	PCT FULL	This field contains the percentage of the dump volume used by all its dump copies. A value of '100' means the last dump copy spans to another volume. A value of '****' means the percentage is unknown.
CP	CP	This field indicates (Y or N) whether the dump volume has at least one source volume that belongs to a Copy Pool.
SET OF DUMP VOLSERS	DUMPVOLS	This field contains a list of volume serial numbers for the tape volumes used to dump the source volume.

Dump volumes

Table 86. Output labels when you request information for dump volumes (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
HWCOMPRESS	HWCOMP	This field indicates (YES or NO) whether the data was compressed before it was encrypted.
ENCRYPT	ENCRYPT	This field contains the encryption method used (RSA, KEYPASSWORD, or NONE).
ENCRYPTION TYPE	ENCTYPE	This field contains the type of encryption performed (CLRAES128, CLRTDES, or ENCTDES). This field only appears in LIST DUMPVOLUME (volser) output.
RSA KEY LABEL/ RSAKEY/KPWD	KEYPASSWORD	This field contains either the RSA key label, or the password used to encrypt the dump. This field only appears in LIST DUMPVOLUME(volser) output.

Figure 134 on page 1366 is a sample printer list of the specified dump volume when you specify the DUMPVOLUME(volser) parameter.

---- DFSMSHsm CONTROL DATASET -DUMP VOLUME-BCDS--- LISTING --- AT 16:18:44 ON 98/09/12 FOR SYSTEM=381A													
DUMP VOLSER A00359	VOL STATUS UNEXP	UNIT TYPE 3490	FILE SEQ 001	SOURCE VOLSER PRIM01	SMS N	CLASS BITON	DUMPED DATE 1998/01/14	DUMPED TIME 16:17:55	EXP DATE 1998/10/12	IDRC N	LIBRARY STORE001	PCT FULL 18	SET OF DUMP VOLSER A00359

ENCTYPE CLRAES128 RSAKEY/KPWD CCA.CRT08.INT.ENC.1024S0F													
----- END OF - DUMP VOLUME - LISTING -----													

Figure 134. Sample Printer List for DUMPVOLUME(volser)

Figure 135 on page 1366 is a sample printer list for all available volumes when you specify the SELECT(AVAILABLE) parameters.

---- DFSMSHsm CONTROL DATASET -DUMP VOLUME-BCDS--- LISTING --- AT 16:18:52 ON 98/09/12 FOR SYSTEM=381A													
DUMP VOLSER A00400	VOL STATUS AVAIL	UNIT TYPE 3590	SOURCE SEQ **	SOURCE VOLSER *****	SMS *	CLASS BITON	DUMPED DATE *****	DUMPED TIME **:*:*	EXP DATE *****	IDRC **	LIBRARY STORE001	PCT FULL ***	SET OF DUMP VOLSER ***
A00423	AVAIL	3590	**	*****	*	T03	*****	**:*:*	*****	**	STORE001	***	***
A00500	AVAIL	3490	**	*****	*	T100	*****	**:*:*	*****	**	*NO LIB*	***	***
A00618	AVAIL	3490	**	*****	*	T100	*****	**:*:*	*****	**	*NO LIB*	***	***
----- END OF - DUMP VOLUME - LISTING -----													

Figure 135. Sample Printer List for all Available Volumes when You Specify SELECT(AVAILABLE)

Figure 136 on page 1366 is a sample printer list of all dump volumes when you specify the DUMPVOLUME parameter without (volser).

-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING --- AT 06:06:26 ON 23/06/16 FOR SYSTEM=3090													
DUMP VOLSER A00359	VOL STATUS UNEXP	UNIT TYPE 3490	FILE SEQ 001	SOURCE VOLSER PRIM01	SMS N	CLASS BITON	DUMPED DATE 2023/07/14	DUMPED TIME 16:17:55	EXP DATE 2023/12/31	IDRC N	LIBRARY STORE001	PCT FULL 22	SET OF DUMP VOLSER A00359

A00100	AVAIL	3490	002	PRIM02	N	BITON	2023/06/15	04:10:30	*****	N	STORE001	***	A00359
A01000	UNEXP	3590	001	SMS001	Y	T01	2023/06/15	04:12:15	2023/12/31	N	STORE200	100	A00359

A01012	UNEXP	3590	001	SMS003	Y	T01	2023/06/15	04:14:20	2023/12/31	N	STORE200	5	A01000
													A01012

Figure 136. Sample Printer List of all Dump Volumes when You Specify DUMPVOLUME without (volser)

Requesting dump content information for dump volumes

Table 87 on page 1367 explains the dump content output labels for dump volumes. These fields are listed for each dump volume and for each dump copy that is stacked on the volume.

Table 87. Output labels when you request dump content information for dump volumes

Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DUMP VOLSER	DUMPVOL	This field contains the volume serial number of the dump volume. If this field contains %CLOUD , it indicates cloud object storage for dump copies.
VOL STATUS	STAT	This field contains the status of the dump volume. Possible values are: Value Meaning AVAIL An available volume is one that has been added to DFSMSHsm's inventory but never used or that has been used but has since been invalidated and remains under DFSMSHsm ownership. UNAVA An unavailable volume is one that has had its contents invalidated, but was not eligible to be automatically reused. EXPIR An expired volume is one that contains parts of one or more valid dump copies, and has reached or passed its expiration date. UNEXP An unexpired volume is one that contains parts of one or more valid dump copies, and has not reached its expiration date. NORET No retention limit.
UNIT TYPE	UNIT	This field contains the unit type of the dump volume.
SEQ	FILESEQ	This field contains the number of the file containing the dump copy. If the field contains '***', then the file shown is continued from another dump tape.
SOURCE VOLSER	SOURCEVOL	This field contains the volume serial number of the only dump copy on the volume, or the dump copy specified with DUMPCONTENTS(srcvol).
SMS	SMS	This field indicates whether a volume is SMS-managed. Value Meaning Yes Identifies the volume as SMS-managed No Identifies the volume as non-SMS-managed.
CLASS	CLASS	This field contains the dump class of the dump volume.
DUMPED DATE	DUMPED DATE	This field contains the date of the dump copy.
DUMPED TIME	DUMPED TIME	This field contains the time of the dump copy.
EXP DATE	EXPDATE	This field contains the expiration date of the dump volume.

Dump volumes

Table 87. Output labels when you request dump content information for dump volumes (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
IDRC	IDRC	<p>Value</p> <p>Meaning</p> <p>Y Volume contains data in the Improved Data Recording Capability format.</p> <p>N Volume does not contain data in the Improved Data Recording Capability format.</p> <p>** Volume is empty - not assigned the Improved Data Recording Capability format.</p>
LIBRARY	LIB	<p>This field contains the tape library information for the volume. The field can be:</p> <p>Value</p> <p>Meaning</p> <p>library name the library with which the tape volume is associated.</p> <p>*NO LIB* the tape volume is not associated with a tape library.</p> <p>*ERROR** an error occurs when the library information is retrieved. Check the command activity log for additional messages.</p>
PCT FULL	PCT FULL	This field contains the percentage of the dump volume used by all its dump copies. A value of '100' means the last dump copy spans to another volume. A value of '***' means that the percentage is unknown.
SET OF DUMP VOLSER	DUMPVOLS	<p>This field contains a list of volume serial numbers for the tape volumes that are used to dump the source volume.</p> <p>Note: Only one VTOC copy data set is kept for a given dump copy. The above entries are given for a single volume serial number. The following entries are repeated once for each data set that has been dumped to the dump volume. The values shown represent the data set names as they appear in the data set VTOC entry for the data sets.</p>
SOURCE VOLUME	SOURCEVOL	This field contains the volume serial number of the source volume.
DUMP COPY DATA SET	DUMP COPY DATA SET	<p>This field contains the data set name of the dump copy where the contents are listed below.</p> <p>If DUMP VOLSER = %CLOUD, this field contains the object prefix which is the pseudo-directory name for the full-volume dump copy residing in cloud object store. It is a common unique prefix for all objects that make up a full-volume dump copy.</p>
CLOUD	CLOUD	The name that is defined in the SMS cloud network connection construct or the CDA provider file where the full-volume dump copies reside.
CONTAINER	CONTAINER	This field contains the name of the container.
DATASET NAME	DSN	This field contains the list of data sets on the dumped volume.

Table 87. Output labels when you request dump content information for dump volumes (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
ORG	ORG	<p>This field contains the data set organization on the dumped volume. Possible values are:</p> <p>Value Meaning</p> <p>DA BDAM</p> <p>PE Partitioned data set extended</p> <p>PO Partitioned organization</p> <p>PS Physical sequential</p> <p>VS VSAM</p> <p>** Unknown data set organization</p>
MULTI	MULTI	<p>This field contains the non-VSAM indicator for determining if a data set is a multivolume data set. Possible values are:</p> <p>Value Meaning</p> <p>Yes Data resides on multiple volumes</p> <p>No Data resides on a single volume</p> <p>*** VSAM data sets whether multivolume, single volume, or VTOC index.</p>
CREATED	CREATE	This field contains the creation date of the data set on the dumped volume.
REFERENCED	REF	This field contains the date the data set was last referred to.
EXP DATE	EXP	This field contains the expiration date of the data set. This may be **NONE** .
RACF	RACF	<p>This field indicates whether the data set is RACF protected.</p> <p>Note: If the data set is protected with a generic profile, this field indicates NO. It is only an indication of VTOC information.</p>
PSWD	PSWD	This field indicates whether the data set is password protected.
CHANGED	CHANGED	<p>This field indicates whether the data set has been opened for something other than input since the last backup copy was made.</p> <p>*** Is presented for a VTOC index.</p>
The list data is obtained from the format-1 label. For some data set VTOC entry types, not all of the above fields are maintained.		

Figure 137 on page 1370 is a sample printer list of dump content information for a specific dump volume when you specify the DUMPVOLUME(*volser*) and DUMPCONTENTS(*srcvol*) parameters.

Dump volumes

---- DFSMSHsm CONTROL DATASET -DUMP VOLUME-BCDS--- DCONTENTS --- AT 16:18:31 ON 98/09/12 FOR SYSTEM=381A

DUMP VOL UNIT SOURCE DUMPED DUMPED EXP DATE IDRC LIBRARY PCT SET OF DUMP									
VOLSER STATUS TYPE SEQ VOLSER SMS CLASS DATE TIME 1998/12/31 N STORE200 100 VOLSERS									
A01000 UNEXP 3590 001 PRIM02 NO 1998/02/20 04:10:30 1998/12/31 N STORE200 100 A01000 *****									
DUMP COPY DATA SET NAME = DFHSM.DMP.T551706.T01.D98255.VPRIM02									
CONTENTS OF VTOC COPY FOR SOURCE VOLUME PRIM02									
DATASET NAME									
HSMATH0.0W27973T.N01.PSFB	PS	NO	91/09/12	91/09/12	00/00/00	NO	NO	YES	
HSMATH0.0W27973T.N02.PSFB	PS	NO	91/09/12	91/09/12	00/00/00	NO	NO	YES	
HSMATH0.0W27973T.N03.PSFB	PS	NO	91/09/12	91/09/12	00/00/00	NO	NO	YES	
HSMATH0.0W27973T.N04.PSFB	PS	NO	91/09/12	91/09/12	00/00/00	NO	NO	YES	
HSMATH0.0W27973T.N05.PSFB	PS	NO	91/09/12	91/09/12	00/00/00	NO	NO	YES	
HSMATH0.0W27973T.N06.ESDS.DATA	VS	***	91/09/12	91/09/12	99/12/31	***	***	YES	
HSMATH0.0W27973T.N07.ESDS.DATA	VS	***	91/09/12	91/09/12	99/12/31	***	***	YES	
HSMATH0.0W27973T.N08.ESDS.DATA	VS	***	91/09/12	91/09/12	99/12/31	***	***	YES	
HSMATH0.0W27973T.N09.ESDS.DATA	VS	***	91/09/12	91/09/12	99/12/31	***	***	YES	
HSMATH0.0W27973T.N10.ESDS.DATA	VS	***	91/09/12	91/09/12	99/12/31	***	***	YES	
SYS1.VVDS.VPRIM02	VS	***	91/09/12	00/00/00	00/00/00	***	***	NO	
DUMP COPY DATA SET NAME = DFHSM.DMP.T551706.BITON.D98255.VPRIM01									
002 SMS001 YES 1998/02/20 04:12:15 A01000 *****									
DUMP COPY DATA SET NAME = DFHSM.DMP.T552012.BITON.D98255.VSMS001									
003 SMS002 YES 1998/02/20 04:13:25 A01000 *****									
DUMP COPY DATA SET NAME = DFHSM.DMP.T551706.BITON.D98255.VSMS002									
004 SMS003 YES 1998/02/20 04:14:20 A01000 A01012									
----- END OF - DUMP VOLUME - LISTING -----									

Figure 137. Sample Printer List when You Specify DUMPVOLUME(volser) and DUMPCONTENTS(srcvol)

Figure 138 on page 1370 is a sample printer list of cloud dump information for a specific source volume when you specify DUMPVOLUME(%CLOUD) and DUMPCONTENTS(srvol) parameters:

-- DFSMSHSM CONTROL DATASET -DUMP VOLUME-BCDS-- LISTING --- AT 17:57:40 ON 22/01/10 FOR SYSTEM=3090

DUMP VOL UNIT FILE SOURCE DUMPED DUMPED EXP DATE IDRC LIBRARY PCT HW ENC C SET OF DUMP													
VOLSER STATUS TYPE SEQ VOLSER SMS CLASS DATE TIME 2021/10/31 ** ***** ** * *** N													
%CLOUD *****	*** SMS062 Y	2021/10/26 01:23:24											
ENCTYPE RSAKEY/KPWD													

DUMP COPY DATA SET NAME = DFHSM.DMP.DCL11.VSMS062.D21299.T242301													
CLOUD = PRODCLOUD01													
CONTAINER = SYSZARC.ARCPLEX.Y.DMP.2021277.A5B1CE55													
CONTENTS OF VTOC COPY FOR SOURCE VOLUME SMS062													
DATASET NAME													
HSMATH1.SC11.SMSDS01.SAM	PS	NO	21/04/28	21/09/22	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS02.SAM	PS	NO	21/04/28	21/09/23	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS03.SAM	PS	NO	21/04/28	21/07/19	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS04.SAM	PS	NO	21/04/28	21/07/19	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS05.SAM	PS	NO	21/04/28	21/07/19	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS15.ESDS.DATA	VS	***	20/08/17	20/08/17	00/00/00	***	***	YES					
HSMATH1.SC11.SMSDS18.KSDS.DATA	VS	***	20/08/17	20/08/17	00/00/00	***	***	YES					
HSMATH1.SC11.SMSDS18.KSDS.INDEX	VS	***	20/08/17	00/00/00	00/00/00	***	***	NO					
SYS1.VTOCIX.SMS062	PS	***	20/07/22	00/00/00	00/00/00	NO	NO	NO					
SYS1.VVDS.VSMS062	VS	***	20/07/22	00/00/00	00/00/00	***	***	NO					
DUMP COPY DATA SET NAME = DFHSM.DMP.DCL11.VSMS062.D21299.T410002													
CLOUD = PRODCLOUD01													
CONTAINER = SYSZARC.ARCPLEX.Y.DMP.2021277.A5B1CE55													
CONTENTS OF VTOC COPY FOR SOURCE VOLUME SMS062													
DATASET NAME													
HSMATH1.SC11.SMSDS01.SAM	PS	NO	21/04/28	21/09/22	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS02.SAM	PS	NO	21/04/28	21/09/23	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS03.SAM	PS	NO	21/04/28	21/07/19	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS04.SAM	PS	NO	21/04/28	21/07/19	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS05.SAM	PS	NO	21/04/28	21/07/19	00/00/00	NO	NO	YES					
HSMATH1.SC11.SMSDS15.ESDS.DATA	VS	***	20/08/17	20/08/17	00/00/00	***	***	YES					
HSMATH1.SC11.SMSDS18.KSDS.DATA	VS	***	20/08/17	20/08/17	00/00/00	***	***	YES					
HSMATH1.SC11.SMSDS18.KSDS.INDEX	VS	***	20/08/17	00/00/00	00/00/00	***	***	NO					
SYS1.VTOCIX.SMS062	PS	***	20/07/22	00/00/00	00/00/00	NO	NO	NO					
SYS1.VVDS.VSMS062	VS	***	20/07/22	00/00/00	00/00/00	***	***	NO					
----- END OF - DUMP VOLUME - LISTING -----													

Figure 138. Sample Printer List of cloud dump information for a specific source volume when you specify DUMPVOLUME(%CLOUD) and DUMPCONTENTS(srvol)

Requesting information for primary and migration volumes

To get a list of information for migration or primary volumes, specify the LIST command with the VOLUME, PRIMARYVOLUME, or MIGRATIONVOLUME parameters. Each parameter gives you the following results:

VOLUME

Specific primary or migration volume

VOLUME(volser)

Specific primary or migration volume

PRIMARYVOLUME

All primary volumes

PRIMARYVOLUME(volser)

Specific primary volume

MIGRATIONVOLUME

All migration volumes

MIGRATIONLEVEL1

All migration level 1 volumes

MIGRATIONLEVEL2

All migration level 2 volumes

MIGRATIONLEVEL2(DASD)

All DASD migration level 2 volumes

MIGRATIONLEVEL2(TAPE)

All tape migration level 2 volumes

When you specify LIST VOLUME, DFSMSHsm prints the list in the following order:

1. Primary volumes
2. Migration level 1 volumes
3. Migration level 2 DASD volumes
4. Migration level 2 tape volumes

When you specify LIST MIGRATIONVOLUME, DFSMSHsm prints the list in the following order:

1. Migration level 1 volumes
2. Migration level 2 DASD volumes
3. Migration level 2 tape volumes

When you request information for migration or primary volumes, you can also request that information be listed from the migration control data set (MCDS), backup control data set (BCDS), or both control data sets. When you specify the MIGRATIONCONTROLDATASET parameter, DFSMSHsm lists information about primary and migration volumes. When you specify the BACKUPCONTROLDATASET parameter, DFSMSHsm lists information about primary volumes that have been backed up.

Requesting primary and migration volume information from the MCDS

Table 88 on page 1371 explains output labels when DFSMSHsm lists information from the MCDS for primary and DASD migration volumes.

Table 88. Output labels when you request information from the MCDS for primary and DASD migration volumes		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
VOLSER	VOL	This field indicates the volume serial number of the primary or migration volume.

Table 88. Output labels when you request information from the MCDS for primary and DASD migration volumes (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DEVICE TYPE	DEVTYPE	This field indicates the name of the unit where this volume can be allocated.
VOLUME TYPE	VOLTYPE	<p>Value Meaning</p> <p>PRIM The volume has been added as a primary volume.</p> <p>P SMS The volume is known as an SMS-managed volume.</p> <p>LEV 1 The volume has been added as a migration level 1 volume.</p> <p>L 1-OV The OVERFLOW subparameter has been specified on the ADDVOL command for the DASD migration level 1 volume.</p> <p>L2-AV The volume has been added as a DASD migration level 2 volume and is still available to be assigned a key range.</p> <p>L2-UN The volume has been added as a DASD migration level 2 volume and has already been assigned a key range.</p> <p>D The DRAIN attribute has been specified on the ADDVOL command for the DASD migration level 1 or level 2 volume. For example: LEV-1 D, L2-AV D, and L2-UN D.</p>
THRESHOLD HI---LOW	HI-THRESH LO-THRESH	This field indicates the high and low threshold of occupancy defined for this volume. The low threshold of occupancy applies only to primary volumes. Values for SMS-managed volumes are from storage group attributes that DFSMSHsm last retrieved from SMS and are not necessarily the most current values.
FRAG INDEX	FRAG	This field indicates the qualitative measure of the scattered free space on a volume. The values of the index can range from 0 to 1. The higher the value, the more fragmented the free space on the volume.
HOSTID AUTO- MIG	AUTO-HOST MIG	This field indicates the ID of the DFSMSHsm host that assigned the primary volume attribute when making the volume eligible for automatic space management (ADDVOL command AUOTMIGRATION parameter). A value of NONE indicates that the primary volume attribute was not assigned by a DFSMSHsm host. SMS-managed volumes and migration level 1 volumes are excluded and this field indicates "*****" or "***". This field applies to primary volumes only.
HOSTID AUTO- BACK	BACK	This field indicates the ID of the DFSMSHsm host that assigned the primary volume attribute when making the volume eligible for automatic backup (ADDVOL command AUTOBACKUP parameter). A value of NONE indicates that the primary volume attribute was not added by a DFSMSHsm host. SMS-managed volumes and migration level 1 volumes are excluded and this field indicates "*****" or "***". This field applies to primary volumes only.

Table 88. Output labels when you request information from the MCDS for primary and DASD migration volumes (continued)

Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
HOSTID AUTO-DUMP	DUMP	This field indicates the ID of the DFSMSHsm host that assigned the primary volume attribute when making the volume eligible for full-volume automatic dump (ADDVOL command AUTODUMP parameter). A value of NONE indicates that the primary volume attribute was not added by a DFSMSHsm host. SMS-managed volumes are excluded and this field indicates "*****" or "***". This field applies to primary and migration level 1 volumes only.
AUTO RECL	AUTO-RECALL	A YES or Y indicates that the volume is eligible to receive data sets that have the primary volume attribute of automatic recall. This field applies only to DFSMSHsm primary volumes. SMS-managed volumes are excluded with asterisks (*).
SDSP AVAIL	SDSP	YES or Y indicates the following: <ul style="list-style-type: none"> • This volume contains a small-data-set-packing data set and • This volume was ADDVOLed with the SDSP parameter and • The SETSYS SMALLDATASETPACKING command was specified This field applies only to migration level 1 volumes.
MIN AGE	MIN-AGE	This field indicates the inactive age of the data set that most recently migrated from the volume during the last volume space management. SMS-managed volumes are excluded with asterisks (*).
MIGRATED DS--TRKS	MIGDS MIGTRKS	This is the number of data sets that has been migrated or deleted from the volume during the last volume space management. Tracks are not applicable for level 0 volumes and are excluded with asterisks (*). Both fields are only applicable for the last VOLUME request, not for a LEVEL request like secondary space management.
DATE---TIME LAST MIGRATED	MIGDATE-TIME	This is the date and time of the last migration from this volume if primary, or to this volume if Level 2 DASD. This field is invalid for Level 1 volumes.
SPACE-MGMT TYPE/AGE	SPACE-MGMT	This is the space management technique assigned to this volume. MIG indicates migration, DBA indicates data set deletion, and DBU indicates data set retirement. Values for SMS-managed volumes are from storage group attributes that DFSMSHsm last retrieved from SMS and are not necessarily the most current values.
SPACE-MGMT TYPE/AGE	AGE	This field applies only to DFSMSHsm primary volumes. SMS-managed volumes are excluded with asterisks (*). This is the number of days a data set on this volume must be inactive before it is eligible for the type of space management indicated under SPACE-MGMT. DEFAULT or DEF indicates that the age criteria is not specified for the volume in the ADDVOL command. In this case, the age criteria is the larger of SETSYS days value and integrity value, which is based on whether DFSMSHsm is running in a single-DFSMSHsm-host environment or in a multiple-DFSMSHsm-host environment.
BUDEV CATGY	BACKUP-DEVICE-CATEGORY	This field indicates the backup device category, tape or DASD, assigned to this volume. If no category has been assigned, the field contains ANY. This field applies only to DFSMSHsm primary volumes. SMS-managed volumes are excluded with asterisks (*).

Table 88. Output labels when you request information from the MCDS for primary and DASD migration volumes (continued)

Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
DUMP CLASS	DUMPCCLASS	This is the dump class of the primary or migration volume. Values for SMS-managed volumes are from storage group attributes that DFSMSHsm last retrieved from SMS and are not necessarily the most current values.
A field containing only **** is not applicable to this volume (see individual field descriptions in the sample lists).		

Figure 139 on page 1374 is a sample printer list of primary volumes when you specify the PRIMARYVOLUME and MIGRATIONCONTROLDATASET

```

- DFSMSHSM CONTROL DATASET - PRIMARY VOLUME-MCDS--- LISTING ----- AT 12:40:59 ON 01/04/05 FOR SYSTEM=3090
VOLSER  DEVICE VOLUME THRESHOLD FRAG HOSTID AUTO AUTO SDSP MIN MIGRATED DATE--TIME SPACE-MGMT BUDEV DUMP
TYPE    TYPE    HI--LOW INDX MIG--BACK-DUMP RECL ELIG AGE DS-----TRKS LAST MIGRATED TYPE/AGE CATGY CLASS
PRIM01  3390  PRIM    100 *** .000 1  NONE NONE YES  *** 000 00001K ***** 01/04/02 00:00 MIG 001 ANY *****
PRIM02  3390  PRIM    100 *** .045 1  NONE NONE YES  *** 000 00000K ***** 01/04/05 10:19 MIG 001 ANY *****
SMS001  3390  P SMS    010 001 .000 **** **** **** ***  *** 00002K ***** 01/04/03 17:00 MIG *** **** *****
SMS002  3390  P SMS    010 001 .303 **** **** **** ***  *** 00001K ***** 01/04/03 17:05 MIG *** **** *****
SMS003  3390  P SMS    010 001 .114 **** **** **** ***  *** 00000K ***** 01/04/03 17:05 MIG *** **** *****
SMS004  3390  P SMS    010 001 .086 **** **** **** ***  *** 00000K ***** 01/04/03 17:06 MIG *** **** *****
----- END OF - PRIMARY VOLUME - LISTING -----

```

Figure 139. Sample Printer List of Primary Volumes from the MCDS when You Specify PRIMARYVOLUME and MIGRATIONCONTROLDATASET

Figure 140 on page 1374 is a sample printer list of DASD migration level 1 volumes when you specify the MIGRATIONLEVEL1 and MIGRATIONCONTROLDATASET parameters. The format of the printer list for a specific primary or DASD migration volume, for all primary and DASD migration volumes, or for all primary volumes or all DASD migration volumes has the same format as that shown in Figure 139 on page 1374 and Figure 140 on page 1374. If you request information for more than one volume, the list has multiple entries.

```

1- DFSMSHSM CONTROL DATASET - MIGRATE VOLUME-MCDS--- LISTING ----- AT 14:09:45 ON 08/12/02 FOR SYSTEM=PCC
VOLSER  DEVICE VOLUME THRESH FRAG HOSTID AUTO AUTO SDSP MIN MIGRATED DATE--TIME SPACE-MGMT BUDEV DUMP
TYPE    TYPE    HI--LOW INDX MIG--BACK-DUMP RECL ELIG AGE DS-----TRKS LAST MIGRATED TYPE/AGE CATGY CLASS
MIG101  3390  LEV 1    075 *** .018 **** **** 8 *** YES *** 00115 009636 00/00/00 00:00 *** *** **** DUMPCLB
MIG102  3390  LEV 1    075 *** .012 **** **** 8 *** NO  *** 00010 000125 00/00/00 00:00 *** *** **** DUMPCLC
MIG103  3390  LEV 1    075 *** .079 **** **** 8 *** NO  *** 00005 000070 00/00/00 00:00 *** *** **** DUMPCLC
OVFML1  3390  L1-OV    080 *** .000 **** **** C *** NO  *** 00012 745930 00/00/00 00:00 *** *** **** DUMPCLB
----- END OF - MIGRATE VOLUME - LISTING -----
1- DFSMSHSM CONTROL DATASET - MIGRATE VOLUME-MCDS--- LISTING ----- AT 14:09:54 ON 08/12/02 FOR SYSTEM=PCC

```

Figure 140. Sample Printer List of Migration Volumes from the MCDS when You Specify MIGRATIONVOLUME and MIGRATIONCONTROLDATASET

Figure 141 on page 1374 is a sample terminal list for a primary volume when you specify the VOLUME, MIGRATIONCONTROLDATASET, and TERMINAL parameters. The format of the terminal list for a specific primary volume or DASD migration volume or for all primary volumes or all DASD migration volumes has the same format as shown in Figure 141 on page 1374, except that the third line is not produced for migration level 1 volumes. If you request information for only one volume, the list has only one entry.

```

VOL=PRIM13  DEVTYP=3390      VOLTYPE=P SMS  MIGDS=0000  MIGTRKS=****
MIGDATE-TIME=00/00/00 00:00  FRAG=.037  HI-THRESH=100  LO-THRESH=***
MIN-AGE=***  BACKUP-DEVICE-CATEGORY=***  AUTO-RECALL=*
SDSP=***  SPACE-MGMT=MIG  AGE=***  AUTO-HOST MIG=**  BACK=**  DUMP=**
DUMPCCLASS = DCLASS01 DCLASS02 DCLASS03 DCLASS04 DCLASS05

```

Figure 141. Sample Terminal List of a Volume from the MCDS when You Specify VOLUME, MIGRATIONCONTROLDATASET, and TERMINAL

Table 89 on page 1375 explains the output labels when DFSMSHsm lists information from the MCDS for tape migration level 2 volumes.

Table 89. Output labels when you request information from the MCDS for tape migration level 2 volumes		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
VOLSER	VOL	This field contains the volume serial number of the tape migration level 2 volume.
DEVICE TYPE	DEVTYPE	This field contains the name of the unit where this volume can be allocated.
MIGRATE TYPE	MIGTYPE	L2-TP: The tape volume has been added as a tape migration level 2 volume.
VOL FULL	VOL FULL	A YES or Y indicates that an end-of-tape marker is reached or a data-movement error occurs while DFSMSHsm is writing on the tape volume.
DATE VOL LAST SP MANAGED	SP-MGT-DATE	This is the date that DBA/DBU was last performed on this volume.
VOL EMPTY	VOL EMPTY	A YES or Y indicates that the volume is empty.
DELETED DS	DELDS	This is the number of data sets that were deleted from the volume during the last volume space management.
PSWD	PSWD	A YES or Y indicates that the tape volume is password protected.
EXP	EXP	A YES or Y indicates that this tape volume is protected by an expiration date.
RACF	RACF	A YES or Y indicates that the tape volume is RACF protected.
AVAILABLE	AVAIL	A YES or Y indicates that the tape volume is available. A NO or N normally results from a reply by an operator indicating that the tape will not be mounted.
IN USE	IN USE	Value Meaning NO The tape volume is not being used. MIGD Data set migration is using the tape volume. MIGV Volume migration is using the tape volume. RECL Recall is using the tape volume. RCYS The tape volume is a recycle source volume. RCYT The tape volume is a recycle target volume. DBAU Data set deletion or data set retirement is using the tape volume.
SELECTED	SELD	A YES or Y indicates that this volume is a target migration tape, is not marked full, and was the last tape selected for a migration function. Issue the QUERY MIGRATION command for each host to determine which DFSMSHsm migration function has selected this volume.

Primary and migration volumes from the MCDS

Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
IDRC	IDRC	<p>Value Meaning</p> <p>Y Volume contains data in the Improved Data Recording Capability format.</p> <p>N Volume does not contain data in the Improved Data Recording Capability format.</p> <p>** Volume is empty—not assigned the Improved Data Recording Capability format.</p>
DUPLEX ALT	DUPLEX ALT	<p>Value Meaning</p> <p>volser indicates the volume serial number of the duplexed alternate.</p> <p>*PEND* indicates that an exception condition occurred and an internal TAPECOPY is pending.</p> <p>*NONE* indicates that the volume was not created in a duplexing environment. Although an alternate copy may exist that was created by TAPECOPY, no duplex alternate exists.</p> <p>NA indicates that the original volumes are DASD rather than tape.</p>
PCT FULL	PCT FULL	This field contains the percentage of the migration volume used for a tape written using 3590 architecture. A value of **** means the percentage is unknown.

Figure 142 on page 1376 is a sample printer list of tape migration level 2 volumes when you specify the `MIGRATIONLEVEL2(TAPE)` and `MIGRATIONCONTROLDATASET` parameters.

-- DFSMSHSM CONTROL DATASET - MIGRATE VOLUME-MCDS--- LISTING ----- AT 09:59:11 ON 01/04/05 FOR SYSTEM=3090																	
VOLSER	DEVICE	VOLUME	THRESH	HOST	AUTO-	MIG-	BACK-DUMP	AUTO	SDSP			MIGRATED	DATE---	TIME	SPACE-MGMT	BUDEV	DUMP
	TYPE	TYPE	HI--LOW	INDX				RECL	ELIG		MIN	DS----	TRKS	LAST MIGRATED	TYPE/AGE	CATGY	CLASS
MIG101	3390	LEV 1	100 ***	.000	****	****	NONE	***	NO	***	00000K	000012K	01/04/02	23:35	***	***	****
MIG102	3390	LEV 1	100 ***	.000	****	****	NONE	***	NO	***	00001K	000023K	01/04/02	23:47	***	***	****
MIG103	3390	LEV 1	100 ***	.000	****	****	NONE	***	NO	***	00001K	000004K	01/04/03	00:34	***	***	****
----- END OF - MIGRATE VOLUME - LISTING -----																	
1- DFSMSHSM CONTROL DATASET - MIGRATE VOLUME-MCDS--- LISTING ----- AT 09:59:11 ON 01/04/05 FOR SYSTEM=3090																	
VOLSER	DEVICE	MIGRATE	VOL	DATE	VOL	LAST	VOL	DELETED	PSWD	EXP	RACF	AVAILABLE	IN	USE	SELECTED	IDRC	DUPLEX
	TYPE	TYPE	FULL	SP	MANAGED		EMPTY	DS									PCT
																	FULL
A00001	3480	L2-TP	NO	00/00/00			NO	0000	YES	NO	NO	YES	-NO-		NO	N	*NONE* 34.8
A00005	3480	L2-TP	NO	00/00/00			NO	0000	YES	NO	NO	YES	-NO-		NO	N	*NONE* 34.8
A00015	3480	L2-TP	NO	00/00/00			NO	0000	YES	NO	NO	YES	-NO-		NO	N	*NONE* 34.8
A00016	3580	L2-TP	YES	00/00/00			NO	0000	YES	NO	NO	YES	-NO-		NO	N	*NONE* 34.8
A00019	3580	L2-TP	NO	00/00/00			NO	0000	YES	NO	NO	YES	-NO-		NO	N	*NONE* 34.8
A00021	3580	L2-TP	NO	00/00/00			NO	0000	YES	NO	NO	YES	-NO-		NO	N	*NONE* 34.8
----- END OF - MIGRATE VOLUME - LISTING -----																	

Figure 142. Sample Printer List of Tape Migration Level 2 Volumes when You Specify `MIGRATIONLEVEL2(TAPE)` and `MIGRATIONCONTROLDATASET`

Requesting primary and migration volume information from the BCDS

[Table 90 on page 1377](#) explains the output labels from the BCDS for the list of migration or primary volumes.

Table 90. Output labels when you request information from the BCDS for primary and migration volumes		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
VOLSER	VOL	This field contains the volume serial number of the volume.
SMS	SMS	This field indicates whether a volume is SMS-managed. Value Meaning Yes Identifies the volume as SMS-managed No Identifies the volume as non-SMS-managed.
OWNED BY catalog	VSAM CTLG	This field contains the data set name of the catalog that currently controls the volume. If the volume is not owned by a VSAM catalog, this field contains NOT OWNED BY catalog. If the controlling catalog is the master catalog, this field contains **MASTER CATALOG**.
CATALOG ON VOLSER	CTLG VOL	This field contains the volume serial number of the volume on which the catalog resides, SYSRES, or blanks.
LAST BACKED UP DATE TIME	LAST BACKED UP ON	This field contains the date and time of the last volume backup for the volume.
DUMPCCLASS	LAST DUMPED TO CLASS	This is the dump class of the primary or migration volume.
DUMPED	ON	This field contains the date of the latest dump in the class.
TIME	AT	This field contains the time of the latest dump in the class.
EXP DATE	EXP DATE	This field contains the expiration date of the latest dump in the class. If RETENTIONPERIOD(NOLIMIT) was specified with the DEFINE DUMPCCLASS or BACKVOL command, the value stored is zero (0) and the value displayed is *NOLIMIT.
A field containing only *** is not applicable to this volume (see individual field descriptions in the sample lists).		

Figure 143 on page 1377 is a sample printer list from the BCDS for a specific primary volume when you specify PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET parameters. The format of the printer list for all primary volumes has the same format as that shown in Figure 143 on page 1377. Similar output is produced for migration level 1 volumes if requested, using the VOLUME, MIGRATIONVOLUME, or MIGRATIONLEVEL1 parameters. If you request information for only one volume, the list has only one entry.

----- DFSMSHsm CONTROL DATASET PRIMARY VOLUME-BCDS --- LISTING ----- AT 13:31:36 ON 91/12/31 FOR SYSTEM=SYSA									
		CATALOG		LAST BACKED UP					
VOLSER	SMS	OWNED BY catalog	ON VOLSER	DATE	TIME	DUMPCCLASS	DUMPED	TIME	EXP DATE
PRIM01	YES	VSAM.CATALOG.PRIM01	PRIM01	91/01/15	00:59	DCLASS01	91/01/31	01:59:05	*NOLIMIT
						DCLASS02	91/01/31	01:59:05	87/12/31
						DCLASS03	91/01/31	01:59:05	87/12/31
						DCLASS04	91/01/31	01:59:05	87/12/31
						DCLASS05	91/01/31	01:59:05	87/12/31
----- END OF - PRIMARY VOLUME - LISTING -----									

Figure 143. Sample Printer List from the BCDS when You Specify PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET

Figure 144 on page 1378 is a sample terminal list from the BCDS for primary volumes when you specify the VOLUME, BACKUPCONTROLDATASET, and TERMINAL. The second line is not produced when the volume is a migration level 1 volume. The format of the terminal list for a specific backup volume is the same as the format shown in Figure 144 on page 1378. If you request information for only one volume, the list has only one entry.


```
VOL=PRIM01, SMS=YES VSAM CTLG=USERCAT.PRIM01
CTLG VOL=PRIM01, LAST BACKED-UP ON 90/04/24 AT 20:49
LAST DUMPED TO CLASS DCLASS01 ON 91/01/31 AT 01:59:05 EXP DATE *NOLIMIT

VOL=PRIM02, SMS=NO VSAM CTLG=USERCAT.PRIM02
CTLG VOL=PRIM02, LAST BACKED-UP ON 90/04/24 AT 20:49
LAST DUMPED TO CLASS DCLASS01 ON 91/01/31 AT 01:59:05 EXP DATE *NOLIMIT
```

Figure 144. Sample Terminal List from the BCDS when You Specify VOLUME, BACKUPCONTROLDATASET, and TERMINAL

Figure 145 on page 1378 is a sample terminal list from the BCDS for primary volumes when you specify the PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET parameters:

```
----- DFSMSshm CONTROL DATASET PRIMARY VOLUME-BCDS --- LISTING -----
AT 13:31:36 ON 20/09/10 FOR SYSTEM=SYSA
VOLSER SMS OWNED BY catalog          CATALOG LAST BACKED UP
SMS062YES VSAM.CATALOG.SMS062      ON VOLSER DATE TIME DUMPCLASS DUMPED TIME EXP DATE
----- END OF - PRIMARY VOLUME - LISTING -----
SMS062  91/01/15 00:59 DCL7CLD  20/06/14 02:20:00 *NOLIMIT
```

Figure 145. Sample Printer List from the BCDS for a Specific Primary Volume with Cloud Dump Classes when You Specify PRIMARYVOLUME(volser), BACKUPCONTROLDATASET

Requesting primary volume dump information from the BCDS

Table 91 on page 1378 explains the output labels when you request dump information from the BCDS for primary volumes.

Table 91. Output labels when you request dump information from the BCDS for primary volumes		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
SOURCE VOLSER	SOURCEVOL	This field contains the volume serial number of the source volume.
GEN	GEN	This field contains relative generation number of the dump for the volume.
SMS	SMS	This field indicates whether a volume is SMS-managed. Value Meaning Yes Identifies the volume as SMS-managed No Identifies the volume as non-SMS-managed.
DUMPED	DATE	This field contains the date of the dump for the volume.
TIME	TIME	This field contains the time of the dump for the volume.
CLASS	DUMPCLASS	This field contains the dump class of the primary volume.
EXP DATE	EXPDATE	This field contains the expiration date of the dump. This might be *NOLIMIT on both the printer listing and on the terminal.
SET OF DUMP VOLSERs / CLOUD INFO	DUMPVOLS	This field indicates the volume serial number of the dump volume. If this field contains %CLOUD, it indicates cloud object storage for dump copies.

Figure 146 on page 1379 is a sample printer list of the tape dump information for a specific primary volume.


```

----- DFSMSHsm CONTROL DATASET PRIMARY VOLUME-BCDS--- ALLDUMPS----- AT 13:31:36 ON 91/12/31 FOR SYSTEM=SYSA
SOURCE
VOLSER GEN SMS DUMPED      TIME      CLASS      EXP DATE      SET OF DUMP
                                VOLSERS / CLOUD INFO
PRIM01  00 YES 91/01/31  01:59:05  DCLASS01  *NOLIMIT  TAP011 TAP012 TAP013 TAP014 TAP015 TAP016 TAP017 TAP018 TAP019
                                TAP01B TAP01C TAP01D TAP01E TAP01F ***** ***** *****
                                01 NO 90/12/31  00:59:05  DCLASS02  91/12/31  TAP021 TAP022 TAP023 TAP024 TAP025 TAP026 TAP027 TAP028 TAP029
                                DCLASS01  91/11/30  TAP001 TAP002 TAP003 TAP004 TAP005 TAP006 TAP007 TAP008 TAP009
----- END OF - PRIMARY VOLUME - LISTING -----

```

Figure 146. Sample Printer List from the BCDS when You Specify ALLDUMPS with PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET

Figure 147 on page 1379 is a sample printer list of the cloud dump information for a specific primary volume when you specify ALLDUMPS with PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET:

```

- DFSMSHSM CONTROL DATASET - PRIMARY VOLUME-BCDS--- ALLDUMPS----- AT 02:57:18 ON 21/11/22 FOR SYSTEM=3090
SOURCE
VOLSER GEN SMS DUMPED      TIME      CLASS      EXP DATE      SET OF DUMP
                                VOLSERS / CLOUD INFO
SMS065  00 YES 21/11/22  02:53:29  DCL12S    21/11/27  %CLOUD PRODCLOUD02
                                SYSZARC.ARCPLEX.Y.DMP.2021277.4914EB65
                                DFHSM.DMP.DCL12S.VSMS065.D21326.T295302
SMS065  01 YES 21/11/22  02:52:14  DCL12     21/11/27  %CLOUD PRODCLOUD02
                                SYSZARC.ARCPLEX.Y.DMP.2021277.4914EB65
                                DFHSM.DMP.DCL12.VSMS065.D21326.T145202
SMS065  02 YES 21/08/19  03:11:54  DCL11     21/08/24  %CLOUD PRODCLOUD01
                                SYSZARC.ARCPLEX.Y.DMP.2021185.81EBEB02
                                DFHSM.DMP.DCL11.VSMS065.D21231.T541103
----- END OF - PRIMARY VOLUME - LISTING -----

```

Figure 147. Sample Printer List of the cloud dump information for a specific primary volume when you specify ALLDUMPS with PRIMARYVOLUME(volser) and BACKUPCONTROLDATASET

Figure 148 on page 1379 is a sample printer list of dump information for a specific primary volume when you specify ALLDUMPS with PRIMARYVOLUME(volser) and BCDS. The primary volume has been dumped to TAPE and CLOUD dump classes at different times.

```

- DFSMSHSM CONTROL DATASET - PRIMARY VOLUME-BCDS--- ALLDUMPS----- AT 13:25:30 ON 22/01/10 FOR SYSTEM=3090
SOURCE
VOLSER GEN SMS DUMPED      TIME      CLASS      EXP DATE      SET OF DUMP
                                VOLSERS / CLOUD INFO
SMS062  00 YES 22/01/10  13:22:53  DCL1      22/02/09  A07551 ***** ***** ***** ***** *****
SMS062  01 YES 21/10/26  02:00:41  DCL11     21/10/31  %CLOUD PRODCLOUD01
                                SYSZARC.ARCPLEX.Y.DMP.2021277.A5B1CE55
                                DFHSM.DMP.DCL11.VSMS062.D21299.T410002
SMS062  02 YES 21/10/26  01:23:24  DCL11     21/10/31  %CLOUD PRODCLOUD01
                                SYSZARC.ARCPLEX.Y.DMP.2021277.A5B1CE55
                                DFHSM.DMP.DCL11.VSMS062.D21299.T242301
----- END OF - PRIMARY VOLUME - LISTING -----

```

Figure 148. Sample Printer List of dump information for a specific primary volume when you specify ALLDUMPS with PRIMARYVOLUME(volser) and BCDS

Requesting contents of backup VTOC for a primary volume from the BCDS

Table 92 on page 1379 explains the output labels when you request the contents of backup VTOC information for primary volumes from the BCDS.

Table 92. Output labels when you request contents of backup VTOC information from the BCDS for primary volumes		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
VTOC COPY	VTOCCOPY	This field contains the VTOC copy number.
PRIMARY VOLUME	SOURCEVOL	This field contains the volume serial number of the source volume.
DATASET NAME	DSN	This field contains the data set name on the primary volume.

Table 92. Output labels when you request contents of backup VTOC information from the BCDS for primary volumes (continued)

Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
ORG	ORG	<p>This field contains the data set organization on the primary volume. Possible values are:</p> <p>Value Meaning</p> <p>DA BDAM</p> <p>PE Partitioned data set extended</p> <p>PO Partitioned organization</p> <p>PS Physical sequential</p> <p>VS VSAM</p> <p>Unknown data set organization</p>
MULTI	MULTI	<p>This field contains the non-VSAM indicator for determining if a data set is a multivolume data set. Possible values are:</p> <p>Value Meaning</p> <p>Yes Data resides on multiple volumes</p> <p>No Data resides on a single volume.</p> <p>VSAM data sets, whether multivolume, single volume, or VTOC index.</p>
CREATED	CREATE	This field contains the creation date of the data set on the primary volume.
REFERENCED	REF	This field contains the date the data set was last referred to.
EXP DATE	EXP	This field contains the expiration date of the data set. This may be **NONE**.
RACF	RACF	This field contains the indication whether the data set is RACF-protected.
PSWD	PSWD	This field contains the indication whether the data set is password protected.
CHANGED	CHANGED	This field contains the indication whether the data set has been opened for something other than input since the last backup copy was made. *** Is presented for a VTOC index.

Note: The list data is obtained from the Format-1 label. For some data set VTOC entry types, not all of the fields in Table 92 on page 1379 are maintained.

Figure 149 on page 1381 is a sample printer list of the contents of a backup VTOC for a specific primary volume.


```

---- DFSMSHsm CONTROL DATASET -PRIMARY VOLUME-BCDS--- BCONTENTS --- AT 13:31:36 ON 91/07/31 FOR SYSTEM=SYSA
CONTENTS OF BACKUP VTOC COPY # 01 FOR PRIMARY VOLUME PRIM01
DATASET NAME                                ORG  MULTI  CREATED   REFERENCED  EXP DATE  RACF  PSWD  CHANGED
C253800.POFB.F40CM012.DS04.TEST12         PO    YES   91/03/31   92/05/24    92/12/31  YES   NO    YES
HSMATH0.OY45746T.N01.PSFB                 PS    NO    93/09/12   93/09/12    00/00/00  YES   YES   NO
B588374.DRIVER.N02.ESDS.DATA               VS    ***   92/01/31   93/09/01    95/12/25  YES   NO    YES

```

Figure 149. Sample Printer List from the BCDS when You Specify PRIMARYVOLUME(volser), BACKUPCONTENTS, and BACKUPCONTROLDATASET

Requesting a list of records serialized by DFSMSHsm host ID

You specify LIST HOST(*hostid*) to get a list of all MCDS, BCDS, and OCDS records currently serialized by the specified DFSMSHsm host ID.

Table 93 on page 1381 explains the output labels when you request a list of records serialized by DFSMSHsm host ID.

Table 93. Output labels when you request information about records serialized by DFSMSHsm host ID		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
HOSTID	HOSTID	This field contains the DFSMSHsm host ID of the DFSMSHsm host that serialized the record.
TYPE	TYPE	This field contains the type of control data set record.
KEY	KEY	This field contains the record key.

Figure 150 on page 1381 is a sample printer list of records serialized by a specific DFSMSHsm host when you specify the HOST(*hostid*) parameter. If you also specify RESET, the HOSTID field in each serialized record is set to X'00'.

```

ARC0817I HOSTID=3 FOUND IN DFSMSHsm CONTROL DATA SET RECORD,
TYPE=V KEY=M2TP01, RESET SUCCESSFUL
ARC0817I HOSTID=3 FOUND IN DFSMSHsm CONTROL DATA SET RECORD,
TYPE=R, KEY=BVR01-0000, RESET SUCCESSFUL
ARC0817I HOSTID=3 FOUND IN DFSMSHsm CONTROL DATA SET RECORD,
TYPE=T, KEY=L2-M2TP01-0000, RESET SUCCESSFUL

```

Figure 150. Sample Printer List when You Specify HOST(*hostid*) and RESET

Requesting information from the tape table of contents (TTOC)

You specify LIST TAPETABLEOFCONTENTS(volser) to get a list of the information contained in the TTOC for a tape backup or migration level 2 volume managed by DFSMSHsm. Preceding the list is an echo of the LIST command issued.

Table 94 on page 1381 explains the output labels when you request information from the TTOC.

Table 94. Output labels when you request information from the tape table of contents		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
VOLSER	VOLSER	This field contains the volume serial number of the tape volume.
UNIT NAME	UNIT NAME	This field contains the name of the unit where this volume can be allocated.

Table 94. Output labels when you request information from the tape table of contents (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
VOL TYPE	VOL TYPE	This field contains the DFSMSHsm volume category of the tape volume. SPILL indicates a spill backup volume. D(nn) indicates a backup volume assigned to day nn in the backup cycle. UNASS indicates that the volume has not been assigned as a daily or spill backup volume, or that it has been assigned as a daily backup volume but not to a specific day in the backup cycle. ML2 indicates that the volume is a tape migration level 2 volume.
REUSE CAPACITY	CAPACITY	This field contains the nominal number of blocks that DFSMSHsm could write to this tape if it were emptied and written again. It is the actual average in this HSMplex for all full tapes of its cartridge type, recording technology, and function as calculated by the reporting host when it last started DFSMSHsm. The blocks are 16 384 bytes each. ***** indicates that no ML2 or backup tape volumes marked full exist.
VALID BLKS	VALID BLOCKS	This field contains the number of blocks of data on the tape volume that are still valid.
PCT VALID	PCT VALID	This field contains the percent that this tape's valid blocks are of the reuse capacity. A value of 100 is shown for tapes whose number of valid blocks exceed the reuse capacity.
VOL STATUS	VOL STATUS	Empty under this heading indicates that the tape is empty. Full under this heading indicates that while DFSMSHsm was writing to tape, either an end-of-tape marker was reached or a data-movement error occurred, and the volume was marked full to prevent further use. Partial under this heading indicates that the tape is neither empty nor full. The tape has been written to but may or may not contain valid data.
RACF	RACF	YES or Y indicates that the tape volume is RACF protected.
PREV VOL	PREV VOL	If this field contains a volume serial number, the first data set on this tape volume is valid and does not begin on this volume, but is a continuation of the last data set on the indicated previous volume. If the field contains NONE, either the first data set is not valid or it begins on this volume.
SUCC VOL	SUCC VOL	If this field contains a volume serial number, the last data set on this tape volume is valid and does not end on this volume. The data set is continued on the indicated succeeding volume as the first data set on that volume. If the field contains NONE, either the last data set is not valid or it ends on this volume.
NUM REC	NUM REC	This field contains the number of offline control data set T records, including the base and extension records, used to contain this TTOC.
ONE FILE	ONE FILE	A YES or Y indicates that the tape volume has a single file containing the user data sets. A NO or N indicates that each user data set on the tape volume is in a separate tape data set. An *** indicates the ONE FILE is irrelevant because DFSMSHsm has not written on the volume.

Table 94. Output labels when you request information from the tape table of contents (continued)		
Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
ALT VOL	ALTVOL	Volume serial number of duplicate tape copy created by DFSMSHsm TAPECOPY command; *NONE* if duplicate tape is not so created; ***** indicating that the information is not known. If the optional parameter SELECT (DISASTERALTERNATEVOLUMES) is specified, this field shows the alternate tape volume that has been flagged as disaster alternate volume.
LIB	LIB	<p>This field contains the tape library information for the volume. The field can be:</p> <p>Value Meaning</p> <p><i>library name</i> the library with which the tape volume is associated.</p> <p>*NO LIB* the tape volume is not associated with a tape library.</p> <p>*ERROR** an error occurs when the library information is retrieved. Check the command activity log for additional messages.</p>
STORAGE GROUP	STORAGE GROUP	<p>This field contains the name of the storage group. If there is a library name for the volume, there must be a storage group name. Connected tape volumes must be in the same storage group and library. The storage group field can be:</p> <p>Value Meaning</p> <p><i>storage group name</i> storage group name associated with the tape volume.</p> <p>**NO SG* a storage group is not associated with this volume.</p> <p>*ERROR** an error occurs when the library information is retrieved. Check the command activity log for additional messages.</p>
DATA SET NAME	DATA SET NAME	This field contains the name of the data sets on the volume. If the data set is on a tape backup volume, this name is the DFSMSHsm-generated backup version name. If a data set is on a tape migration level 2 volume, this name is the original data set name, not the migrated data set name that DFSMSHsm generates.
NUM BLOCKS	NUM BLOCKS	This field contains the number of blocks used on this volume for this data set.
RELATIVE FBID	REL FBID	This field contains the file sequence number of this data set or the set of blocks relative to the beginning of the single tape volume. For example, the first data set or data set segment on the tape volume has a relative FBID of 4 independent of any predecessor tapes.
VSAM	VSAM	YES or Y indicates that the data set is a backup version or a migration copy of a VSAM data set.
RACF	RACF	YES or Y indicates that the data set is discretely RACF-protected.
LAST REF DATE	LAST REF DATE	This field contains the last date when the migrated data set was referred to. This field does not apply to backup versions.
EXP DATE	EXP DATE	This field contains the expiration date of the user's data set. This field does not apply to backup versions. For ML2 tapes, asterisks means the date is a never expire date.

Table 94. Output labels when you request information from the tape table of contents (continued)

Printer Output Label (OUTDATASET or SYSOUT)	Terminal Label	Description
A field containing only **** is not applicable to this volume (see individual field descriptions in the sample lists.)		

Note: The LIST TTOC *volser* ODS *dsn* command can take an extensive amount of time to complete processing when trying to write out over 1 million data set entries in the ODS *dsn*. You can only get a million records if you are using extended TTOCs.

Nine reports are possible for the LIST TTOC display output. The report columns are all the same. Only the header and trailer inserts will change to indicate what type of report has been requested on the input command.

- If the input command is issued for a list of tape volumes that have DISASTERALTERNATEVOLUMES, the header and trailer insert will be DISASTER ALTERNATE - LISTING.
- If the input command is issued for a list of tape volumes that have failed recycle, then the header and trailer insert will be FAILED RECYCLE - LISTING.
- If the input command is issued for a list of tape volumes that have failed create, then the header and trailer insert will be FAILED CREATE - LISTING.
- If the input command is issued for tape volumes that contain data sets that span four or more volumes, then the header and trailer insert will be EXCESSIVE VOLUME - LISTING.
- If the input command is issued for a list of tape volumes that have TAPETABLEOFCONTENTS SELECT(CONNECTED), then the trailer insert will be ***END OF CONNECTED TAPE VOLUME SET ***.
- If the input command is issued for a list of ML2 volumes taken away by RECALL or ABACKUP, then the header and trailer insert will be RECALL TAKE AWAY — LISTING.
- If the input command is issued for a list of volumes having alternate tapes in a library, then the header and trailer insert will be LIBRARY(ALTERNATE) — LISTING.
- If the input command is issued for a list of volumes having alternate tapes not in a library, then the header and trailer insert will be NOLIB(ALTERNATE) — LISTING.
- If the input command issued is other than those listed above, the header and trailer insert will be TAPE VOLUME TTOC - LISTING.

Figure 151 on page 1384 is a sample of an ODS output dataset list when you have specified the TAPETABLEOFCONTENTS SELECT(CONNECTED) parameter.

```
- DFSMSHSM CONTROL DATASET - TAPE VOLUME TTOC - LISTING - AT 15:05:37 ON 11/05/23 FOR SYSTEM=SYSA
VOLSER  UNIT  VOL  REUSE  VALID  PCT  VOL  RACF  PREV  SUCC  NUM  ONE  ALT  LIB  STORAGE
NAME    TYPE CAPACITY BLKS  VALID STATUS  VOL  VOL  VOL  REC  FILE VOL  VOL  GROUP
A00200  3490  D(01) 0000124000 0000000017 001  FULL  NO  *NONE* A00300 001  YES  *NONE* *NO LIB* **NO SG*
A00300  3490  D(01) 0000124000 0000000009 001  FULL  NO  A00200 *NONE* 001  YES  *NONE* *NO LIB* **NO SG*
***END OF CONNECTED TAPE VOLUME SET ***
A00800  3490  D(01) 0000124000 0000000007 001  FULL  NO  *NONE* A00700 001  YES  *NONE* *NO LIB* **NO SG*
A00700  3490  D(01) 0000124000 0000000005 001  FULL  NO  A00800 A00600 001  YES  *NONE* *NO LIB* **NO SG*
A00600  3490  D(01) 0000124000 0000000004 001  FULL  NO  A00700 *NONE* 001  YES  *NONE* *NO LIB* **NO SG*
***END OF CONNECTED TAPE VOLUME SET ***
----- END OF - TAPE VOLUME TTOC - LISTING -----
```

Figure 151. Sample Printer List when You Specify TAPETABLEOFCONTENTS SELECT(CONNECTED)

Figure 152 on page 1384 is a sample of a terminal list when you have specified TAPETABLEOFCONTENTS, NODATASETINFORMATION and TERMINAL parameter.

```
VOLSER=A00363 UNITNAME=3590-1 VOL TYPE=D(07) CAPACITY=0000008800
RACF= NO
VALID BLKS=0000003874 PCT VALID= 045 VOL STATUS= FULL PREV VOL=*NONE*
SUCC VOL=*NONE* NUM REC=001 ONE FILE=YES ALTVOL=*NONE*
LIBRARY=*NO LIB* STORAGE GROUP=*NO SG*

ARC0140I LIST COMPLETED, 5 LINE(S) OF DATA
ARC0140I (CONT.) OUTPUT
```

Figure 152. Sample Terminal List when You Specify TAPETABLEOFCONTENTS, NODATASETINFORMATION and TERMINAL

Figure 153 on page 1385 is a sample of a printer list when you have specified the TAPETABLEOFCONTENTS parameter with DATASETINFORMATION.

```
- DFSMSHSM CONTROL DATASET - TAPE VOLUME TTOC - LISTING - AT 15:05:37 ON 11/05/23 FOR SYSTEM=SYSA
```

VOLSER	UNIT	VOL	REUSE	VALID	PCT	VOL	RACF	PREV	SUCC	NUM	ONE	ALT	LIB	STORAGE
NAME	TYPE	CAPACITY	BLKS	VALID	STATUS	VOL	NO	VOL	VOL	REC	FILE	VOL	*NO LIB*	*NO SG*
BATP01	3480X	D(01)	0000031000	0000000006	001	FULL	NO	*NONE*	*NONE*	001	YES	*NONE*		
DATA SET NAME														
DFHSM.BACK.T380306.HSMATH0.F2553L03.I1001						NUM BLOCKS		RELATIVE FBID	VSAM	RACF	LAST REF DATE	EXP DATE		
DFHSM.BACK.T550306.HSMATH0.F2553L03.I1001						0000000001		00000001	NO	NO	*****	*****		
DFHSM.BACK.T560306.HSMATH0.F2553L03.I1001						0000000001		00000002	NO	NO	*****	*****		
DFHSM.BACK.T153306.HSMATH0.F2553L03.I1001						0000000001		00000003	NO	NO	*****	*****		
DFHSM.BACK.T153306.HSMATH0.F2553L03.I1001						0000000003		00000004	NO	NO	*****	*****		
----- END OF - TAPE VOLUME TTOC - LISTING -----														

Figure 153. Sample Printer List when You Specify TAPETABLEOFCONTENTS with DATASETINFORMATION

Figure 154 on page 1385 is a sample of a terminal list when you have specified TAPETABLEOFCONTENTS, DATASETINFORMATION and TERMINAL parameter.

```
VOLSER=A00363 UNITNAME=3590-1 VOL TYPE=D(07) CAPACITY=0000008800
RACF= NO
VALID BLKS=0000003874 PCT VALID= 045 VOL STATUS= FULL PREV VOL=*NONE*
SUCC VOL=*NONE* NUM REC=001 ONE FILE=YES ALTVOL=*NONE*
LIBRARY=*NO LIB* STORAGE GROUP=*NO SG*
```

DATA SET NAME=DFHSM.BACK.T094109.HSMATH1.VSAM.A1137 NUM
BLOCKS=0000000043
REL FBID=00000021 VSAM=NO RACF=NO LAST REF DATE=***** EXP
DATE=*****

DATA SET NAME=DFHSM.BACK.U094112.HSMATH1.VSAM.A1137 NUM
BLOCKS=0000000040
REL FBID=00000022 VSAM=NO RACF=NO LAST REF DATE=***** EXP
DATE=*****

DATA SET NAME=DFHSM.BACK.T124119.HSMATH7.SMSDS3.A1137 NUM
BLOCKS=0000003652
REL FBID=00000064 VSAM=NO RACF=NO LAST REF DATE=***** EXP
DATE=*****

ARC0140I LIST COMPLETED, 17 LINE(S) OF DATA
ARC0140I (CONT.) OUTPUT

Figure 154. Sample Terminal List when You Specify TAPETABLEOFCONTENTS, DATASETINFORMATION and TERMINAL

Figure 155 on page 1385 is a sample printer list when you have specified TAPETABLEOFCONTENTS, SELECT, NOALTERNATEVOLUMES, and LIBRARY parameters.

```
- DFSMSHSM CONTROL DATASET - TAPE VOLUME TTOC - LISTING - AT 15:05:37 ON 11/05/23 FOR SYSTEM=SYSA
```

VOLSER	UNIT	VOL	REUSE	VALID	PCT	VOL	RACF	PREV	SUCC	NUM	ONE	ALT	LIB	STORAGE
NAME	TYPE	CAPACITY	BLKS	VALID	STATUS	VOL	YES	VOL	VOL	REC	FILE	VOL		GROUP
M2TP02	3490	ML2	0000124000	0000000002	001	FULL	NO	*NONE*	*NONE*	001	*NONE*	*NONE*	ATLL1B01	ATLSG001
M2TP03	3490	ML2	0000124000	0000000002	001	PART	NO	*NONE*	*NONE*	001	*NONE*	*NONE*	ATLL1B01	ATLSG001
----- END OF - TAPE VOLUME TTOC - LISTING -----														

Figure 155. Sample Printer List when You Specify TAPETABLEOFCONTENTS, SELECT, NOALTERNATEVOLUMES, and LIBRARY

Figure 156 on page 1385 is a sample printer list when you have specified TAPETABLEOFCONTENTS, SELECT, and DISASTERALTERNATEVOLUMES parameter. The volumes under the heading ALT VOL are disaster alternate volumes.

```
- DFSMSHSM CONTROL DATASET - DISASTER ALTERNATE - LISTING - AT 15:05:37 ON 11/05/23 FOR SYSTEM=SYSA
```

VOLSER	UNIT	VOL	REUSE	VALID	PCT	VOL	RACF	PREV	SUCC	NUM	ONE	ALT	LIB	STORAGE
NAME	TYPE	CAPACITY	BLKS	VALID	STATUS	VOL	YES	VOL	VOL	REC	FILE	VOL		GROUP
M2TP02	3490	ML2	0000124000	0000000002	001	FULL	YES	*NONE*	*NONE*	001	*NONE*	M2TP12	ATLL1B01	ATLSG001
M2TP03	3490	ML2	0000124000	0000000002	001	PART	NO	*NONE*	*NONE*	001	*NONE*	M2TP13	ATLL1B01	ATLSG001
----- END OF - DISASTER ALTERNATE - LISTING -----														

Figure 156. Sample Printer List when You Specify TAPETABLEOFCONTENTS, SELECT, and DISASTERALTERNATEVOLUMES

Figure 157 on page 1386 is a sample of a printer list when you have specified TAPETABLEOFCONTENTS, SELECT, ML2 and FAILEDRECYCLE parameters.

User authorization

```
- DFSMSHSM CONTROL DATASET - FAILED RECYCLE - LISTING - AT 15:05:37 ON 11/05/23 FOR SYSTEM=SYSA

VOLSER  UNIT  VOL  REUSE  VALID  PCT  VOL  RACF  PREV  SUCC  NUM  ONE  ALT  LIB  STORAGE
NAME    TYPE  CAPACITY  BLKS  VALID  STATUS  YES  VOL  VOL  VOL  REC  FILE  VOL  LIB*  GROUP
M2TP03  3490  ML2  0000124000 0000000003 001  FULL  YES  *NONE* *NONE* 001  YES  *NONE* *NO LIB* **NO SG*
M2TP04  3490  ML2  0000124000 0000000003 001  FULL  YES  *NONE* *NONE* 001  YES  *NONE* *NO LIB* **NO SG*
BATP04  3490  ML2  0000124000 0000000003 001  FULL  YES  *NONE* *NONE* 001  YES  *NONE* *NO LIB* **NO SG*
----- END OF - FAILED RECYCLE - LISTING -----
```

Figure 157. Sample Printer List when You Specify TAPETABLEOFCONTENTS, SELECT, ML2 and FAILEDRECYCLE.

Figure 158 on page 1386 is a sample of a printer list when you have specified TAPETABLEOFCONTENTS and FAILEDCREATE parameters.

```
- DFSMSHSM CONTROL DATASET - FAILED CREATE - LISTING - AT 15:05:37 ON 11/05/23 FOR SYSTEM=SYSA

VOLSER  UNIT  VOL  REUSE  VALID  PCT  VOL  RACF  PREV  SUCC  NUM  ONE  ALT  LIB  STORAGE
NAME    TYPE  CAPACITY  BLKS  VALID  STATUS  YES  VOL  VOL  VOL  REC  FILE  VOL  LIB*  GROUP
BATP01  3490  D(01) 0000124000 0000000009 001  YES  NO  *NONE* *NONE* 001  NO  *NONE* *NO LIB* **NO SG*

*** ARC0378I TT0C RECORD AND TAPE MEDIA CONTENTS ARE INCONSISTENT ON TAPE VOLUME BATP01 ***
*** REUSE CAPACITY, VALID BLOCKS, PCT VALID, AND NUM REC DO NOT CONTAIN VALUES ***
*** CONSISTENT WITH THE TAPE MEDIA ***

M2TP01  3480X ML2  0000031000 0000000014 001  YES  NO  *NONE* *NONE* 001  NO  *NONE* *NO LIB* **NO SG*
*** ARC0378I TT0C RECORD AND TAPE MEDIA CONTENTS ARE INCONSISTENT ON TAPE VOLUME M2PT01 ***
*** REUSE CAPACITY, VALID BLOCKS, PCT VALID, AND NUM REC DO NOT CONTAIN VALUES ***
*** CONSISTENT WITH THE TAPE MEDIA ***
----- END OF - FAILED CREATE - LISTING -----
```

Figure 158. Sample Printer List when You Specify TAPETABLEOFCONTENTS and FAILEDCREATE

Figure 159 on page 1386 is a sample printer list for the EXCESSIVEVOLUMES parameter.

```
- DFSMSHSM CONTROL DATASET - EXCESSIVE VOLUMES - LISTING - AT 15:05:37 ON 11/05/23 FOR SYSTEM=SYSA

VOLSER  UNIT  VOL  REUSE  VALID  PCT  VOL  RACF  PREV  SUCC  NUM  ONE  ALT  LIB  STORAGE
NAME    TYPE  CAPACITY  BLKS  VALID  STATUS  YES  VOL  VOL  VOL  REC  FILE  VOL  LIB*  GROUP
BATP01  3490  D(01) 0000124000 00000000783 001  FULL  NO  *NONE* *NONE* 001  NO  *NONE* *NO LIB* **NO SG*
BATP02  3490  D(01) 0000124000 00000000783 001  FULL  NO  BATP01 BATP03 001  NO  *NONE* *NO LIB* **NO SG*
BATP03  3490  D(01) 0000124000 00000000783 001  FULL  NO  BATP02 BATP04 001  NO  *NONE* *NO LIB* **NO SG*
BATP04  3490  D(01) 0000124000 00000000783 001  FULL  NO  BATP03 BATP05 001  NO  *NONE* *NO LIB* **NO SG*
BATP05  3490  D(01) 0000124000 00000000783 001  PART  NO  BATP04 BATP06 001  NO  *NONE* *NO LIB* **NO SG*
----- END OF - EXCESSIVE VOLUMES - LISTING -----
```

Figure 159. Sample Printer List when You Specify EXCESSIVEVOLUMES

Figure 160 on page 1386 is a sample printer list for the SELECT(NOTASSOCIATED) parameter.

```
- DFSMSHSM CONTROL DATASET - TAPE VOLUME TT0C - LISTING - AT 15:05:37 ON 11/05/23 FOR SYSTEM=SYSA

VOLSER  UNIT  VOL  REUSE  VALID  PCT  VOL  RACF  PREV  SUCC  NUM  ONE  ALT  LIB  STORAGE
NAME    TYPE  CAPACITY  BLKS  VALID  STATUS  YES  VOL  VOL  VOL  REC  FILE  VOL  LIB*  GROUP
100058  3590  ML2  0002464400 0001125520 046  PART  NO  *NONE* *NONE* 009  YES  100071 LIB01  ATLSTRGP
100459  3590  ML2  0002464400 0000880269 036  PART  NO  100778 *NONE* 004  YES  100503 LIB01  ATLSTRGP
100511  3590  ML2  0002464400 0001215931 050  PART  NO  101812 *NONE* 008  YES  100638 LIB01  ATLSTRGP
100855  3590  ML2  0002464400 0002410026 098  PART  NO  *NONE* *NONE* 022  YES  100885 LIB01  ATLSTRGP
101916  3590  ML2  0002464400 0001856704 076  PART  NO  *NONE* *NONE* 038  YES  101967 LIB01  ATLSTRGP
102409  3590  ML2  0002464400 0000474331 020  PART  NO  *NONE* *NONE* 008  YES  102490 LIB01  ATLSTRGP
102563  3590  ML2  0002464400 0002725093 100  PART  NO  101714 *NONE* 014  YES  102620 LIB01  ATLSTRGP
----- END OF - TAPE VOLUME TT0C - LISTING -----
```

Figure 160. Sample Printer List when You Specify SELECT(NOTASSOCIATED)

Requesting information for all user entries or a specific user entry

To get a list of the information for all user entries, specify the LIST command with the USER parameter. User entries exist only for those users who have been, or are currently, authorized. The output is in alphanumeric sequence by user identification. To get a list of the information for a specific user entry, specify the LIST command with the USER(userid) parameter. The list for all the users or a specific user includes the user identity and authorization.

The last line of this output specifies whether DFSMSHsm commands are protected by RACF FACILITY Class profiles or by the DFSMSHsm AUTH command. Use this information to determine whether the user IDs listed with the LIST USER command are to be used for authorization to storage administrator commands.

Figure 161 on page 1387 is a sample printer list of user entries when you specify USER.


```
- DFSMSHSM CONTROL DATASET - USER-- LISTING ----- AT 08:49:11 ON 02/08/20 FOR SYSTEM 3090

USERID  AUTH
HSMDBA1  CNTL
HSMDBA2  CNTL
HSMUSR1  USER
HSMUSR2  USER
HSMUSR3  USER
$20274   USER
#61175   USER
@818181  USER

ARC1700I DFSMSHSM COMMANDS ARE RACF PROTECTED
----- END OF - USER - LISTING -----
```

Figure 161. Sample Printer List of User Entries when You Specify USER

Chapter 77. Using the QUERY command

You use the QUERY command to determine what values you specified with the DFSMSHsm commands, to list DFSMSHsm statistics, or to list the status of pending requests.

When you issue the QUERY command, the information you have requested is printed as a message at your terminal and is sent to the DFSMSHsm log. The following list shows the messages, arranged by parameter name, associated with the QUERY command.

Note:

1. Paired braces { } mean that only one of the items within the braces will appear in the actual message text.
2. Paired brackets [] mean that the item within the brackets may or may not appear in the actual message text.
3. A vertical bar (|) separates alternative choices.

Parameter

Message ID and text

ABARS

- ARC0101I QUERY ABARS COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC6008I APPLICATION BACKUP/RECOVERY PROCNAME = *procname*
- ARC6009I APPLICATION BACKUP/RECOVERY MAXADDRESSSPACE = *number*
- ARC6033I AGGREGATE RECOVERY UNIT NAME = *unittype*
- ARC6036I AGGREGATE BACKUP OPTIMIZE = {1 | 2 | 3 | 4}
- ARC6036I AGGREGATE {BACKUP | RECOVERY | BACKUP/RECOVERY} *parmname* = *parmvalue*
- ARC6366I AGGREGATE BACKUP/RECOVERY UNIT NAME = UNITNAME
- ARC6367I AGGREGATE BACKUP/RECOVERY EXITS = {NONE | EXITNAME}
- ARC6368I AGGREGATE BACKUP/RECOVERY ACTIVITY LOG MESSAGE LEVEL IS {FULL | REDUCED}
- ARC6371I AGGREGATE RECOVERY ML2 TAPE UNIT NAME = UNITNAME
- ARC6372I NUMBER OF ABARS I/O BUFFERS = NUMBER
- ARC6373I ABARS ACTIVITY LOG OUTPUT TYPE = {SYSOUT(CLASS) | DASD}

ACTIVE

- ARC0101I QUERY ACTIVE COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0142I {BACKUP OF MCDS | BACKUP OF BCDS | BACKUP OF OCDS | BACKUP OF JRNL | MOVEMENT OF BACKUP VERSIONS | BACKUP OF MIGRATED DATA SETS | MIGRATION CLEANUP | LEVEL 1 TO LEVEL 2 MIGRATION | CDS BACKUP} CURRENTLY IN PROCESS [, TCB=*x'tcbaddress'*]
- ARC0144I AUDIT={HELD | NOT HELD} AND {ACTIVE | INACTIVE}, LIST={HELD | NOT HELD} AND {ACTIVE | INACTIVE}, RECYCLE={HELD | NOT HELD} AND {ACTIVE | INACTIVE}, REPORT={HELD | NOT HELD} AND {ACTIVE | INACTIVE}
- ARC0160I MIGRATION={NOT HELD | HELD AT END OF VOLUME | LIMITED | HELD}, AUTOMIGRATION={NOT HELD | HELD AT END OF VOLUME | LIMITED | HELD}, RECALL={TOTALLY HELD | PARTIALLY HELD | NOT HELD}, TAPERECALL={TOTALLY HELD | TSO HELD | NOT HELD}, DATA SET MIGRATION={ACTIVE | INACTIVE}, VOLUME MIGRATION={ACTIVE | INACTIVE}, DATA SET RECALL={ACTIVE | INACTIVE}, EVENT DRIVEN MIGRATION={HELD | NOT HELD}

- ARC0161I {MIGRATING | MIGRATION PREPROCESSING | FREEVOL PROCESSING | DBA/DBU PROCESSING | BACKING UP | RECOVERING | RECYCLING | AUDITING | {DUMPING | FRBACKUP DUMP OF | FRBACKUP DUMPONLY OF} | MOUNTING INITIAL TAPE FOR DS BACKUP, | RESTORING } VOLUME {*volser* | *volser*, SGROUP=*sg* | *volser*, COPY POOL = *cpname* } FOR USER {*userid* | **AUTO** | * } REQUEST {*request-number*[ON HOST=*hostid*] | NONE | * } [,TASK=*taskname*] [,TCB=X'*tcbaddress*' | TCB2=X'*tcbaddress*' | TCB3=X'*tcbaddress*' | TCB=X'****?'***':VOL = [*volser1* | NONE], ADDR = {*address1* | NONE}; VOL = {*volser2* | NONE}, ADDR = {*address2* | NONE}; VOL = {*volser3* | NONE}, ADDR = {*address3* | NONE}]
- ARC0162I {MIGRATING | BACKING UP | RECALLING | RECOVERING | DELETING | RESTORING | FRRECOV OF} DATA SET *dsname* FOR USER *userid*, REQUEST *request* ON HOST *hostid* [[,TCB=X'*tcbaddress*']: VOL = {*volser1* | NONE}, ADDR = {*address1* | NONE}; VOL = {*volser2* | NONE}, ADDR = {*address2* | NONE}; VOL = {*volser3* | NONE}, ADDR = {*address3* | NONE}]
- ARC0163I BACKUP={NOT HELD | HELD AT END OF VOLUME | HELD}, AUTOBACKUP={NOT HELD | HELD AT END OF VOLUME | HELD}, RECOVERY={NOT HELD | HELD AT END OF VOLUME | HELD}, TAPEDATASETRECOVERY={HELD | NOT HELD}, DATA SET BACKUP={HELD | NOTHELD | TAPEHELD | DASDHELD}, DATA SET BACKUP={ACTIVE | INACTIVE | DASDACTIVE | TAPEACTIVE}, DATA SET BACKUP ACTUAL IDLETASKS= (ALLOC=ALLOCATED, MAX=MAXIMUM), DATA SET RECOVERY={ACTIVE | INACTIVE},VOLUME BACKUP={ACTIVE | INACTIVE}, VOLUME RECOVERY={ACTIVE | INACTIVE}
- ARC0415I EXPIREBV={HELD | NOT HELD} AND {ACTIVE | INACTIVE}, LAST STORED BACKUP VERSION KEY = *bcds-key*, LAST STORED ABARS VERSION KEY = *abars-key*, LAST PLANNED END KEY = *last-planned-end-key*
- ARC0437I {TAPECOPY | TAPEREPL} {HELD | NOT HELD} AND {ACTIVE | INACTIVE }
- ARC0460I {PRIVATE | EXTENDED PRIVATE} AREA LIMIT=*limit* UNALLOCATED=*unallocated* LARGEST FREE AREAS=*first*, *second*
- ARC0642I DUMP={NOT HELD | HELD AT END OF VOLUME | HELD}, AUTODUMP={NOT HELD | HELD AT END OF VOLUME | HELD}, VOLUME DUMP={ACTIVE | INACTIVE}, VOLUME RESTORE={ACTIVE | INACTIVE}, DATA SET RESTORE={ACTIVE | INACTIVE}
- ARC0841I RECYCLED NET *number category* VOLUMES TOWARD A LIMIT OF *limit*
- ARC1540I COMMON RECALL QUEUE PLACEMENT FACTORS: CONNECTION STATUS={CONNECTING | CONNECTED | DISCONNECTING | UNCONNECTED | RETRY | FAILED}, CRQPLEX HOLD STATUS={NONE | ALL | RECALL(TAPE) | RECALL(TAPE(TSO))}, HOST COMMONQUEUE HOLD STATUS={NONE | CQ | CQ(RECALL) | CQ(RECALL(PLACEMENT))}, STRUCTURE ENTRIES=*ent*% FULL, STRUCTURE ELEMENTS=*elem*% FULL
- ARC1541I COMMON RECALL QUEUE SELECTION FACTORS: CONNECTION STATUS={CONNECTING | CONNECTED | DISCONNECTING | UNCONNECTED | RETRY | FAILED}, HOST RECALL HOLD STATUS={NONE | ALL | RECALL(TAPE) | RECALL(TAPE(TSO))}, HOST COMMONQUEUE HOLD STATUS={NONE | CQ | CQ(RECALL) | CQ(RECALL(SELECTION))}
- ARC1560I COMMON *type* QUEUE FACTORS: GROUPNAME={*groupname* | *}, CONNECTION STATUS={CONNECTING | CONNECTED | DISCONNECTING | UNCONNECTED | QUIESCED}, MASTERSCHEDULER HOSTID={*hostid* | NONE | *}, MASTER SCHEDULER STATUS={DISABLED | ENABLED | QUIESCED | *}
- ARC1822I {FRBACKUP | FRRECOV | FRBACKUP DUMP OR DUMPONLY | FRRECOV FROMDUMP} OF COPY POOL *cpname* FOR USER *userid*, REQUEST *request-number* ON HOST *host_id* IS IN PROGRESS: NOT PROCESSED = *xx*, TOTAL = *yy*
- ARC1826I FRBACKUP = {HELD | NOT HELD } AND {ACTIVE | INACTIVE}, FRRECOV={HELD | NOT HELD } AND {ACTIVE | INACTIVE}, FRBACKUP DUMP = {HELD | HELD EOVS | NOT HELD} AND {ACTIVE | INACTIVE}, FRRECOV(TAPE) = { HELD | HELD EOVS | NOT HELD} AND {ACTIVE | INACTIVE}, FRRECOV(DATASET) = {HELD | NOT HELD} AND {ACTIVE | INACTIVE}
- ARC6018I AGGREGATE BACKUP/RECOVERY = {ACTIVE | INACTIVE | DISABLED}
- ARC6019I AGGREGATE BACKUP = {HELD | HELD EOD | NOT HELD}, AGGREGATE RECOVERY = {HELD | HELD EOD | NOT HELD}

- ARC6020I {BACKING UP | RECOVERING} {AGGREGATE GROUP *agname* | CONTROL FILE *dsname*} FOR USER *userid* REQUEST *request number* [,SASINDEX = *index*]
- ARC6380I AGGREGATE BACKUP{AGNAME} = {HELD | HELD EOD}, AGGREGATE RECOVERY {DATASETNAME *datasetname* | AGGREGATE *agname*} = {HELD | HELD EOD}

ARPOOL

- ARC0101I QUERY ARPOOL COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0230I NO {DATA SET | VOLUME | AGGREGATE RECOVERY} POOLS DEFINED
- ARC6015I ARPOOL NAME = *agname* ML1VOLS = *volser1 ... volsern* LOVOLS = *volser1 ... volsern*
- ARC6363I ALL | ML1VOLS{*} | LOVOLS{*} USED IN DEFINE

AUTOPROGRESS

- ARC0101I QUERY AUTOPROGRESS COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0246I {ODM | SMS} VOLUMES RESTRICTED TO PROCESSING BY THIS PROCESSING UNIT: NOT PROCESSED = *number of volumes*, TOTAL = *number of volumes* {ODM | SMS} VOLUMES NOT RESTRICTED TO PROCESSING BY ANY PROCESSING UNIT: NOT PROCESSED = *number of volumes*, TOTAL = *number of volumes* NON-SMS VOLUMES: NOT PROCESSED = *number of volumes*, TOTAL = *number of volumes*
- ARC0247I {AUTO BACKUP | AUTO DUMP | PRIMARY SPACE MANAGEMENT | INTERVAL MIGRATION | ON-DEMAND MIGRATION | NO AUTOMATIC FUNCTION} IS CURRENTLY PROCESSING DFSMSHSM MANAGED VOLUMES
- ARC1824I COPY POOLS RESTRICTED TO PROCESSING BY THIS PROCESSING UNIT: NOT PROCESSED = *uu*, TOTAL = *vv*. COPY POOLS NOT RESTRICTED TO PROCESSING BY THIS PROCESSING UNIT: NOT PROCESSED = *yy*, TOTAL = *zz*
- ARC1828I COPY POOL *cpname* { RESTRICTED | NOT RESTRICT } TO PROCESSING BY THIS PROCESSING UNIT: NOT PROCESSED = *uu*, TOTAL = *m*.

BACKUP

- ARC0101I QUERY BACKUP COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0154I MAXBACKUPTASKS=*tasks*, ABSTART=(*time1 time2 time3*), VERSIONS=*versions*, FREQUENCY=*days*, SKIPABPRIMARY={YES | NO}, BACKUP PREFIX=*bprefix*, INCREMENTALBACKUP={CHANGEDONLY | ORIGINAL}, PROFILEBACKUP={YES | NO}, INUSE=(YES | NO, DELAY= *min*, SERIALIZATION=REQUIRED | PREFERRED)
- ARC0164I DAY=*day* | SPILL | UNASSIGNED VOLS=*volser-flag*, ...
- ARC0269I DS DASD BACKUP TASKS={ *number of tasks*}, DS TAPE BACKUP TASKS={ *number of tasks*}, DEMOUNTDELAY={ *delay minutes*}, MAXIDLETASKS={*number idle drives*}, DS BACKUP MAX DASD SIZE={ *kbytes*}, DS BACKUP STD DASD SIZE={ *kbytes*}, SWITCHTAPES TIME={*time to demount idle tapes*}, PARTIALTAPE={MARKFULL | SETSYS | REUSE}, GENVSAMCOMPNames={YES | NO}
- ARC0271I BACKUP CYCLE LENGTH=*nday(s)*, TODAY IS DAY={** | *n*}, CYCLE START DATE=*yy/mm/dd*, VOLUME LIMIT/DAY=*limit*, AVAILABLE BACKUP VOLUMES=*total*
- ARC0273I DUMP CYCLE LENGTH=*ndays* DAYS, CYCLE={*NONE* | *string*}, TODAY IS DAY = {** | *n*} CYCLE START DATE = {*yy/mm/dd* | *NONE*} {LEVEL | AUTODUMP | AUTODUMP | LEVEL} FUNCTIONS {ELIGIBLE | NOT ELIGIBLE} TO BE {STARTED | RE-STARTED} {DUMP CYCLE NOT DEFINED | CYCLE START TIME NOT SPECIFIED}
- ARC0274I BACKUP={YES({ANY | DASD | TAPE { (*unit*) }}} | NO}, SPILL={YES({ANY | DASD | TAPE {(*unit*)}}) | NO}, MAXDSRECOVERTASKS=*nn*, MAXDSTAPERECOVERTASKS=*nn*
- ARC0638I MAXDUMPTASKS=*dtasks*, ADSTART=(*time1d time2d time3d*), DUMPIO=(*n,m*), VOLUMEDUMP(CC={STANDARD | PREFERRED | REQUIRED | VIRTUALREQUIRED | VIRTUALPREFERRED | CACHEREQUIRED | CACHEPREFERRED}), MAXDUMPRECOVERTASKS=*rdtasks*

CDSVERSIONBACKUP

- ARC0101I QUERY CDSVERSIONBACKUP COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0375I CDSVERSIONBACKUP,{MCDSBACKUPDSN= *dsname1*, BCDSBACKUPDSN=*dsname2*, OCDSBACKUPDSN=*dsname3*, JRNLBACKUPDSN=*dsname4*}
- ARC0376I BACKUPCOPIES=*backupcopies*, BACKUPDEVICECATEGORY={DASD | TAPE (UNITNAME=*unitname*, DENSITY=*density*, {RETPD=*retentionperiod* | EXPDT=*expirationdate*} {PARALLEL | NOPARALLEL})}, LATESTFINALQUALIFIER=*Xnnnnnnnn* DATAMOVER = {HSM | DSS}

COMMONQUEUE

- ARC1543I *type* MWE FOR DATASET *name*, FOR USER *userid*, REQUEST *request-number*, WAITING TO BE PROCESSED ON A COMMON QUEUE, *nmwe* MWES AHEAD OF THIS ONE[, REQUEST ORIGINATED ON HOST *hostid*]
- ARC1545I COMMON QUEUE STRUCTURE FULLNESS: COMMON RECALL QUEUE: STRUCTURE ENTRIES=*crq_ent%* FULL, STRUCTURE ELEMENTS=*crq_elem%* FULL
- ARC1562I VOLUME {*volser* | *volser*, SGROUP=*sg* | *volser*, COPY POOL=*cpname*,} {DUMPING | FRBACKUP | FRBACKUP DUMP | FRBACKUP DUMPNLY | RESTORING | FRRECOV | FRRECOV FROMDUMP}, FOR USER {*userid* | **AUTO** | * } REQUEST {*request_number* [ON HOST=*hostid*] | NONE }, BEING PROCESSED FROM A COMMON QUEUE, GROUPNAME=*groupname*
- ARC1563I VOLUME {*volser* | *volser*, SGROUP=*sg* | *volser*, COPY POOL=*cpname*,} {DUMPING | FRBACKUP | FRBACKUP DUMP | FRBACKUP DUMPNLY | RESTORING | FRRECOV | FRRECOV FROMDUMP }, FOR USER {*userid* | **AUTO** | * } REQUEST {*request_number* [FROM HOST=*hostid*] | NONE }, WAITING TO BE PROCESSED ON A COMMON QUEUE, *nmwe* MWES AHEAD OF THIS ONE, GROUPNAME=*groupname*

CONTROLDATASETS

- ARC0101I QUERY CONTROLDATASETS COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0947I CDS SERIALIZATION TECHNIQUE IS *technique*
- ARC0148I {MCDS | BCDS | OCDS | JOURNAL} TOTAL SPACE = *k-bytes* K-BYTES, CURRENTLY ABOUT *percent%* FULL, WARNING THRESHOLD = *thresh%*, TOTAL FREESPACE = *percent%*, EA={YES | NO}, CANDIDATE VOLUMES = *numvols*
- ARC0948I {MCDS | BCDS | OCDS} INDEX TOTAL SPACE = *kbytes* KBYTES, CURRENTLY ABOUT *percent%* FULL, WARNING THRESHOLD = *thresh%*, CANDIDATE VOLUMES=*numvols*
- ARC0946I {MCDS|BCDS} {Low Key|High Key} = {*low key*|*high key*}

COPYPOOL

- ARC1820I THE FOLLOWING VOLUMES IN COPY POOL *cpname*, VERSION *ver*, HAVE AN ACTIVE FLASHCOPY BACKGROUND COPY
- ARC1821I NONE OF THE VOLUMES IN COPY POOL *cpname*, VERSION *vernum*, ARE IN AN ACTIVE FLASHCOPY RELATIONSHIP

CSALIMITS

- ARC0101I QUERY CSALIMITS COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0203I CSALIMITS={NO | YES}, CSA CURRENTLY USED=*nnnnnn* BYTES, MWE=*www*. MAXIMUM=*xxxxxx*K BYTES, ACTIVE=*yyy%*, INACTIVE=*zzz%*

DATASETNAME(*dsname*)

- ARC0101I QUERY {DATASETNAME | USER | REQUEST} COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0161I {MIGRATING | BACKING UP | RECOVERING | RECYCLING | AUDITING | {DUMPING | FRBACKUP DUMP OF | FRBACKUP DUMPONLY OF} | MOUNTING INITIAL TAPE FOR DS BACKUP, | RESTORING } VOLUME {*volser* | *volser*, SGROUP=*sg* | *volser*, COPY POOL = *cpname* } FOR USER {*userid* | **AUTO** | * } REQUEST {*request-number* | NONE | * } [, TASK = *taskname*] [, TCB=X'*tcbaddress*' | TCB2=X'*tcbaddress*' | TCB3=X'*tcbaddress*' | TCB=X'****?****': VOL = {*volser1* | NONE}, ADDR = {*address1* | NONE}; VOL = {*volser2* | NONE}, ADDR = {*address2* | NONE}; VOL = {*volser3* | NONE}, ADDR = {*address3* | NONE}]
- ARC0162I {MIGRATING | BACKING UP | RECALLING | RECOVERING | DELETING | RESTORING | FRRECOV OF} DATA SET *dsname* FOR USER *userid*, REQUEST *request* ON HOST *hostid* [[,TCB=X'*tcbaddress*']: VOL = {*volser1* | NONE}, ADDR = {*address1* | NONE}; VOL = {*volser2* | NONE}, ADDR = {*address2* | NONE}; VOL = {*volser3* | NONE}, ADDR = {*address3* | NONE}]
- ARC0165I USER NOT AUTHORIZED TO QUERY REQUESTS FOR OTHER USERIDS OR REQNUM MISSING
- ARC0166I NO DFSMSHSM REQUEST FOUND FOR QUERY
- ARC0167I *type* MWE FOR {VOLUME | DATA SET | COMMAND | AGGREGATE GROUP | CONTROL FILE DATA SET | COPY POOL} {*name* | *name*, SGROUP = *sg* | *name*, COPY POOL = *cpname*} FOR USER *userid*, REQUEST *request*, WAITING TO BE PROCESSED, *nmwe* MWE(S) AHEAD OF THIS ONE
- ARC1543I *type* MWE FOR DATASET *name*, FOR USER *userid*, REQUEST *request-number*, WAITING TO BE PROCESSED ON A COMMON QUEUE, *nmwe* MWES AHEAD OF THIS ONE[, REQUEST ORIGINATED ON HOST *hostid*]

IMAGE

- ARC0101I QUERY IMAGE COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0250I HOST PROCNAME JOBID ASID MODE
- ARC0250I *host procname jobid asid* {MAIN | AUX}

MIGRATIONLEVEL2

- ARC0101I QUERY MIGRATIONLEVEL2 COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0224I LOW KEY HIGH KEY VOLSER
- ARC0225I *lowkey highkey* {*volser* | *NONE*}
- ARC0226I MIGRATION LEVEL 2 UNDEFINED
- ARC0227I ML2 TAPE TARGET VOLS: DSMIG = { *volserA* | *NONE* } VOLMIG=(T01={*volser1* | *NONE*}, T02={*volser2* | *NONE*}, T03={*volser3* | *NONE*}, T04={*volser4* | *NONE*}, T05={*volser5* | *NONE*}, T06={ *volser6* | *NONE*}, T07={*volser7* | *NONE*}, T08={*volser8* | *NONE*}, T09={*volser9* | *NONE*}, T10={*volser10* | *NONE*}, T11={*volser11* | *NONE*}, T12={ *volser12* | *NONE*}, T13={*volser13* | *NONE*}, T14={*volser14* | *NONE*}, T15={*volser15* | *NONE*}, T16={*volser16* | *NONE*}) | SSMMIG=(S01={*volser1* | *NONE*}, S02={*volser2* | *NONE*}, S03={*volser3* | *NONE*}, S04={*volser4* | *NONE*}, S05={*volser5* | *NONE*}, S06={*volser6* | *NONE*}, S07={*volser7* | *NONE*}, S08={*volser8* | *NONE*}, S09={*volser9* | *NONE*}, S10={*volser10* | *NONE*}, S11={*volser11* | *NONE*}, S12={*volser12* | *NONE*}, S13={*volser13* | *NONE*}, S14={*volser14* | *NONE*}, S15={*volser15* | *NONE*})*NONE*)

NETWORK

- ARC0101I QUERY NETWORK COMMAND STARTING ON HOST=*host*
- ARC1590I CLOUD *cloud_name* ENDPOINTS AND STATISTICS
- ARC1591I ENDPOINT=*endpoint*, PORT=*port*, PREFERENCE=PREferred|BACKUP, STATUS=GOOD|BAD|UNKNOWN. ACTIVE CONNECTIONS=*number_of_active_connections*
- ARC0101I QUERY NETWORK COMMAND COMPLETED ON HOST=*host*

POOL

- ARC0101I QUERY {POOL | VOLUMEPOOL} COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0230I NO {DATA SET | VOLUME | APPLICATION RECOVERY} POOLS DEFINED
- ARC0232I {DATA SET | VOLUME} POOLS={ *char* | *poolid*} VOLS=*volser* {*volser* ... *volser*}

REQUEST

- ARC0101I QUERY {DATASETNAME | USER | REQUEST} COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0161I {MIGRATING | MIGRATION PREPROCESSING | FREEVOL PROCESSING | DBA/DBU PROCESSING | BACKING UP | RECOVERING | RECYCLING | AUDITING | {DUMPING | FRBACKUP DUMP OF | FRBACKUP DUMPOONLY OF} | MOUNTING INITIAL TAPE FOR DS BACKUP,| RESTORING } VOLUME {*volser* | *volser*, SGROUP=*sg* | *volser*, COPY POOL = *cpname* } FOR USER {*userid* | **AUTO** | * } REQUEST {*request-number*[ON HOST=*hostid*] | NONE | * } [,TASK=*taskname*] [,TCB=X'*tcbaddress*' | TCB2=X'*tcbaddress*' | TCB3=X'*tcbaddress*' | TCB=X'*****?***':VOL = [*volser1* | NONE], ADDR = {*address1* | NONE}; VOL = {*volser2* | NONE}, ADDR = {*address2* | NONE}; VOL = {*volser3* | NONE}, ADDR = {*address3* | NONE}]
- ARC0162I {MIGRATING | BACKING UP | RECALLING | RECOVERING | DELETING | RESTORING | FRRECOV OF} DATA SET *dsname* FOR USER *userid*, REQUEST *request* ON HOST *hostid* [[,TCB=X'*tcbaddress*']: VOL = {*volser1* | NONE}, ADDR = {*address1* | NONE}; VOL = {*volser2* | NONE}, ADDR = {*address2* | NONE}; VOL = {*volser3* | NONE}, ADDR = {*address3* | NONE}]
- ARC0165I USER NOT AUTHORIZED TO QUERY REQUESTS FOR OTHER USERIDS OR REQNUM MISSING
- ARC0166I NO DFSMSHSM REQUEST FOUND FOR QUERY
- ARC0167I *type* MWE FOR {VOLUME | DATA SET | COMMAND | AGGREGATE GROUP | CONTROL FILE DATA SET | COPY POOL} {*name* | *name*, SGROUP = *sg* | *name*, COPY POOL = *cpname*} FOR USER *userid*, REQUEST *request*, WAITING TO BE PROCESSED, *nmwe* MWE(S) AHEAD OF THIS ONE
- ARC1543I *type* MWE FOR DATASET *name*, FOR USER *userid*, REQUEST *request-number*, WAITING TO BE PROCESSED ON A COMMON QUEUE, *nmwe* MWES AHEAD OF THIS ONE[, REQUEST ORIGINATED ON HOST *hostid*]
- ARC1822I {FRBACKUP | FRRECOV | FRBACKUP DUMP OR DUMPOONLY | FRRECOV FROMDUMP} OF COPY POOL *cpname* FOR USER *userid*, REQUEST *request-number* ON HOST *host_id* IS IN PROGRESS: NOT PROCESSED = *xx*, TOTAL = *yy*

RETAIN

- ARC0101I QUERY RETAIN COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0174I NO RETAIN LEVEL ENTRIES
- ARC0175I LEVEL QUALIFIER AND MIGRATION RESTRICTION TYPE
- ARC0176I QUALIFIER=*qualifier* RESTRICTION TYPE=*type*

SECURITY

- ARC1700I DFSMSHSM COMMANDS ARE {RACF | AUTH} PROTECTED

SETSYS

- ARC0101I QUERY SETSYS COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0139I MAXINTERVALTASKS=*xx*, ONDEMANDMIGRATION={YES | NO}, ODMNOTIFICATIONLIMIT=*limit*, MAXSSMTASKS(TAPEMOVEMENT=*mm* , CLEANUP=*nn*), EVENTDRIVENMIGRATIONQUIESCE=(*hhmm1 hhmm2 hhmm3 hhmm4*)

- ARC0147I BUDENSITY=*density*, BUUNIT=*unittype*, BU RECYCLE PERCENTAGE= *percent%*, MOUNT WAIT TIME=*min* MINUTE(S), TAPESPANSIZE(*size*)
- ARC0149I MONITOR({STARTUP | NOSTARTUP}, {SPACE | NOSPACE}, {VOLUME | NOVOLUME}), MCDS(*thresh*), BCDS(*thresh*), OCDS(*thresh*), JOURNAL(*thresh*)
- ARC0150I JOURNAL={NONE | SPEED | RECOVERY}, LOG={YES | NO | HELD}, TRACE={YES | NO}SMFID={ *smfid* | NONE}, DEBUG={YES | NO}, EMERG={YES | NO}, JES={2 | 3}, SYS1DUMP={YES | NO}, RACFIND={YES | NO}, ERASEONSCRATCH (YES | NO), DSSXMMODE={Y | N}, PDA={NONE | OFF | ON}
- ARC0151I DAYS=*days1*, ML1DAYS=*days2*, PRIMARYSPMGMTSTART=(*time1 time2* | NONE), MAXMIGRATIONTASKS=*nn*, INTERVALMIGRATION={YES | NO}, MIGRATIONCLEANUPDAYS=(*days3, days4, days5, days6*), SDSP={*number* KB | NONE}, MIGRATION PREFIX= *mprefix*, SCRATCH EXPIRED DATA SETS={YES | NO}, SSMSTART=(*time1 time2* | NONE)
- ARC0152I MAXRECALLTASKS=*tasks*, RECALL={ANYSTORAGEVOLUME(LIKE | UNLIKE) | PRIVATEVOLUME(LIKE | UNLIKE)}, MAXEXTENTS=*extents*, CONVERSION={NO | REBLOCKBASE | REBLOCKTOANY | REBLOCKTOUNLIKE}, VOLCOUNT ={*NONE* | ANY} TAPERECALLLIMITS(TASK=*time1*, TAPE= *time2*)
- ARC0153I SCRATCHFREQ=*days*, SYSOUT(CLASS=*class*, COPIES=*number*, SPECIAL FORMS={*form* | NONE}), SWAP={YES | NO}, PERMISSION={YES | NO}, EXITS={NONE | *exits*}, UNLOAD={YES | NO}, DATASETSERIALIZATION= {USER | DFHSM}, USECMS={YES | NO}
- ARC0154I MAXBACKUPTASKS=*tasks*, ABSTART=(*time1 time2 time3*), VERSIONS=*versions*, FREQUENCY=*days*, SKIPABPRIMARY={YES | NO}, BACKUP PREFIX=*bprefix*, INCREMENTALBACKUP={CHANGEDONLY | ORIGINAL}, PROFILEBACKUP={YES | NO}, INUSE=(YES | NO, DELAY= *min*, SERIALIZATION=REQUIRED | PREFERRED)
- ARC0169I USER UNIT NAMES=*unitname(s)*
- ARC0203I CSALIMITS={NO | YES}, CSA CURRENTLY USED=*nnnnnn* BYTES, MWE=*www*. MAXIMUM=*xxxxxx*K BYTES, ACTIVE=*yyy%*, INACTIVE=*zzz%*
- ARC0267I MIGRATIONSUBTASKS={YES | NO}, ADDITIONALSUBTASKS=*nn*
- ARC0269I DS DASD BACKUP TASKS={ *number of tasks*}, DS TAPE BACKUP TASKS={ *number of tasks*}, DEMOUNTDELAY={ *delay minutes*}, MAXIDLETASKS={*number idle drives*}, DS BACKUP MAX DASD SIZE={ *kbytes*}, DS BACKUP STD DASD SIZE={ *kbytes*}, SWITCHTAPES TIME={*time to demount idle tapes*}, PARTIALTAPE={MARKFULL | SETSYS | REUSE}, GENVSAMCOMPNames={YES | NO}
- ARC0272I PRIMARY SPACE MGMT CYCLE LENGTH= *ndays* DAY(S), CYCLE={*NONE* | *string*}, TODAY IS DAY=*n*, CYCLE START DATE= *yy/mm/dd* SECONDARY SPACE MANAGEMENT CYCLE LENGTH=*ndays* DAY(S), CYCLE={*NONE* | *string*}, TODAY IS DAY=*n*, CYCLE START DATE=*yy/mm/dd*
- ARC0274I BACKUP={YES({ANY | DASD | TAPE{ (*unit*) }}} | NO}, SPILL={YES({ANY | DASD | TAPE {(*unit*)}}) | NO}, MAXDSRECOVERTASKS=*nn*, MAXDSTAPERECOVERTASKS=*nn*
- ARC0278I CLASSTRANSITIONS EVENTDRIVENMIGRATION={Y | N} SERIALIZATIONEXIT={Y | N}
- ARC0339I OPTIMUMDASDBLOCKING = {2KB | OPTIMAL | NO}, LOGGING LEVEL = {FULL | REDUCED | EXCEPTIONONLY}, LOG TYPE = {SYSOUT(*sysout-class*) | DASD}
- ARC0340I COMPACTION OPTIONS ARE: TAPEMIGRATION=[YES | NO], DASDMIGRATION=[YES | NO], TAPEBACKUP=[YES | NO], DASDBACKUP=[YES | NO], TAPEHARDWARECOMPACT=[YES | NO], ZCOMPRESS OPTIONS ARE: TAPEMIGRATE=[YES|NO], DASDMIGRATE=[YES|NO], TAPEBACKUP=[YES|NO], DASDBACKUP=[YES|NO], CLOUDMIGRATE[YES|NO]
- ARC0341I COMPACT PERCENT IS *percent%*
- ARC0342I {SOURCENAMES | OBJECTNAMES}: *name(s)*
- ARC0374I ACCEPTPSCBUSERID={YES | NO}
- ARC0375I CDSVERSIONBACKUP,{MCDSBACKUPDSN= *dsname1*, BCDSBACKUPDSN=*dsname2*, OCDSBACKUPDSN=*dsname3*, JRNLBACKUPDSN=*dsname4*}
- ARC0376I BACKUPCOPIES=*backupcopies*, BACKUPDEVICECATEGORY={DASD | TAPE (UNITNAME=*unitname*, DENSITY=*density*, {RETPD=*retentionperiod* | EXPDT=*expirationdate*} {PARALLEL | NOPARALLEL})}, LATESTFINALQUALIFIER=*Xnnnnnnnn* DATAMOVER = {HSM | DSS}

- ARC0408I INPUT TAPE ALLOCATION=(WAIT | NOWAIT), OUTPUT TAPE ALLOCATION=(WAIT | NOWAIT), RECYCLE TAPE ALLOCATION=(WAIT | NOWAIT), TAPEFORMAT={SINGLEFILE, | MULTIFILE}, TAPELIBRARYPRESELECT={YES | NO}
- ARC0410I TAPEMIGRATION={DIRECT(TAPE(ANY | *unitname*)) NONE={ROUTETOTAPE(ANY | *unitname*)}, MIGDENSITY=*density*, MIGUNIT= *unittype*, ML2 RECYCLE PERCENTAGE=*percent*%, TAPEMAXRECALLTASKS=*tasks*, ML2PARTIALSNOTASSOCIATED GOAL=(*nnn* | NOLIMIT), RECONNECT(NONE | ALL | ML2DIRECTEDONLY)
- ARC0411I TAPESECURITY={RACF | RACFINCLUDE} {PASSWORD} {EXPIRATION | EXPIRATIONINCLUDE}, {DEFERMOUNT | NODEFERMOUNT}
- ARC0412I RECYCLEOUTPUT BACKUP={*unittype* | **NONE**}, MIGRATION={*unittype* | **NONE**}, RECYCLETAKEAWAYRETRY=(YES MAXRETRYATTEMPTS(*nn*) DELAY(*ssss*) | NO)
- ARC0417I TAPE INPUT PROMPT FOR BACKUPTAPES | DUMPTAPES | MIGRATION TAPES = {Y | N}
- ARC0418I TAPEUTILIZATION PERCENT = (*pct* | NOLIMIT), UNIT = *unit* | LIBRARYMIGRATION | LIBRARYBACKUP, CAPACITYMODE = COMPATIBILITY | EXTENDED | **NONE**
- ARC0419I SELECTVOLUME={SPECIFIC | SCRATCH | SPECIFIC FOR *func*[,*func*], SCRATCH FOR *func*[,*func*]}, TAPEDELETION={SCRATCHTAPE | HSM TAPE | SCRATCHTAPE FOR *func*[,*func*], HSM TAPE FOR *func*[,*func*]}, PARTIALTAPE={MARKFULL | REUSE | MARKFULL FOR *func*, REUSE FOR *func*}, DISASTERMODE = {YES | NO}
- ARC0442I TAPE OUTPUT PROMPT FOR TAPECOPY = *x*, DUPLEX BACKUP TAPES = *x*, ERRORALTERNATE=CONTINUE(TAPECOPY | RECYCLE), DUPLEX MIGRATION TAPES = *x*, ERRORALTERNATE= {CONTINUE(TAPECOPY | RECYCLE) | MARKFULL}
- ARC0444I CLOUDMIGRATION RECONNECT(ALL | NONE)
- ARC0468I EXTENDED TIOC = {Y | N}
- ARC0638I MAXDUMPTASKS=*dtasks*, ADSTART=(*time1d time2d time3d*), DUMPIO=(*n,m*), VOLUMEDUMP(CC={STANDARD | PREFERRED | REQUIRED | VIRTUALREQUIRED | VIRTUALPREFERRED | CACHEREQUIRED | CACHEPREFERRED}), MAXDUMPRECOVERTASKS=*rdtasks*
- ARC0840I MAXRECYCLETASKS = *ytasks*, RECYCLE INPUT DEALLOCATION FREQUENCY BACKUP=*bfreq*, MIGRATION=*mfreq*
- ARC1500I PLEXNAME = *HSMplex_name*, PROMOTE PRIMARYHOST = {YES | NO}, PROMOTE SSM = {YES | NO}, COMMON RECALL QUEUE BASE NAME = {*basename* | *****}, COMMON RECALL QUEUE TAPEDATASETORDER = {PRIORITY | FBID | *****}, COMMON DUMP QUEUE BASE NAME={*basename* / **** AND MSC={YES | NO | ***}}, COMMON RECOVER QUEUE BASE NAME = {*basename* / ****} AND MSC={YES | NO | ***}
- ARC1823I MAXCOPYPOOL(FRBACKUP TASKS = *backup_tasks*, FRRECOV TASKS = *recover_tasks*, DSS TASKS = *dss_tasks*), FASTREPLICATION (DATASETRECOVERY = {PREFERRED | REQUIRED | NONE } FCRELATION= {EXTENT | FULL} VOLUMEPAIRMESSAGES={YES | NO} MESSAGEDATASET({YES | NO} HLQ=*hlq*))
- ARC6008I APPLICATION BACKUP/RECOVERY PROCNAME = *procname*
- ARC6009I APPLICATION BACKUP/RECOVERY MAXADDRESSSPACE = *number*
- ARC6011I {REPLACE | NOREPLACE} ISSUED FOR APPLICATION RECOVERY
- ARC6033I AGGREGATE RECOVERY UNIT NAME = *unittype*
- ARC6036I AGGREGATE BACKUP OPTIMIZE = {1 | 2 | 3 | 4}
- ARC6036I AGGREGATE RECOVERY TGTGDS = {DEFERRED | ACTIVE | ROLLEDOFF | SOURCE}
- ARC6036I AGGREGATE RECOVERY ABARSVOLCOUNT = {*NONE* | ANY}
- ARC6036I AGGREGATE RECOVERY PERCENTUTILIZED = *percent*,
- ARC6036I AGGREGATE BACKUP/RECOVERY ABARSDELETEACTIVITY = {YES | NO}
- ARC6036I AGGREGATE BACKUP/RECOVERY ABARSTAPES = {STACK | NOSTACK}
- ARC6036I AGGREGATE BACKUP/RECOVERY ABARSKIP = {PPRC | NOPPRC, XRC | NOXRC}

- ARC6366I AGGREGATE BACKUP/RECOVERY UNIT NAME = UNITNAME
- ARC6368I AGGREGATE BACKUP/RECOVERY ACTIVITY LOG MESSAGE LEVEL IS {FULL | REDUCED}
- ARC6371I AGGREGATE RECOVERY ML2 TAPE UNIT NAME = UNITNAME
- ARC6372I NUMBER OF ABARS I/O BUFFERS = NUMBER
- ARC6373I ABARS ACTIVITY LOG OUTPUT TYPE = {SYSOUT(CLASS) | DASD}

SPACE

- ARC0400I VOLUME *volser* IS *percent%* FREE, *tracks* FREE TRACKS, *cylinders* FREE CYLS, FRAG *fragx*
- ARC0401I LARGEST EXTENTS ARE CYLINDERS *cylinders*, TRACKS *tracks*
- ARC0402I VTOC IS *tracks* TRACKS(*totdscbs* DSCBS), *free* FREE DSCBS(*percent%* OF TOTAL), *format5* FORMAT 5 DSCBS
- ARC0406I SPACE PARAMETER ON QUERY COMMAND ONLY VALID WHEN ENTERED BY CONSOLE OPERATOR OR AUTHORIZED USER
- ARC0407I QUERY SPACE FAILED, {VOLUME *volser* NOT | NO PRIMARY OR MIGRATION LEVEL 1 VOLUMES} CURRENTLY MANAGED BY DFSMSHSM
- ARC0413I QUERY SPACE FOR VOLUME *volser* ALREADY ACTIVE - RETRY THIS VOLUME AGAIN

STARTUP

- ARC0101I QUERY STARTUP COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0143I PARMLIB MEMBER=ARCCMD*xx*, DFSMSHSM AUTHORIZED USERID=*uid*, HOSTID=*hostid*, PRIMARY HOST={YES | NO}, LOGSW={YES | NO}, STARTUP={YES | NO}, EMERGENCY={YES | NO}, CDSQ={YES | NO}, CDSR={YES | NO}, PDA={YES | NO}, RESTART = {IS SPECIFIED | NOT SPECIFIED}, CDSSHR = {YES | NO | RLS}, RNAMEDSN={YES | NO}, STARTUP PARMLIB MEMBER = {ARCSTR *yy* | NONE}
- ARC0249I CELLS=(*a,b,c,d,e*), HOSTMODE={MAIN | AUX}

STATISTICS

- ARC0101I QUERY STATISTICS COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0145I DS DELETED=*datasets*, DS DELETE FAILED=*fails*
- ARC0146I RECYCLED {BACKUP | MIGRATION} VOLUMES=*volumes*, DS=*datasets*, BLOCKS=*blocks*
- ARC0155I DFSMSHSM STATISTICS FOR *date*
- ARC0156I STARTUPS=*starts*, SHUTDOWNS=*stops*, ABENDS=*abends*, MWES=*requests*, CPU TIME=*time* SECONDS
- ARC0157I DS MIGRATE L1=*nlev1*, DS MIGRATE L2=*nlev2*, DS MIGRATE CLOUD *ncloud*, DS EXTENT REDUCTIONS= *exts* DS MIGRATE FAIL=*fails*, TRKS MIGRATE=*tracks*, {KBYTES | MBYTES | GBYTES | TBYTES}, MIGRATE=*nbytes*
- ARC0158I DS RECALL L1=*nlev1*, DS RECALL L2=*nlev2*, DS RECALL CLOUD *ncloud*, DS RECALL FAIL=*fails*, {KBYTES | MBYTES | GBYTES | TBYTES}, RECALL=*nbytes*, RECALL MOUNTS AVOIDED=*avoided*, EXTRA ABACKUP MOUNTS=*extras*
- ARC0159I DS BACKUP=*ndatasets1*, DS BACKUP FAIL=*fails1*, DS RECOVER=*ndatasets2*, DS RECOVER FAIL=*fails2*, RECOVER MOUNTS SAVED=*saved*
- ARC0277I DS TRANSITIONED=*ds*, DS TRANSITION FAIL=*fails*, {TRKS | KTRKS} TRANSITIONED=*ntracks*, {BYTES | KBYTES | MBYTES | GBYTES | TBYTES} TRANSITIONED=*nbytes*
- ARC0641I VOL DUMP=*nvols1*, VOL DUMP FAIL=*fails3*, VOL RESTORE=*nvols2*, VOL RESTORE FAIL=*fails4*, DS RESTORE= *ndatasets*, DS RESTORE FAIL=*fails5*,
- ARC1825I FAST REPLICATION VOLUME BACKUPS = *backup_req* REQUESTED, *backup_fail* FAILED; VOLUME RECOVERIES = *recover_req* REQUESTED, *recover_fail* FAILED

TRAPS

- ARC0101I QUERY TRAPS COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0204I NO ACTIVE TRAP ENTRIES
- ARC0205I TRAP IN MODULE *modname* FOR CODE *errcode*, TIMES=*errtimes*, TYPE={LOG ALWAYS | BY OCCURRENCE | SNAP {ALWAYS | ONCE | NEVER} | ABEND {ALWAYS | ONCE | NEVER}}

USER

- ARC0101I QUERY {DATASETNAME | USER | REQUEST} COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0161I {MIGRATING | MIGRATION PREPROCESSING | FREEVOL PROCESSING | DBA/DBU PROCESSING | BACKING UP | RECOVERING | RECYCLING | AUDITING | {DUMPING | FRBACKUP DUMP OF | FRBACKUP DUMPPONLY OF} | MOUNTING INITIAL TAPE FOR DS BACKUP,| RESTORING } VOLUME {*volser* | *volser*, SGROUP=*sg* | *volser*, COPY POOL = *cpname* } FOR USER {*userid* | **AUTO** | * } REQUEST {*request-number*[ON HOST=*hostid*] | NONE | * } [,TASK=*taskname*] [,TCB=X'*tcbaddress*' | TCB2=X'*tcbaddress*' | TCB3=X'*tcbaddress*' | TCB=X'*****?***':VOL = [*volser1* | NONE], ADDR = {*address1* | NONE}; VOL = {*volser2* | NONE}, ADDR = {*address2* | NONE}; VOL = {*volser3* | NONE}, ADDR = {*address3* | NONE}]
- ARC0162I {MIGRATING | BACKING UP | RECALLING | RECOVERING | DELETING | RESTORING | FRRECOV OF} DATA SET *dsname* FOR USER *userid*, REQUEST *request* ON HOST *hostid* [,TCB=X'*tcbaddress*']: VOL = {*volser1* | NONE}, ADDR = {*address1* | NONE}; VOL = {*volser2* | NONE}, ADDR = {*address2* | NONE}; VOL = {*volser3* | NONE}, ADDR = {*address3* | NONE}]
- ARC0165I USER NOT AUTHORIZED TO QUERY REQUESTS FOR OTHER USERIDS OR REQNUM MISSING
- ARC0166I NO DFSMSHSM REQUEST FOUND FOR QUERY
- ARC0167I *type* MWE FOR {VOLUME | DATA SET | COMMAND | AGGREGATE GROUP | CONTROL FILE DATA SET | COPY POOL} {*name* | *name*, SGROUP = *sg* | *name*, COPY POOL = *cpname*} FOR USER *userid*, REQUEST *request*, WAITING TO BE PROCESSED, *nmwe* MWE(S) AHEAD OF THIS ONE
- ARC1543I *type* MWE FOR DATASET *name*, FOR USER *userid*, REQUEST *request-number*, WAITING TO BE PROCESSED ON A COMMON QUEUE, *nmwe* MWES AHEAD OF THIS ONE[, REQUEST ORIGINATED ON HOST *hostid*]
- ARC1822I {FRBACKUP | FRRECOV | FRBACKUP DUMP OR DUMPPONLY | FRRECOV FROMDUMP} OF COPY POOL *cpname* FOR USER *userid*, REQUEST *request-number* ON HOST *host_id* IS IN PROGRESS: NOT PROCESSED = *xx*, TOTAL = *yy*

VOLUMEPOOL

- ARC0101I QUERY {POOL | VOLUMEPOOL} COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0230I NO {DATA SET | VOLUME | APPLICATION RECOVERY} POOLS DEFINED
- ARC0232I {DATA SET | VOLUME} POOLS={*char* | *poolid*} VOLS=*volser* {*volser* ... *volser*}

WAITING

- ARC0101I QUERY WAITING COMMAND {STARTING | COMPLETED | IGNORED} ON HOST= *x*
- ARC0168I WAITING MWES: MIGRATE=*nmigrate*, RECALL=*nrecall*, DELETE=*ndelete*, BACKUP=*nbackup*, RECOVER=*nrecover*, COMMAND=*ncommand*, ABACKUP=*nabackup*, ARECOVER=*narcovr*, FRBACKUP=*nfrbackup*, FRRECOV=*nfrrecov*, TOTAL=*ntotal*
- ARC1542I WAITING MWES ON COMMON QUEUES: COMMON RECALL QUEUE=*type_mwes*, COMMON DUMP QUEUE=*type_mwes*, COMMON RECOVER QUEUE=*type_mwes*,TOTAL=*total_mwes*

For more information about the messages you receive when you use QUERY command, refer to [z/OS MVS System Messages, Vol 2 \(ARC-ASA\)](#).

Chapter 78. Using the REPORT command

You use the REPORT command to generate reports based on the DFSMSHsm statistics records in the MCDS.

When you request a statistics report, you can specify the DELETE parameter to scratch the statistics records used as input to the report after the REPORT command finishes its processing. This parameter is not related to the DELETE subparameter of the FUNCTION parameter. You can also have the output from the REPORT command sent either to the SYSOUT class or to an alternative output data set. For an alternative method of creating customized reports about HSM activities from SMF (FSR and WWFSR) records, see [Chapter 14, “Obtaining information from DFSMSHsm,”](#) on page 559 and the *z/OS DFSMSrmm Reporting*.

Requesting a report of the daily statistics for all functions

Specify REPORT DAILY to get a report of the daily statistics. [Table 95 on page 1401](#) shows the output headings for the daily statistics report.

If you specify selected functions such as REPORT DAILY FUNCTION BACKUP or REPORT DAILY FUNCTION RECOVER, then line 5 is output only if BACKUP is requested and line 6 is output only if RECOVER is requested.

Table 95. Headings of output when you request a daily statistics report		
Line number	Daily statistics output heading	Description
1	DAILY STATISTICS REPORT FOR YY/MM/DD	When you specify REPORT DAILY, this line contains today's date. When you specify REPORT DAILY FROMDATE(yyyy/mm/dd) TODATE(yyyy/mm/dd), this line contains the first date that you specify. DFSMSHsm reports all statistics for the first date. Then it reports the daily statistics for the second day, and so forth.
2	STARTUPS=xxx	This field shows how many times DFSMSHsm was started on the specified day.
2	SHUTDOWNS=xxx	This field shows how many times DFSMSHsm was stopped on the specified day.
2	ABENDS=xxx	This field shows how many times DFSMSHsm abnormally ended for the specified day.
2	WORK ELEMENTS PROCESSED=xxxxxx	This field shows how many management work elements DFSMSHsm processed for the specified day. If DFSMSHsm is reporting daily statistics for all functions, this field means how many management work elements DFSMSHsm processed.
2	BKUP VOL RECYCLED=xxxxx	This field shows how many tape backup volumes DFSMSHsm recycled for the specified day.
2	MIG VOL RECYCLED=xxxxx	This field shows how many tape migration level 2 volumes DFSMSHsm recycled for the specified day.
3	DATA SET MIGRATIONS BY VOLUME REQUEST=xxxxxx	This field shows how many data sets were processed during automatic and command volume migration.
3	DATA SET MIGRATIONS BY DATA SET REQUEST=xxxx	This field shows how many data sets were processed by the DATASETNAME parameter of the MIGRATE command
3	BACKUP REQUESTS=xxxxxx	This field shows how many data sets were backed up during automatic and command backup processing.
4	EXTENT REDUCTIONS=xxxxxx	This field shows the number of extent reductions performed each day.
4	RECALL MOUNTS AVOIDED=xxxxxx	This field shows how many times a recall request was satisfied by a tape already mounted.
4	RECOVER MOUNTS AVOIDED=xxxxxx	This field shows how many times a recover request was satisfied by a tape already mounted.

Table 95. Headings of output when you request a daily statistics report (continued)

Line number	Daily statistics output heading	Description
5	FULL VOLUME DUMPS = xxxxx REQUESTED	This field shows how many full-volume dumps were requested.
5	xxxxx FAILED	This field shows how many of the full-volume dumps failed.
5	DUMP COPIES=xxxxxx REQUESTED	This field shows how many dump copies were requested.
5	xxxxxx FAILED	This field shows how many of the dump copies failed. You get this version of line 5 only when you issue the command REPORT DAILY FUNCTION BACKUP.
6	FULL VOLUME RESTORES = xxxxx REQUESTED	This field shows how many full-volume restores were requested.
6	xxxxx FAILED	This field shows how many of the full-volume restores failed.
6	DATASET RESTORES = xxxxx REQUESTED	This field shows how many data set restores were requested.
6	xxxxx FAILED	This field shows how many of the data set restores failed. You get this version of line 4 only when you issue the command REPORT DAILY FUNCTION RECOVER.
7	ABACKUPS=xxxxx REQUESTED	This field shows how many aggregate backups were requested.
7	xxxxx FAILED	This field shows how many of the aggregate backups failed.
7	EXTRA ABACKUP MOUNTS=xxxxx	This field shows how many times ABACKUP had to remount an ML2 tape taken away by recall.
8	DATA SET MIGRATIONS BY RECONNECTION=xxxxx	This field shows the number of data sets reconnected.
8	NUMBER OF TRACKS RECONNECTED=xxxxx	This field shows the number of tracks on level 0 volumes that were freed by reconnection.
9	FAST REPLICATION VOLUME BACKUP=xxxxxxxx REQUESTED, xxxxxxxx FAILED	This field shows the number of fast replication backup versions that were requested and how many of the fast replication backups failed.
10	FAST REPLICATION VOLUME RECOVER=xxxxxxxx REQUESTED, xxxxxxxx FAILED	This field shows the number of fast replication recoveries that were requested and how many of the fast replication recoveries failed.
11	DATA SET CLASS TRANSITION=xxxxxxxx REQUESTED, xxxxxxxx FAILED	This field shows the number of class transitions that were requested and how many of the class transitions failed.

Table 95. Headings of output when you request a daily statistics report (continued)

Line number	Daily statistics output heading	Description
12 and 13	HSM FUNCTION	<p>This field contains the names of the functions DFSMSHsm processed. When you specify REPORT DAILY FUNCTION, DFSMSHsm reports the daily statistics in the following order:</p> <ol style="list-style-type: none"> 1. MIGRATION <p>The category SUBSEQUENT MIGS reflects all movements of data sets from ML1 to ML1, ML1 to ML2, and ML2 to ML2. This includes FREEVOL moves but does not include RECYCLE moves. The PRIMARY to LEVEL2 category includes data set migrations by reconnection. The PRIMARY to CLOUD category includes data set migrations by reconnection.</p> 2. RECALL 3. DELETE <p>Deletion of migrated data sets include only those deleted by the DELETEBYAGE (DBA) or DELETEBACKEDUP (DBU) functions from migration volumes, generation data sets that have been rolled-off from migration volumes, and migrated data sets that have expired by date or MCLASS attributes.</p> <p>Deletion of primary data sets include only those deleted by the DELETEBYAGE (DBA) or DELETEBACKEDUP (DBU) functions, generation data sets that have been rolled-off from primary volumes, and primary data sets that have expired by date or MCLASS attributes.</p> <p>The DBA and DBU functions apply only to non-SMS data sets.</p> 4. BACKUP <p>SUBSEQUENT BACKUP results from command backup, daily backup, FREEVOL processing and is the combined total of the following:</p> <ul style="list-style-type: none"> • Data sets moved from backup volumes to spill volumes. • Data sets moved from ML1 volumes to backup volumes. <p>DELETE BACKUP results from command backup, daily backup, BDELETE and EXPIREBV processing and is the combined total of the following:</p> <ul style="list-style-type: none"> • Backup volumes deleted after DFSMSHsm processes the EXPIREBV command • Deletion of excessive incremental backups • Data sets deleted after DFSMSHsm processes the BDELETE command 5. RECOVER 6. RECYCLE 7. CLASS TRANSITION <p>You can also choose to ask DFSMSHsm to report statistics for specific DFSMSHsm functions. For example, when you specify REPORT DAILY FUNCTION RECYCLE(ALL), DFSMSHsm prints a daily statistics report of the number of tape backup and ML2 volumes DFSMSHsm recycled during the time period you specified.</p> <p>MIGRATION and RECALL statistics do not include amounts for extent reductions.</p>
12 and 13	NUMBER DATASETS	This field shows how many data sets DFSMSHsm processed for each function.
12 and 13	-----READ----- TRK/BLK BYTES	This field shows how many tracks or blocks DFSMSHsm read when processing the specific function. For data movement involving DASD, the first part of the field contains the number of tracks. For recycle, the first part of the field contains the number of 16K tape blocks. The last part of the field contains the number of kilobytes DFSMSHsm read when processing the specific function.

Table 95. Headings of output when you request a daily statistics report (continued)		
Line number	Daily statistics output heading	Description
12 and 13	-----WRITTEN----- TRK/BLK BYTES	This field shows how many tracks or blocks DFSMShsm wrote when processing the specific DFSMShsm function. For data movement involving DASD, the first part of the field contains the number of tracks. TRK/BLK represents the number of tracks of compressed data written and BYTES represents the number of bytes written (uncompressed). For recycle, the first part of the field contains the number of 16K tape blocks. The last part of the field contains the number of kilobytes DFSMShsm wrote when processing the specific function.
12 and 13	-----REQUESTS----- SYSTEM USER FAILED	This field shows how many requests came from DFSMShsm during its automatic processing, how many came from the user, and how many of the total requests failed.
12 and 13	AVERAGE AGE	This field contains the average age of the data sets that each function processed. For the MIGRATION and DELETE functions, this is the age of the average data set since it was last referred to. It is calculated by subtracting the date last referred to from the age of migration and dividing the result by the total number of data sets processed. For the RECALL function, this age is how long the average data set has been migrated. For the BACKUP, SPILL, and RECOVER functions, this age is the last time DFSMShsm made a backup version of the data set. The average age does not apply to the RECYCLE function.
12 and 13	-----AVERAGE TIME----- QUEUED WAIT PROCESS TOTAL	This field shows the average time in seconds that the request for each data set was waiting to be processed, the time spent allocating and opening each data set, and the time spent processing each data set. TOTAL shows the average time it took DFSMShsm to process each data set request. For the SUMMARY, the time for each day is accumulated prior to calculating the daily average. The AVERAGE TIME field is the total time for all days divided by the total number of data sets, giving the average time per data set.

If you specify the FROMDATE and TODATE parameters, DFSMShsm prints a summary of all the daily statistics after it prints the daily statistics for each day you specified. If you have requested statistics for specific functions, the report contains statistics for only those functions.

Note: If the letter K, M, G, or T follows any field, then the reported number is represented to the nearest kilobyte (kilotrack), megabyte, gigabyte, or terabyte, respectively. All output will be reported to the smallest possible unit measure. For example:

```
1 KILOBYTE = 1024 bytes
1 KILOTRACK = 1024 tracks
1 MEGABYTE = 1,048,576 bytes
1 GIGABYTE = 1,073,741,824 bytes
1 TERABYTE = 1,099,511,627,776 bytes.
```

```
999,945,993,123 will be represented as 976509759K.
103,999,945,993,123 will be represented as 99182077M.
112,103,999,945,993,123 will be represented as 104404986G.
4,112,103,999,945,993,123 will be represented as 3739937T.
```

The maximum reportable number is 4 611 686 018 427 387 904 or 4194304T. After this value is exceeded, the output reported resets and begins at zero. If it is suspected that this condition exists, rerun the report with a closer specified FROMDATE/TODATE.

Figure 162 on page 1405 is a sample of a daily statistics report that you get when you specify REPORT DAILY FUNCTION FROMDATE(2002/08/19) TODATE(2002/08/20). Therefore, you have the daily statistics for each day and then you have a summary report for the two days.


```
--DFSMShM STATISTICS REPORT ----- AT 15:07:26 ON 2002/08/20 FOR SYSTEM=MVS4
```

```
DAILY STATISTICS REPORT FOR 02/08/19
```

```
STARTUPS=000, SHUTDOWNS=001, ABENDS=000, WORK ELEMENTS PROCESSED=000544, BKUP VOL RECYCLED=00000, MIG VOL RECYCLED=00000
DATA SET MIGRATIONS BY VOLUME REQUEST= 0000446, DATA SET MIGRATIONS BY DATA SET REQUEST= 00091, BACKUP REQUESTS= 0000364
EXTENT REDUCTIONS= 0000000 RECALL MOUNTS AVOIDED= 00101 RECOVER MOUNTS AVOIDED= 00000
FULL VOLUME DUMPS= 000034 REQUESTED, 00000 FAILED; DUMP COPIES= 000034 REQUESTED, 00000 FAILED
FULL VOLUME RESTORES= 000000 REQUESTED, 00000 FAILED; DATASET RESTORES= 000000 REQUESTED, 00000 FAILED
ABACKUPS= 00009 REQUESTED,00000 FAILED; EXTRA ABACKUP MOUNTS=00000
DATA SET MIGRATIONS BY RECONNECTION = 000000, NUMBER OF TRACKS RECONNECTED = 00000000
FAST REPLICATION VOLUME BACKUP = 000000012 REQUESTED, 00000000 FAILED
FAST REPLICATION VOLUME RECOVER = 00000003 REQUESTED, 00000001 FAILED
DATA SET CLASS TRANSITION = 00000000 REQUESTED, 00000000 FAILED
```

HSM FUNCTION	NUMBER DATASETS	-----READ----- TRK/BLK BYTES	-----WRITTEN----- TRK/BLK BYTES	-----REQUESTS----- SYSTEM USER FAILED	AGE	QUEUED	WAIT	PROCESS	TOTAL
MIGRATION									
PRIMARY - LEVEL 1	0000069	00028656	000986175K	00023524	000986498K	000025	000046	00002	00002
SUBSEQUENT MIGS	0000291	00428038	017973894K	00000000	017976272K	000361	000000	00070	00022
PRIMARY 2 - LEVEL 2	0000177	00252337	011726247K	00000000	011731648K	000151	000045	00019	00006
RECALL									
LEVEL 1 - PRIMARY	0000085	00021181	000887920K	00017263	000887577K	000000	000088	00003	00004
LEVEL 2 - PRIMARY	0000139	00000000	007242288K	00137149	007239717K	000000	00139	00000	00018
DELETE									
MIGRATE DATA SETS	0000152	00006850	000497014K	00000000	00000000	000000	00152	00000	00067
PRIMARY DATA SETS	0000035	00002629	000000000	00000000	00000000	000598	00047	00610	00011
BACKUP									
DAILY BACKUP	0000364	00096278	004511279K	00000000	004549328K	000798	00003	00437	00023
SUBSEQUENT BACKUP	0000000	00000000	000000000	00000000	00000000	000000	00000	00000	00000
DELETE BACKUPS	0000293	00000000	000000000	00000000	00000000	000293	00000	00000	00002
RECOVER									
BACKUP - PRIMARY	0000000	00000000	000000000	00000000	00000000	000000	00000	00000	00000
RECYCLE									
BACKUP - SPILL	0000000	00000000		00000000		000000	00000	00000	00000
MIG L2 - MIG L2	0000000	00000000		00000000		000000	00000	00000	00000
CLASS TRANSITION									
PRIMARY - PRIMARY	0000000	00000000	000000000	00000000	00000000	000000	00000	00000	00000

```
DAILY STATISTICS REPORT FOR 02/08/20
```

```
STARTUPS=003, SHUTDOWNS=002, ABENDS=000, WORK ELEMENTS PROCESSED=000105, BKUP VOL RECYCLED=00000, MIG VOL RECYCLED=00000
DATA SET MIGRATIONS BY VOLUME REQUEST= 0000012, DATA SET MIGRATIONS BY DATA SET REQUEST= 00000, BACKUP REQUESTS= 0000723
EXTENT REDUCTIONS= 0000000 RECALL MOUNTS AVOIDED= 00028 RECOVER MOUNTS AVOIDED= 00000
FULL VOLUME DUMPS= 000023 REQUESTED, 00000 FAILED; DUMP COPIES= 000023 REQUESTED, 00000 FAILED
FULL VOLUME RESTORES= 000000 REQUESTED, 00000 FAILED; DATASET RESTORES= 000000 REQUESTED, 00000 FAILED
ABACKUPS= 00000 REQUESTED,00000 FAILED; EXTRA ABACKUP MOUNTS=00000
DATA SET MIGRATIONS BY RECONNECTION = 000000, NUMBER OF TRACKS RECONNECTED = 00000000
FAST REPLICATION VOLUME BACKUP = 000000012 REQUESTED, 00000000 FAILED
FAST REPLICATION VOLUME RECOVER = 00000000 REQUESTED, 00000000 FAILED
DATA SET CLASS TRANSITION = 00000000 REQUESTED, 00000000 FAILED
```

HSM FUNCTION	NUMBER DATASETS	-----READ----- TRK/BLK BYTES	-----WRITTEN----- TRK/BLK BYTES	-----REQUESTS----- SYSTEM USER FAILED	AVERAGE AGE	-----AVERAGE TIME----- QUEUED WAIT PROCESS TOTAL
MIGRATION						
PRIMARY - LEVEL 1	0000006	00000050	000001838K	00000046	000001846K	000006
SUBSEQUENT MIGS	0000000	00000000	000000000	00000000	00000000	000000
PRIMARY 2 - LEVEL 2	0000006	00003810	000124905K	00000000	00012492K	000008
RECALL						
LEVEL 1 - PRIMARY	0000002	00001059	000044410K	00002496	000044396K	000000
LEVEL 2 - PRIMARY	0000061	00000000	004880544K	00139930	004878598K	000000
DELETE						
MIGRATE DATA SETS	0000017	00000118	000000000	00000000	00000000	000000
PRIMARY DATA SETS	0000005	00000645	000000000	00000000	00000000	000567
BACKUP						
DAILY BACKUP	0000723	00383775	017124106K	00000000	017173664K	001170
SUBSEQUENT BACKUP	0000000	00000000	000000000	00000000	00000000	000000
DELETE BACKUPS	0001764	00000000	000000000	00000000	00000000	001764
RECOVER						
BACKUP - PRIMARY	0000000	00000000	000000000	00000000	00000000	000000
RECYCLE						
BACKUP - SPILL	0000000	00000000		00000000		000000
MIG L2 - MIG L2	0000000	00000000		00000000		000000
CLASS TRANSITION						
PRIMARY - PRIMARY	0000000	00000000	000000000	00000000	00000000	000000

Figure 162. Sample report of daily statistics by function

Requesting a report of the volume statistics for all functions

You specify REPORT VOLUMES to get a report of the volume statistics for all volumes managed by DFSMSHsm. You specify VOLUMES with a volume serial number to get a statistics report for a specific volume.

Note: REPORT VOLUMES shows only the last full-volume dump and full-volume restore of each source volume for each day. It will not show multiple dumps or restores of a given source volume on a given day. To find out the total number of full-volume dumps and full-volume restores on a given day, issue the REPORT DAILY command.

Table 96 on page 1406 contains the volume statistics report for all the functions.

Table 96. Headings of output when you request a volume statistics report		
Line Number	Volume Statistics Output Heading	Description
1	VOLUME STATISTICS REPORT FOR VOLUME <i>volser</i> FOR YY/MM/DD	When you specify REPORT VOLUMES, this line contains today's date. When you specify REPORT VOLUMES FROMDATE(<i>yyyy/mm./dd</i>) TODATE(<i>yyyy/mm./dd</i>), this line contains the first date you specify. DFSMSShsm reports all volume statistics for the first date, then reports the volume statistics for the second day, and so forth. This line also shows the volume serial number of the volume whose statistics are being reported.
2	UNIT TYPE= <i>unittype</i>	This field shows the type of unit where the volume is mounted. These fields will have U to represent an L0 non-DFSMSShsm volume.
2	HSM VOLUME TYPE= <i>voltype</i>	This field shows the type of volume whose statistics are being reported. The types of volumes are: PRIMARY, ML1, ML2, DAILY, SPILL and BACKUP.
3	MIGRATED DATA SETS BY VOLUME REQUEST= <i>xxxx</i>	This field shows how many data sets were processed during automatic and command volume migration.
3	DATA SET MIGRATIONS BY DATA SET REQUEST= <i>xxxx</i>	This field shows how many data sets were processed by the DATASETNAME PARAMETER of the MIGRATE command.
3	DATA SETS ON VOLUME= <i>xxxxxx</i>	This field contains the number of non-VSAM data sets that DFSMSShsm has processed. When DFSMSShsm generates a report for primary volumes, this field contains N/A. This field also contains N/A for the summary reports.
4	MINIMUM AGE= <i>xxx</i>	This field contains the age of the youngest data set that migrated from the volume the last time DFSMSShsm did volume migration.
4	TOTAL TRACKS= <i>xxxxxxxx</i>	This field shows the total number of allocated tracks on the volume as of the last free space check prior to the creation of this VSR.
4	FREE TRACKS= <i>xxxxxxxx</i>	This field shows the total number of free tracks available for allocation for new data sets other than suballocated VSAM data sets. This value is accurate as of the last free space check prior to the creation of this VSR.
4	FRAGMENT INDEX= <i>xxx</i>	This field shows the fragmentation index for the volume.
5	VOLUME DUMP= <i>xxxxxxxx</i> ;	This field shows the status of the volume dump. The status <i>xxxxxxxx</i> may be one of the following: <ul style="list-style-type: none"> • DONE • NOT DONE • FAILED
5	DUMP COPIES= <i>xxx</i> REQUESTED	This field shows the number of dump copies requested.
5	FAILED= <i>xxx</i>	This field shows the number of dump copies that failed. You get this version of line 3 only when you issue the command REPORT VOLUMES FUNCTION BACKUP.
6	VOLUME RESTORE= <i>xxxxxxxx</i> ;	This field shows the status of the volume restore. The status <i>xxxxxxxx</i> may be one of the following: <ul style="list-style-type: none"> • DONE • NOT DONE • FAILED
6	DATASET RESTORES = <i>xxxxxx</i> REQUESTED	This field shows the number of data sets requested to be restored.
6	FAILED= <i>xxxxxx</i>	This field shows the number of data sets that failed restoration. You get this version of line 6 only when you issue the command REPORT VOLUMES FUNCTION RECOVER.

Table 96. Headings of output when you request a volume statistics report (continued)

Line Number	Volume Statistics Output Heading	Description
7	DATA SET MIGRATIONS BY RECONNECTION=xxxxx	This field shows the number of data sets reconnected.
7	NUMBER OF TRACKS RECONNECTED=xxxxx	This field shows the number of tracks on level 0 volumes that were freed by reconnection.
7	NUMBER OF TRACKS RECONNECTED TO CLOUD=xxxxx	This field shows the number of tracks on level 0 volumes that were freed by reconnection.
8	FAST REPLICATION BACKUP=xxxxx REQUESTED, xxxxx FAILED	This field shows the number of times that a fast replication backup was attempted for the volume and the number of failed attempts for that volume.
9	FAST REPLICATION RECOVER=xxxxx REQUESTED, xxxxx FAILED	This field shows the number of times that a fast replication recovery was attempted for the volume and the number of failed attempts for that volume.
10	DATA SET CLASS TRANSITION=xxxxx REQUESTED, xxxxx FAILED	This field shows the number of class transitions that were requested and how many of the class transitions failed.

Table 96. Headings of output when you request a volume statistics report (continued)		
Line Number	Volume Statistics Output Heading	Description
11 and 12	HSM FUNCTION	<p>This field contains the names of the functions DFSMSHsm processed. When you specify REPORT VOLUMES FUNCTION, DFSMSHsm reports the volume statistics in the following order:</p> <ol style="list-style-type: none"> 1. MIGRATION <p>The category SUBSEQUENT MIGS reflects all movement of data sets from ML1 to ML1, ML1 to ML2, and ML2 to ML2. This includes FREEVOL moves but does not include RECYCLE moves. The PRIMARY to LEVEL2 category includes data set migrations by reconnection. The PRIMARY to CLOUD category includes data set migrations by reconnection.</p> 2. RECALL 3. DELETE <p>Deletion of migrated data sets include only those deleted by the DELETEBYAGE (DBA) or DELETEBACKEDUP (DBU) functions from migration volumes, generation data sets that have been rolled-off from migration volumes, and migrated data sets that have expired by date or MCLASS attributes.</p> <p>Deletion of primary data sets include only those deleted by the DELETEBYAGE (DBA) or DELETEBACKEDUP (DBU) functions, generation data sets that have been rolled-off from primary volumes, and primary data sets that have expired by date or MCLASS attributes.</p> <p>The DBA and DBU functions apply only to non-SMS data sets.</p> 4. BACKUP <p>SUBSEQUENT BACKUP results from command backup, daily backup, FREEVOL processing and is the combined total of the following:</p> <ul style="list-style-type: none"> • Data sets moved from backup volumes to spill volumes • Data sets moved from ML1 volumes to backup volumes <p>DELETE BACKUP results from command backup, daily backup, BDELETE and EXPIREBV processing and is the combined total of the following:</p> <ul style="list-style-type: none"> • Backup volumes deleted after DFSMSHsm processes the EXPIREBV command • Deletion of excessive incremental backups • Data sets deleted after DFSMSHsm processes the BDELETE command 5. RECOVER 6. RECYCLE 7. CLASS TRANSITION <p>You can also choose to ask DFSMSHsm to report statistics for specific DFSMSHsm functions. For example, when you specify REPORT VOLUME FUNCTION RECOVER, DFSMSHsm prints a volume statistics report of the number of data sets DFSMSHsm recovered during the time period you specified.</p> <p>MIGRATION and RECALL statistics do not include amounts for extent reductions.</p>
11 and 12	NUMBER DATASETS	This field shows how many data sets DFSMSHsm processed for each function.
11 and 12	-----READ----- TRK/BLK K-BYTES	This field is not applicable.
11 and 12	----WRITTEN---- TRK/BLK K-BYTES	This field is not applicable.

Table 96. Headings of output when you request a volume statistics report (continued)		
Line Number	Volume Statistics Output Heading	Description
11 and 12	----REQUESTS---- SYSTEM USER FAILED	This field shows how many requests came from DFSMSHsm during its automatic processing, how many came from the user, and how many of the total requests failed.
11 and 12	AVERAGE AGE	This field contains the average age of the data sets that each function processed. For the MIGRATION and DELETE functions, this is the age of the average data set since it was last referred to. It is calculated by subtracting the date last referred to from the age of migration and dividing the result by the total amount of data sets processed. For the RECALL function, this age is how long the average data set has been migrated. For the BACKUP, SPILL, and RECOVER functions, this age is the last time DFSMSHsm made a backup version of the data set. The average age does not apply to the RECYCLE function.
11 and 12	AVERAGE TIME QUEUED WAIT PROCESS TOTAL	This field shows the average time in seconds that the request for each data set was waiting to be processed, the time spent allocating and opening each data set, and the time spent processing each data set. TOTAL shows the average time it took DFSMSHsm to process each data set request.

If you specify the FROMDATE and TODATE parameters, DFSMSHsm prints a summary of all the volume statistics after it prints the volume statistics for each day you specified. If you have requested statistics for specified functions, the report contains statistics for only those functions.

Figure 163 on page 1410 is a sample of a volume statistics report you get when you specify REPORT VOLUMES FUNCTION FROMDATE(2002/08/16) TODATE(2002/08/175). Therefore, you have the volume statistics for all volumes for each day and then you have a summary report for two days.

Using the REPORT command

```
--DFSMSHSM STATISTICS REPORT ----- AT 15:10:12 ON 2002/08/20 FOR SYSTEM=MVS4  
  
VOLUME STATISTICS REPORT FOR VOLUME MIG410 FOR 02/08/16  
  
UNIT TYPE = 3390      , HSM VOLUME TYPE = LEVEL 1  
MIGRATED DATA SETS BY VOLUME REQUEST=000000, DATA SET MIGRATIONS BY DATA SET REQUEST=000045, DATA SETS PROCESSED=0000216  
MINIMUM AGE = 000, TOTAL TRACKS = 00056381, FREE TRACKS = 00057619, FRAGMENT INDEX = .021  
VOLUME DUMP= NOT DONE; DUMP COPIES=000000 REQUESTED, FAILED=000000  
VOLUME RESTORE= NOT DONE; DATASET RESTORES=000000 REQUESTED, FAILED=000000  
DATA SET MIGRATIONS BY RECONNECTION = 000000 , NUMBER OF TRACKS RECONNECTED =00000000  
FAST REPLICATION BACKUP = 00001 REQUESTED, 00000 FAILED  
FAST REPLICATION RECOVER = 00000 REQUESTED, 00000 FAILED  
DATA SET CLASS TRANSITION = 00000 REQUESTED, 00000 FAILED
```

HSM FUNCTION	NUMBER DATASETS	-----READ----- TRK/BLK K-BYTES	-----WRITTEN----- TRK/BLK K-BYTES	-----REQUESTS----- SYSTEM USER FAILED	AVERAGE AGE	-----AVERAGE TIME----- QUEUED WAIT PROCESS TOTAL
MIGRATION						
PRIMARY - LEVEL 1	0000045	00000000 000000000	00006687 000279848	000000 00045 00000	00000	0005 00001 00001 00007
SUBSEQUENT MIGS	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
PRIMARY - LEVEL 2	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
RECALL						
LEVEL 1 - PRIMARY	0000125	00018700 000782550	00000000 000000000	000000 00125 00000	00000	0000 00000 00002 00002
LEVEL 2 - PRIMARY	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
DELETE						
MIGRATE DATA SETS	0000046	00006746 000279848	00000000 000000000	000000 00046 00000	00000	0000 00000 00000 00000
PRIMARY DATA SETS	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
BACKUP						
DAILY BACKUP	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
SUBSEQUENT BACKUP	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
DELETE BACKUPS	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
RECOVER						
BACKUP - PRIMARY	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
RECYCLE						
BACKUP - SPILL	0000000	00000000	00000000	000000 00000 00000	00000	0000 00000 00000 00000
MIG L2 - MIG L2	0000000	00000000	00000000	000000 00000 00000	00000	0000 00000 00000 00000
CLASS TRANSITION						
PRIMARY - PRIMARY	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000

VOLUME STATISTICS REPORT FOR VOLUME MIG410 FOR 02/08/17

```
UNIT TYPE = 3390      , HSM VOLUME TYPE = LEVEL 1  
MIGRATED DATA SETS BY VOLUME REQUEST=000000, DATA SET MIGRATIONS BY DATA SET REQUEST=000035, DATA SETS PROCESSED=0000155  
MINIMUM AGE = 000, TOTAL TRACKS = 00056749, FREE TRACKS = 00057251, FRAGMENT INDEX = .021  
VOLUME DUMP= NOT DONE; DUMP COPIES=000000 REQUESTED, FAILED=000000  
VOLUME RESTORE= NOT DONE; DATASET RESTORES=000000 REQUESTED, FAILED=000000  
DATA SET MIGRATIONS BY RECONNECTION = 000000 , NUMBER OF TRACKS RECONNECTED =00000000  
FAST REPLICATION BACKUP = 00001 REQUESTED, 00000 FAILED  
FAST REPLICATION RECOVER = 00001 REQUESTED, 00000 FAILED  
DATA SET CLASS TRANSITION = 00000 REQUESTED, 00000 FAILED
```

HSM FUNCTION	NUMBER DATASETS	-----READ----- TRK/BLK K-BYTES	-----WRITTEN----- TRK/BLK K-BYTES	-----REQUESTS----- SYSTEM USER FAILED	AVERAGE AGE	-----AVERAGE TIME----- QUEUED WAIT PROCESS TOTAL
MIGRATION						
PRIMARY - LEVEL 1	0000035	00000000 000000000	00004959 000207462	000000 00035 00000	00000	0001 00001 00002 00004
SUBSEQUENT MIGS	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
PRIMARY - LEVEL 2	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
RECALL						
LEVEL 1 - PRIMARY	0000085	00012716 000532134	00000000 000000000	000000 00085 00000	00000	0000 00000 00002 00002
LEVEL 2 - PRIMARY	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
DELETE						
MIGRATE DATA SETS	0000035	00004959 000207462	00000000 000000000	000000 00035 00000	00000	0000 00000 00000 00000
PRIMARY DATA SETS	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
BACKUP						
DAILY BACKUP	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
SUBSEQUENT BACKUP	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
DELETE BACKUPS	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
RECOVER						
BACKUP - PRIMARY	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000
RECYCLE						
BACKUP - SPILL	0000000	00000000	00000000	000000 00000 00000	00000	0000 00000 00000 00000
MIG L2 - MIG L2	0000000	00000000	00000000	000000 00000 00000	00000	0000 00000 00000 00000
CLASS TRANSITION						
PRIMARY - PRIMARY	0000000	00000000 000000000	00000000 000000000	000000 00000 00000	00000	0000 00000 00000 00000

Figure 163. Sample report of volume statistics by function

Requesting a summary of the daily or volume statistics

You specify REPORT DAILY FUNCTION SUMMARY to get the summary totals of the daily statistics for all functions. You specify REPORT VOLUME FUNCTION SUMMARY to get the summary totals of the volume statistics for all volumes and all functions. You specify the VOLUMES parameter with a volume serial number to get summary statistics for only one volume. If you request a summary of specified functions, the summary contains the statistics totals for only the specified functions.

When you specify SUMMARY, you get only the summary report. If you do not specify SUMMARY, you get the statistics report you requested and a summary at the end of the report if DFSMSHsm reports the statistics for more than one day. A sample of the summary statistics report is not shown because the

information is in the same format as the summary information shown in [Figure 162 on page 1405](#) and [Figure 163 on page 1410](#).

Note: For an equivalent time period, it is possible that the number of data sets reported by the REPORT VOLUME command and the number of data sets reported by the REPORT DAILY command will not be the same on the summary page, under the heading "Number Datasets". Some functions (such as MIGRATION, RECALL, BACKUP, RECOVER, and SPILL) involve two volumes, while other functions do not. As each volume counts each data set action, some data set actions will be counted more than once.

Appendix A. Accessibility

Accessible publications for this product are offered through [IBM Documentation for z/OS \(www.ibm.com/docs/en/zos\)](http://www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see [How to Send Feedback to IBM](#) to leave documentation feedback.

Notices

This information was developed for products and services that are offered in the USA or elsewhere.

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

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

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

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

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation
Site Counsel
2455 South Road*

Poughkeepsie, NY 12601-5400
USA

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

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, and MVS, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those

products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: [IBM Lifecycle Support for z/OS \(www.ibm.com/software/support/systemsz/lifecycle\)](http://www.ibm.com/software/support/systemsz/lifecycle)
- For information about currently-supported IBM hardware, contact your IBM representative.

Programming interface information

This information primarily contains information that is NOT intended to be used as Programming Interfaces of DFSMSHsm.

This information also contains intended Programming Interfaces that allow the customer to write programs to obtain the services of DFSMSHsm. This information is identified where it occurs, either by an introductory statement to a topic or by the following marking:

Programming Interface Information
End Programming Interface Information

Trademarks

IBM, the IBM logo, and [ibm.com](http://www.ibm.com) are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [Copyright and Trademark information \(www.ibm.com/legal/copytrade.shtml\)](http://www.ibm.com/legal/copytrade.shtml).

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Glossary

The terms in this glossary are defined as they pertain to technical terms and abbreviations used in support of DFSMSHsm.

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published parts of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

3480

An IBM 3480 Magnetic Tape Subsystem device.

3480X

A 3480 XF device or an IBM 3490 Magnetic Tape Subsystem device.

3490

An IBM 3490 Magnetic Tape Subsystem device.

3490E

See 3490.

3590

An IBM TotalStorage Enterprise Tape Drive 3590. See 3590-B1x, 3590-E1x and 3590-H1x.

3590B1x

An IBM TotalStorage Enterprise Tape Drive 3590 Model B1x.

3590E1x

An IBM TotalStorage Enterprise Tape Drive 3590 Model E1x.

3590H1x

An IBM TotalStorage Enterprise Tape Drive 3590 Model H1x.

3592E05

An IBM TotalStorage Enterprise Tape Drive 3592.

3592E06

IBM System Storage Tape System 3592 Model E06.

3592E07

An IBM Enterprise Tape Drive 3592 Model E07.

3592J1A

An IBM TotalStorage Enterprise Tape Drive 3592.

ABARS

Aggregate backup and recovery support. See *aggregate backup* and *aggregate recovery*.

ABEND

Abnormal end.

ACB

Access method control block.

accompany data set

In aggregate backup and recovery processing, a data set that is physically transported from the backup site to the recovery site instead of being copied to the aggregate data tape. It is cataloged during recovery.

ACCOMPANY keyword

The keyword used in the selection data set to create an accompany list.

accompany list

An optional list in the selection data set that identifies the accompany data sets.

ACEE

Access control environment element.

ACS

See *automatic class selection (ACS) routine*.

active copies

A set of backup copies created that have not yet rolled off. The number of active copies is determined by the SMS management class or SETSYS value. The maximum number of active copies will remain one hundred.

active data

Data that is frequently accessed by users and that resides on level 0 volumes.

activity log

In DFSMSHsm, a SYSOUT or DASD-type data set used to record activity and errors that occurred during DFSMSHsm processing.

AG

See *aggregate group (AG)*.

aggregate backup

The process of copying the data sets and control information of a user-defined group of data sets so that they may be recovered later as an entity by an aggregate recovery process.

aggregate data sets

In aggregate backup and recovery processing, data sets that have been defined in an aggregate group as being related.

aggregate group (AG)

A Storage Management Subsystem class that defines control information and identifies the data sets to be backed up by a specific aggregate backup.

aggregate recovery

The process of recovering a user-defined group of data sets that were backed up by aggregate backup.

AIX

See *alternate index (AIX)*.

allocate data set

In aggregate backup and recovery processing, a data set name that is listed in the selection data set. The space for this data set is allocated and the data set is cataloged at the recovery location, but the actual data is not restored.

ALLOCATE keyword

The keyword used in the selection data set to create an allocate list.

allocate list

An optional list in the selection data set that identifies the allocate data sets.

alternate index (AIX)

In systems with VSAM, a collection of index entries related to a given base cluster and organized by an alternate key, that is, a key other than the prime key of the associated base cluster data records. Its function is to provide an alternate directory for locating records in the data component of a base cluster. See *path*.

alternate index cluster

In VSAM, the data and index components of an alternate index.

alternate tape volumes

In DFSMSHsm, copies of original tape volumes created during tape copy processing. The volumes can either be stored on-site or off-site for use later in the event of a disaster. During the tape replace processing, these volumes can replace the original volumes that may be lost.

APAR

Authorized program analysis report.

APF

Authorized program facility.

ASID

Address space identifier.

ATL

See *automated tape library (ATL)*.

audit

A DFSMSHsm process that detects discrepancies between data set information in the VTOCs, the computing system catalog, the MCDS, BCDS, and OCDS.

AUTH

The DFSMSHsm command used to identify an authorized user who can issue DFSMSHsm system programmer and storage administrator commands.

authorized user

In DFSMSHsm, the person or persons who are authorized through the DFSMSHsm AUTH command to issue DFSMSHsm system programmer, storage administrator, and operator commands.

automated tape library (ATL)

A device consisting of robotic components, cartridge storage frames, tape subsystems, and controlling hardware and software, together with the set of volumes which reside in the library and may be mounted on the library tape drives. See *manual tape library (MTL)*.

automatic backup

In DFSMSHsm, the process of automatically copying eligible data sets from DFSMSHsm-managed volumes or migration volumes to backup volumes during a specified backup cycle.

automatic cartridge loader feature

A feature of the 3480 Magnetic Tape Subsystem providing the operator with the capability of preloading multiple tapes to be used as migration, backup, or dump tapes.

automatic class selection (ACS) routine

A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

automatic dump

In DFSMSHsm, the process of using DFSMSdss to automatically do a full volume dump of all allocated space on DFSMSHsm-managed volumes to designated tape dump volumes.

automatic primary space management

In DFSMSHsm, the process of automatically deleting expired data sets, deleting temporary data sets, releasing unused overallocated space, and migrating data sets from DFSMSHsm-managed volumes.

automatic secondary space management

In DFSMSHsm, the process of automatically deleting expired migrated data sets from the migration volumes, deleting expired records from the migration control data set, and migrating eligible data sets from level 1 volumes to level 2 volumes.

automatic space management

In DFSMSHsm, includes automatic volume space management, automatic secondary space management, and automatic recall.

automatic volume space management

In DFSMSHsm, includes automatic primary space management and automatic interval migration.

availability management

In DFSMSHsm, the process of ensuring that a current version (backup copy) of the installation's data sets resides on tape or DASD.

backup

In DFSMSHsm, the process of copying a data set residing on a level 0 volume, a level 1 volume, or a volume not managed by DFSMSHsm to a backup volume. See *automatic backup* and *incremental backup*.

backup control data set (BCDS)

A VSAM, key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMSHsm.

backup copy

In DFSMSHsm, a copy of a data set that is kept for reference in case the original data set is destroyed.

backup cycle

In DFSMSHsm, a period of days for which a pattern is used to specify the days in the cycle on which automatic backup is scheduled to take place.

backup frequency

In DFSMSHsm, the number of days that must elapse since the last backup version of a data set was made until a changed data set is again eligible for backup.

backup profile

In DFSMSHsm, a RACF discrete data set profile associated with the backup version of a cataloged data set that is protected by a RACF discrete data set profile.

backup version

Synonym for backup copy.

backup volume

A volume managed by DFSMSHsm to which backup versions of data sets are written.

backup volume cleanup process

A DFSMSHsm process that scratches data set backup versions that are no longer needed on DASD.

backup VTOC copy data set

In DFSMSHsm, a copy of the VTOC of a volume that was backed up by DFSMSHsm. This VTOC data set contains only part of the data set VTOC entry for each data set from the original data set. This data set is written on a migration level 1 volume.

base cluster

In systems with VSAM, a key-sequenced or entry-sequenced file over which one or more alternate indexes are built. See *cluster*.

base data component

In VSAM, a component of the base cluster containing data of a data set.

base sysplex

A base (or basic) sysplex is the set of one or more z/OS systems that are given a cross-system coupling facility (XCF) name and in which the authorized programs can then use XCF coupling services. A base system does not include a coupling facility. See *parallel sysplex* and *sysplex*.

basic format

The format of a data set that has a data set name type (DSNTYPE) of BASIC. A basic format data set is a sequential data set that is specified to be neither large format nor extended format. The size of a basic format data set cannot exceed 65 535 tracks on each volume.

BCDS

Backup control data set.

BCR

Backup control record.

BDAM

Basic direct access method.

BSAM

Basic sequential access method.

cartridge system tape

The base tape cartridge media used with 3480 or 3490 magnetic tape subsystems. See also *Enterprise Tape Cartridge*.

catalog

(1) A data set that contains extensive information required to locate other data sets, to allocate and deallocate storage space, to verify the access authority of a program or operator, and to accumulate data set usage statistics. A catalog has a basic catalog structure (BCS) and its related volume tables of contents (VTOCs) and VSAM volume data sets (VVDSs). (A) (ISO) (2) To enter information about a file or a library into a catalog. (A) (ISO) (3) The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set. (4) To include the volume identification of a data set in the catalog.

CDS

Control data set.

CDSQ serialization

A technique that serializes DFSMSHsm CDSs with a global enqueue product (such as GRS).

CDSR serialization

A technique that serializes DFSMSHsm CDSs with volume reserves.

CDT

Class descriptor table.

CF

Coupling facility.

changed data set

In DFSMSHsm, a data set that has been opened for other than read-only access.

CLIST

Command list.

cluster

In systems with VSAM, a named structure consisting of a group of related components, for example, a data component with its index component. See also *base cluster*.

command list

A command procedure containing executable sequences of TSO commands, subcommands, and command procedure statements.

command procedure

In TSO, a data set or a member of a partitioned data set containing TSO commands to be performed sequentially by the EXEC command. See also *command list*.

common data set descriptor record

The record that precedes a user's data set on a DFSMSHsm-owned volume and that is used to return the data set to the user's format.

common filter services

A subcomponent of DFP common services. Common filter services compares data items with filter keys and indicates which data items match the keys and how many matches have been found.

common recall queue (CRQ)

A single recall queue shared by multiple DFSMSHsm hosts that enables the recall workload to be balanced across each of those hosts.

common service area (CSA)

One of the data areas in the system that is addressable by all address spaces, in which any address space can request a space that is protected during use by the requestor's storage protection key.

compaction

In DFSMSHsm, a method of compressing and encoding data that is migrated or backed up.

comprehensive command authority

In DFSMSHsm, allowing a user to issue the ABACKUP command for all aggregate groups.

compress

In DFSMSHsm, to release unused space in a partitioned data set during the migrate/recall and backup/recovery processes.

computing system catalog

In DFSMSHsm, the master catalog and any associated user catalogs used as sources during the audit process.

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 typical application program processing.

connected set

A group of tapes volumes that are related to each other because one or more data sets span the physical tape volumes. Or, a single tape volume with a data set or data sets that do not span to any other volumes.

contiguous space

An unbroken consecutive series of storage locations.

control data set

In DFSMSHsm, one of three data sets (BCDS, MCDS, and OCDS) that contain records used in DFSMSHsm processing.

control file

In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the catalog, allocation, volume, and related information necessary to perform aggregate recovery.

converter/interpreter processing

The job segment that converts and interprets JCL for the operating system.

copy pool

A named set of storage groups that contains data to be copied and recovered collectively.

copy pool backup storage group

A type of SMS storage group that contains all of the volumes that DFSMSHsm might select as the target of a fast replication.

coupling facility

A special logical partition that provides high-speed caching, list processing, and locking functions in a sysplex.

cross-system coupling facility (XCF)

A component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

CRQ

Common recall queue.

CRQplex

One or more DFSMSHsm hosts that are connected to the same CRQ. There may be multiple CRQplexes within a single HSMplex.

CSA

Common service area.

CSECT

Control section.

CTC

Channel-to-channel.

current backup version

In DFSMSHsm, a backup copy of the data set that was created on a date after the data set was last updated.

CVT

Communications vector table.

cycle start date

In DFSMSHsm, the date a backup cycle, dump cycle, or migration cleanup cycle is started.

DADSM

Direct Access Device Space Management.

DAE

Dump analysis elimination.

daily backup volume

In DFSMSHsm, a volume associated with a given day in the backup cycle and assigned to contain backup versions created on that cycle day.

DAIR

Dynamic allocation interface return code.

DASD

Direct access storage device. See *direct access storage device (DASD)*.

DASD calculation services (DCS)

A subcomponent of DFP common services. DCS retrieves and calculates data set information for both VSAM and non-VSAM data sets based on the user's input request.

DASD volume

A DASD space identified by a common label and accessed by a set of related addresses. Also referred to generically as "volume" in some instances.

data class

A collection of allocation and space attributes, defined by the storage administrator, that is used to create a data set.

data control block (DCB)

A control block used by access method routines in storing and retrieving data.

Data Facility Sort (DFSORT)

An IBM-licensed program that is a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, and copying operations, as well as providing versatile data manipulation at the record, field, and bit level.

Data Facility Storage Management Subsystem (DFSMS)

An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

data file

In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the backup copies of the data sets to be recovered.

data set

The major unit of data storage and retrieval in z/OS DFSMS, consisting of a collection of data in a prescribed arrangement and described by control information to which the system has access.

data migration

See *migration*.

data set change indicator

A bit in the Format 1 data set VTOC entry that indicates whether the data set has been opened for output.

data set deletion

In DFSMSHsm, the space management technique of deleting non-SMS-managed data sets that have not been used for a specified number of days and that do not have expiration date protection.

data set group

Data sets that have the same set of initial characters in their names.

data set organization

The type of arrangement of data in a data set. Examples are sequential organization or partitioned organization.

data set pool

One or more volumes managed by DFSMSHsm to which data sets that have migrated can be recalled, depending on the set of initial characters of the data set name.

data set retirement

In DFSMSHsm, the space management technique of deleting non-SMS-managed data sets that have not been referred to for a specified number of days, and that have a backup version. See *delete-if-backed-up (DBU)*.

date last referred to

In DFSMSHsm, the last date when a data set was opened.

DBA

Delete-by-age.

DBU

Delete-if-backed-up.

DCB

Data control block.

DCR

Dump control record.

DCS

DASD calculation services.

debug mode

In DFSMSHsm, the method of operation that projects the changes that would occur in normal operation but in which no user data moves.

decompaction

In DFSMSHsm, the process of decoding and expanding data that was compacted during daily space management or backup.

delete-by-age (DBA)

In DFSMSHsm, the space management technique of deleting non-SMS-managed data sets that have not been opened for a specified number of days.

delete-if-backed-up (DBU)

In DFSMSHsm, the space management technique of deleting non-SMS-managed data sets that have not been opened for a specified number of days, and that have a current backup version. See *data set retirement*.

demotion

The process of one host losing level functions to another. Both original and promoted hosts can go through the demotion process. See *promotion*.

DFP

Data Facility Product.

DFP common services

A component of DFP that contains three subcomponents: common filter services (CFS), DASD calculation services (DCS), and device information services (DIS).

DFSMS

See *Data Facility Storage Management Subsystem (DFSMS)*.

DFSMSdfp

DFSMS data facility product. A z/OS DFSMS functional component and a base element of z/OS that provides functions for storage management, data management, device management, and distributed data access.

DFSMSdss

DFSMS data set services. A z/OS DFSMS functional component that provides functions for copying, moving, dumping, and restoring data sets and volumes.

DFSMSHsm

DFSMS hierarchical storage manager. A z/OS DFSMS functional component that provides functions for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMSHsm-authorized user

The person or persons who are authorized through the DFSMSHsm AUTH command to issue system programmer and storage administrator commands.

DFSMSHsm log

In DFSMSHsm, a pair of sequential data sets, X and Y, containing a chronological list of transactions and statistics occurring in DFSMSHsm.

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. See *system-managed volume* and *DFSMSrmm-managed volume*.

DFSMSHsm-owned volumes

A storage volume on which DFSMSHsm stores backup versions, dump copies, or migrated data sets.

DFSMSHsm secondary address space

A separate address space started and controlled by DFSMSHsm to perform aggregate backup or aggregate recovery processing.

DFSMSrmm

DFSMS removable media manager. A z/OS DFSMS functional component that provides functions for managing removable media.

DFSMSrmm-managed volume

A tape volume that is defined to DFSMSrmm. See *system-managed volume* and *DFSMSHsm-managed volume*.

DFSORT

See *Data Facility Sort (DFSORT)*.

direct access storage device (DASD)

A device in which the access time is effectively independent of the location of data.

directed recall

Moving a migrated data set from a level 1 or a level 2 volume to a level 0 volume and specifying the target volume and unit name where the data set can be allocated.

disaster

An unplanned occurrence that keeps a company or organization from conducting its normal business for some time period.

disaster backup

A means to protect a computing system complex against data loss in the event of a disaster.

disaster recovery

A means to replace lost data at another location with sufficient resources in order to resume operation.

discrete backup profile

A RACF profile created when DFSMSHsm backs up a cataloged, RACF-indicated data set.

discrete profile

A RACF profile that contains security information about a specific data set on a particular volume.

disposition processing

In OS/VS, a function performed by the initiator at the end of a job step to keep, delete, catalog, or uncatalog data sets, or to pass them to a subsequent job step, depending on the data set status of the disposition specified in the DISP parameter of the DD statement.

Distributed System License Option (DSLO)

A license option available to IBM customers with a basic license that permits them to copy certain IBM-licensed materials for the purpose of installing multiple systems.

DSLO

Distributed Systems License Option.

DSR

Daily statistics record.

dump

See *full volume dump*.

dump class

A DFSMSHsm-named set of characteristics that describe how volume dumps are managed.

dump copy

In DFSMSHsm, a copy of the volume image produced by the DFSMSdss full volume dump function.

dump cycle

In DFSMSHsm, a period of days for which a pattern is used to specify the days in the cycle on which automatic full volume dump is scheduled to take place.

dump generation

A successful full volume dump of a volume that may contain one to five identical dump copies.

dump VTOC copy data set

In DFSMSHsm, a copy of the VTOC of a volume dumped by DFSMSHsm. This VTOC data set contains only part of the data set VTOC entry for each data set from the original data set. This data set is written on a migration level 1 volume.

DVT

Device vector table.

EA

Extended addressability.

EATTR

Keyword controlling whether the allocation of EAS-eligible data sets can reside in cylinder-managed space. Can be used as follows:

- In the ALLOCATE or DEFINE CLUSTER commands, see [z/OS DFSMS Access Method Services Commands](#).
- As a JCL keyword, see [z/OS MVS JCL Reference](#).
- On the ISMF Data Class Define pane, see [z/OS DFSMSdfp Storage Administration](#).

EBCDIC

Extended Binary Coded Decimal Interchange Code.

ECB

Event control block.

EEMT2

Enterprise encrypted format 2 recording technology.

EEMT3

Enterprise encrypted format 3 recording technology.

EEMT4

Enterprise encrypted format 4 recording technology.

EFMT1

Enterprise format 1 recording technology.

EFMT2

Enterprise format 2 recording technology.

EFMT3

Enterprise format 3 recording technology.

EFMT4

Enterprise format 4 recording technology.

eligibility age

The number of days since a data set met its criteria to be migrated.

emergency mode

In DFSMSHsm, the method of operation that prevents data set movement and deletion in space management, backup, and recovery processes.

encode

(T) To convert data by the use of a code in such a manner that reconversion to the original form is possible.

Enterprise Systems Architecture/390 (ESA/390)

An IBM architecture for mainframe computers and peripherals. Processor systems that follow the ESA/390 architecture include the ES/9000 family. See also *z/Architecture*®.

Enterprise Systems Connection (ESCON)

A peripheral interface for an Enterprise Systems Architecture/390 and zSeries computer. The I/O interface uses ESA/390 logical protocols over a serial interface that configures attached units to a communication fabric.

Enterprise Tape Cartridge

Cartridge system tape with increased capacity that can only be used with 3592 Magnetic Tape Subsystems (MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, MEDIA10, MEDIA11, MEDIA12, and MEDIA13). See *cartridge system tape*.

erase-on-scratch

A security function that overwrites the space occupied by a data set when the data set is scratched from a DASD device.

ESA/390

See *Enterprise Systems Architecture/390 (ESA/390)*.

ESCON

See *Enterprise Systems Connection (ESCON)*.

ESDS

Entry-sequenced data set.

esoteric unit names

The names a user assigns to I/O devices that have the same device type. When the user specifies the assigned unit name to DFSMSHsm, DFSMSHsm associates the unit name to its device type.

ESTAE

Extended subtask ABEND exit.

ESTAI

Extended subtask ABEND intercept.

event-driven migration

In DFSMSHsm, a space management function similar to primary space management, except it is event-driven instead of scheduled. Low activity data sets are moved from the volume and occupied space is reduced until the volume low threshold is reached. There are two methods of event-driven migration: hourly (interval migration) and immediate (on-demand migration). See *interval migration* and *on-demand migration*.

exclude data set

In aggregate backup and recovery processing, a data set in the selection data set exclude list. This data set is to be excluded from being processed by aggregate backup.

EXCLUDE keyword

The keyword used in the selection data set to create an exclude list.

exclude list

An optional list in the selection data set that identifies those data sets that are to be excluded from being processed by aggregate backup.

expiration

The removal of a user data set from either a user (non-DFSMSHsm-owned) volume, or from a DFSMSHsm-owned volume when the user data set has been migrated. If there is an explicit expiration date, it is found in the data set VTOC entry for a nonmigrated data set, or in the MCD record for a migrated data set. If there is no explicit expiration date, the management class attributes are checked to determine an implicit expiration date.

extended addressability

A type of extended format VSAM data set that allows greater than 4GB of data storage.

extended format compressed data set

A SAM or VSAM data set whose processing detects errors caused by an incomplete transfer of a physical block of data to a physical record in the data set. Each physical record is ended by a “hidden” suffix. SAM or VSAM processing determines, by examining these suffixes, if a data transfer to a physical record has ended prematurely.

extended remote copy (XRC)

A hardware- and software-based remote copy service option that provides an asynchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

extent reduction

In DFSMSHsm, the releasing of unused space, reducing the number of extents, and compressing partitioned data sets.

extents

A continuous space on a direct access storage volume, occupied by or reserved for a particular data set, data space, or file.

facility class profile

A way that RACF protects all DFSMSHsm command access. An active RACF facility class establishes the security environment.

fallback

Pertaining to returning to use of an earlier release of a program after a later release has been installed and used.

fast replication

The functions that create an instant data copy. The IBM System Storage DS8000 FlashCopy function is an example of fast replication.

fast replication target volume

A DFSMSHsm-owned volume that contains the fast replication backup copy of a DFSMSHsm-managed volume. Fast replication target volumes are contained within SMS copy pool backup storage groups.

fast reverse restore

A FlashCopy option that reverses a FlashCopy relationship without waiting for the completion of the background copy of the previous FlashCopy.

fast subsequent migration

The function that reconnects a data set to the same ML2 tape volume from which it was recalled instead of migrating the data set again. The data set must satisfy a set of requirements when recalled and when a fast subsequent migration reconnect is attempted.

fast subsequent migration reconnect

The action that is attempted when fast subsequent migration is performed on a recalled data set. This is also referred to in context as "reconnect", "reconnecting", and "reconnection."

FBA

Fixed-block architecture.

FIFO

First in, first out.

fixed-block architecture

Data stored in blocks of fixed size; these blocks are addressed by block number relative to the beginning of the particular file.

FlashCopy

An optional feature of the Storage System DS family that can make an instant copy of data, that is, a point-in-time copy of a volume.

FMID

Function modification identifier.

fragmentation index

The qualitative measure of the scattered free space on a volume.

FSR

Functional statistics record.

full volume dump

In DFSMSHsm, the process of using a DFSMSdss function that backs up the entire allocated space on a volume.

full volume restore

In DFSMSHsm, the process of using a DFSMSdss function that restores the entire volume image.

functional statistics record

A record that is created each time a DFSMSHsm function is processed. It contains a log of system activity and is written to the system management facilities (SMF) data set.

functional verification procedure

A procedure distributed with DFSMSHsm that tests to verify that all basic DFSMSHsm functions are working correctly.

GDG

Generation data group.

GDS

Generation data set.

general pool

In a DFSMSHsm environment with JES3, the collection of all DFSMSHsm primary volumes added to that processor that have a mount status of permanently-resident or reserved, that have the automatic recall attribute specified, and that have a mount attribute of storage or private.

generation data group

A collection of data sets with the same base name, such as PAYROLL, that are kept in chronological order. Each data set is called a generation data set.

generic profile

A RACF profile that contains security information about multiple data sets, users, or resources that may have similar characteristics and require a similar level of protection.

global scratch pool

A group of empty tapes that do not have unique serial numbers and are not known individually to DFSMSHsm. The tapes are not associated with a specific device. See *specific scratch pools*.

GRS

Global resource serialization.

GRSplex

One or more z/OS systems using global serialization to serialize access to shared resources.

GTF

Generalized trace facility.

hardware/software compression facility

A function for both SAM and VSAM extended format compressed data sets that has the ability to use either hardware data compression for best performance, or software data compression for DASD space considerations.

high threshold of occupancy

In DFSMSHsm, the upper limit of space to be occupied on a volume managed by DFSMSHsm. See *low threshold of occupancy*.

HMT

HSM Monitor/Tuner.

HSMplex

One or more DFSMSHsm hosts that share a set of control data sets. The hosts may be on single or multiple z/OS images.

IBM System Storage

Follow-on product line to the IBM TotalStorage products.

IBM System Storage and TotalStorage Enterprise Tape Cartridge

A Cartridge system tape with increased capacity that can only be used with 3592 Magnetic Tape Subsystems (MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, and MEDIA10).

ICETOOL

DFSORT's multipurpose data processing and reporting utility.

IDRC

Improved data recording capability.

improved data recording capability

An improved data recording mode that, depending on data set characteristics and machine model, may allow a significant increase in effective cartridge data capacity and performance.

inactive age

In DFSMSHsm, the number of days since the data set was last referred to.

inactive data

Copies of active or low-activity data that reside on DFSMSHsm-owned dump and incremental backup volumes. See *low-activity data*.

include data set

In aggregate backup and recovery processing, a data set in the selection data set include list. This data set is processed by aggregate backup.

INCLUDE keyword

The keyword used in the selection data set to create an include list.

include list

In the selection data set, the list of include data sets to be processed by aggregate backup.

incremental backup

In DFSMSHsm, the process of copying a data set that has been opened for other than read-only access since the last backup version was created, and that has met the backup frequency criteria.

incremental recovery

Recovery of the latest backup copy of a data set or data sets made by incremental backup.

inline backup

The process of copying a specific data set to a migration level 1 volume from a batch environment. This process allows you to back up data sets in the middle of a job.

installation verification procedure (IVP)

A procedure distributed with DFSMSHsm that tests to verify that the basic facilities of DFSMSHsm are functioning correctly.

installation exit

A means specified by an IBM software product by which a customer's system programmers may modify or extend the functions of the IBM software product. Such modifications or extensions consist of exit routines written to replace an existing module of an IBM software product, or to add one or more modules or subroutines to an IBM software product for the purpose of modifying the functions of the IBM software product.

instruction data set

In aggregate backup and recovery processing, a data set that contains instructions, commands, or any data the aggregate backup site defines as needed for aggregate recovery at the recovery site.

instruction file

In aggregate backup and recovery processing, one of three aggregate files generated by the aggregate backup process. It contains the instruction data set.

Interactive Storage Management Facility (ISMF)

The interactive panels of DFSMSdftp that allow users and storage administrators access to the storage management functions of DFSMSdss and DFSMSHsm.

Interactive System Productivity Facility (ISPF)

An IBM licensed program used to develop, test, and run application programs interactively. ISPF is the interactive access method for all storage management functions.

interval migration

A space management function that performs hourly space management on eligible SMS-managed and non-SMS-managed volumes that are at or have exceeded a threshold level (such as the midpoint between the high and low threshold or the high threshold). Data sets are migrated from the eligible

volume until the low threshold is reached or there are no more eligible data sets. See [“Using interval migration”](#) on page 25.

IPCS

Interactive problem control system.

ISMF

See *Interactive Storage Management Facility (ISMF)*.

ISPF

See *Interactive System Productivity Facility (ISPF)*.

ISPF/PDF

Interactive System Productivity Facility/Program Development Facility.

IVP

Installation verification procedure.

JCL

Job control language.

JES

Job entry subsystem.

JES2

Job entry subsystem 2.

JES3

Job entry subsystem 3.

JFCB

Job file control block.

journal data set

In DFSMSHsm, a sequential data set used by DFSMSHsm for recovery of the MCDS, BCDS, and OCDS. The journal contains a duplicate of each record in the control data sets that has changed since the MCDS, BCDS, and OCDS were last backed up.

KB

Kilobyte; 1024 bytes.

key-range data set

A key-sequenced data set that is defined with one or more KEYRANGE groups.

KSDS

Key-sequenced data set.

large format

The format of a data set that has a data set name type (DSNTYPE) of LARGE. A large format data set has the same characteristics as a sequential (non-extended format) data set, but its size on each volume can exceed 65 535 tracks. There is no minimum size requirement for a large format data set.

LE

Logical end.

level functions

(1) Functions that must be performed but need be performed by only one processing unit. Level functions include backing up the control data sets, backing up migrated data sets, deleting expired dump copies, moving backup versions from level 1 volumes to backup volumes, migration cleanup and level-1-to-level-2 migration. (2) See *primary-processing-unit functions*.

level 0 volume (L0)

A volume that contains data sets directly accessible by the user. The volume may be either DFSMSHsm-managed or non-DFSMSHsm-managed.

level 1 volume (L1)

A volume owned by DFSMSHsm containing data sets that migrated from a level 0 volume.

level 2 volume (L2)

A volume under control of DFSMSHsm containing data sets that migrated from a level 0 volume, from a level 1 volume, or from a volume not managed by DFSMSHsm.

like device

Pertaining to (DASD) devices with identical geometry: that is, the same number of bytes per track, the same number of tracks per cylinder, and the same number of cylinders per actuator.

linear data set

In VSAM, a named linear string of data, stored in such a way that it can be retrieved or updated in 4096-byte units.

list of names (LON) data set

In ABARS, a data set that contains the fully-qualified names of data sets to be recovered from an aggregate backup version during ARECOVER processing.

logical end (LE)

Signifies that a concurrent copy environment has been successfully initialized. After logical end, the data is again available for unrestricted application access. Actual movement of data begins after logical end. See *physical end*.

low-activity data

Data that is infrequently accessed by users and is eligible to be moved or has been moved to DFSMSHsm-owned migration volumes.

low threshold of occupancy

The lower limit of space to be occupied on a volume managed by DFSMSHsm. See *high threshold of occupancy*.

managed volume

See *DFSMSHsm-managed volume* and *primary volume*.

management class

A list of data set migration, backup, and retention attributes that DFSMSHsm uses to manage storage at the data set level.

management work element (MWE)

A control block containing the necessary information to direct DFSMSHsm functions.

manual tape library (MTL)

A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which may be mounted on those drives. See *automated tape library (ATL)*.

master catalog

A key-sequenced data set or file with an index containing extensive data set and volume information that VSAM requires to locate data sets or files, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set or file, and to accumulate usage statistics for data sets or files.

MB

Megabyte; 1 048 576 bytes.

MCB

Backed up data set record.

MCBR

Backed up data set record for retained backup copies.

MCC

Backup version record.

MCD

MCDS data set record.

MCDS

See *migration control data set (MCDS)*.

MCR

Management control record.

MCVT

Management communication vector table.

MEDIA5

IBM Enterprise Tape Cartridge.

MEDIA6

IBM Enterprise WORM Tape Cartridge.

MEDIA7

IBM Enterprise Economy Tape Cartridge.

MEDIA8

IBM Enterprise Economy WORM Tape Cartridge.

MEDIA9

IBM Enterprise Extended Tape Cartridge.

MEDIA10

IBM Enterprise Extended WORM Tape Cartridge.

MEDIA11

IBM Enterprise Advanced Tape Cartridge.

MEDIA12

IBM Enterprise Advanced WORM Tape Cartridge.

MEDIA13

IBM Enterprise Advanced Economy Tape Cartridge.

MHCR

Multiple-processing-unit control record.

migration

In DFSMSHsm, the process of moving a cataloged data set from a DFSMSHsm-managed volume to a migration level 1 or migration level 2 volume, from a migration level 1 volume to a migration level 2 volume, or from a volume not managed by DFSMSHsm to a migration level 1 or migration level 2 volume.

migration cleanup

In DFSMSHsm, a process of automatic secondary space management that deletes unnecessary records or migration copies.

migration control data set (MCDS)

A VSAM, key-sequenced data set that contains statistics records, control records, user records, records for data sets that have migrated, and records for volumes under migration control of DFSMSHsm.

migration level 1 (ML1) volume

Level 1 volume used for migration from level 0 or primary volumes. See *level 0 volume*, *level 1 volume*, *level 2 volume*, *migration*, *migration volume*, and *primary volume*.

migration level 2 (ML2) volume

Level 2 volume used for migration from level 1 or directly from primary volumes. See *level 0 volume*, *level 1 volume*, *level 2 volume*, *migration*, *migration volume*, and *primary volume*.

migration volume

A volume, under control of DFSMSHsm, that contains migrated data sets.

minimal discrete profile

A profile with no access list or model profile. The minimal discrete profile is used when recovering a RACF-indicated data set whose original profile or backup profile no longer exists.

minimum migration age

In DFSMSHsm, the number of days a data set must remain unopened before DFSMSHsm can select it to migrate from a volume.

MIPS

Million instructions per second.

ML1

Migration level 1. Synonym for level 1 volume.

ML2

Migration level 2. Synonym for level 2 volume.

model entity

A model data set name that defines a discrete data set profile for RACF protection.

MSS

Mass Storage System.

MTL

See *manual tape library (MTL)*.

multicluster, key-range control data set

A control data set (MCDS or BCDS) that is represented by two or more key-sequenced data set (KSDS) clusters. Each cluster is a single key range. See *single-cluster, key range control data set*.

multiple-file format

In DFSMSHsm, a 3480 tape format, or the equivalent, that requires a unique standard label data set for each user data set written. When DFSMSHsm writes in multiple-file format, it writes one tape data set for every user data set to all 3480 migration and backup volumes.

multiple DFSMSHsm-host environment

Any environment in which two or more DFSMSHsm hosts share a common set of control data sets.

mutually exclusive parameters

A set of parameters of which only one can be used. If more than one parameter is specified, only the last parameter specified is used.

MWE

See *management work element (MWE)*.

non-DFSMSHsm-managed volume

A volume not defined to DFSMSHsm containing data sets that are directly accessible to users.

non-intrusive journal backup

A method of journal backup during which all DFSMSHsm activity is allowed to continue while DFSMSHsm creates a backup of the preexisting journal records. After the initial backup of the journal completes, all DFSMSHsm activity is quiesced and DFSMSHsm completes the backup of the journal. Non-intrusive journal backup can be used to reduce the impact of journal backup on DFSMSHsm availability and performance. See *quiesced journal backup* and [“Using non-intrusive journal backup” on page 496](#).

OCDS

Offline control data set.

O/C/EOV

Open/close/end-of-volume.

OCO

Object code only.

offline control data set (OCDS)

In DFSMSHsm, a VSAM, key-sequenced data set that contains information about tape backup volumes and tape migration level 2 volumes.

on-demand migration

A space management function that performs near immediate space management on eligible SMS-managed volumes that have exceeded the specified volume high threshold, instead of waiting for interval migration to run at the top of each hour. Data sets are migrated from the eligible volume until the low threshold is reached or there are no more eligible data sets. See [“Using on-demand migration” on page 27](#).

online

(1) Pertaining to the operation of a functional unit when under the direct control of a computer. (A) (ISO) (2) Pertaining to a user’s ability to interact with a computer. (A) (3) Pertaining to a user’s access to a computer via a terminal. (A) (4) Controlled by, or communicating with, a computer.

original tape volume

In DFSMSHsm, a 3480 or 3490 single-file tape volume, or the equivalent, used to store data during migration or backup processing, and from which a copy (called the alternate volume) is made for disaster recovery.

owned space

The storage space on a set of volumes to which DFSMSHsm allocates migrated data sets and backup versions, but to which user jobs should not allocate. Included in this set are migration level 1, migration level 2, and backup volumes.

owned volume

A volume on which DFSMSHsm writes dump, migration, or backup data sets.

parallel sysplex

A sysplex with one or more coupling facilities. See *base sysplex* and *sysplex*.

partitioned data set (PDS)

A data set in DASD that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE)

A DFP library structure that is an enhanced replacement for a partitioned data set.

path

In VSAM, a named logical entity providing access to the records of a base cluster either directly or through an alternate index.

PCDD

Pseudo common data set descriptor (CDD) record.

PDA

Problem Determination Aid.

PDF

Program Development Facility.

PDS

Partitioned data set.

PDSE

Partitioned data set extended.

PE

Physical end.

peer-to-peer remote copy (PPRC)

A hardware-based remote copy option that provides a synchronous volume copy across storage subsystems for disaster recovery, device migration, and workload migration.

performance scaling

Formatting a tape so that less than its full capacity is used. Formatting a tape to its optimal performance capacity decreases the amount of time required to access the data on the tape. A tape that is formatted to its optimal performance capacity can later be reused and formatted to its full capacity.

physical data set restore

In DFSMSHsm, the process of using a DFSMSdss function to restore one data set from a dump copy created by using the DFSMSHsm full volume dump function.

physical end (PE)

Signifies that the concurrent copy process has finished copying the data to the output device. See *logical end*.

physical sequential

See *sequential data set*.

pool of volumes

See *data set pool*, *general pool*, and *volume pool*.

PPRC

Peer-to-peer remote copy.

primary processing unit

In a multiple processing-unit-environment, the processing unit assigned to do level functions for backup and dump. Level functions for backup are: backup of CDS, move backup versions, and backup of migrated data sets. Level functions for dump are: delete expired dump copies, and delete excess dump VTOC copy data sets.

primary-processing-unit functions

(1) The level functions (backing up migrated data sets, deleting expired dump copies, and moving backup versions from level 1 volumes backup volumes) that must be performed by the primary processing unit. (2) See *level functions*.

primary space management

See *automatic primary space management*.

primary volume

A non-SMS volume managed by DFSMSHsm containing data sets that are directly accessible to the user.

problem determination aid trace

This data set is used to gather information about DFHSM processing.

promotion

The process of one host taking over level functions for another. See *demotion*.

PSCB

Protected step control block.

PSP

Preventive service planning.

PSW

Program status word.

PTF

Program temporary fix.

quiesced journal backup

A method of journal backup during which all DFSMSHsm activity is quiesced while DFSMSHsm complete backup of the journal. See *non-intrusive journal backup* and [“Using non-intrusive journal backup” on page 496](#).

quiesce time

A time of day after which an automatic function does not start processing any more volumes.

RACF

See *Resource Access Control Facility (RACF)*.

recall

The process of moving a migrated data set from a level 1 or level 2 volume to a DFSMSHsm-managed volume or to a volume not managed by DFSMSHsm.

reconnection

See *fast subsequent migration reconnect*.

record-level sharing (RLS)

See *VSAM record-level sharing (VSAM RLS)*.

recording format

For a tape volume, the format of the data on the tape; for example, 18, 36, 128, 256, 384 tracks, EFMT1, EFMT2, EEFMT2, EFMT3, EEFMT3, EFMT4, or EEFMT4.

recovery

In DFSMSHsm, the process of copying a backup version of a data set from a backup volume to a specified volume or to the volume from which the backup version was created.

recycle process

A DFSMSHsm process that, based on the percentage of valid data on a tape backup or migration level 2 volume, copies all valid data on the tape to a tape spill backup or migration level 2 volume.

reentrant

The attribute of a program or routine that allows the same copy of that program or routine to be used concurrently by two or more tasks.

Resource Access Control Facility (RACF)

An IBM licensed program that provides access control by identifying and by verifying users to the system. RACF authorizes access to protected resources, logs unauthorized access attempts, and logs accesses to protected data sets.

restart data set

A data set created by DFSMSHsm if aggregate recovery fails. It contains a list of all the data sets successfully restored during the aggregate recovery and allows the user to restart the aggregate recovery after the cause of the failure has been resolved.

restore

In DFSMSHsm, the process of invoking DFSMSdss to perform the program's recover function. In general, it is to return to an original value or image, for example, to restore data in main storage from auxiliary storage.

restricted command authority

In DFSMSHsm, allowing a user having at least read authority to each data set within an aggregate group to process ABACKUP or ARECOVER.

retained copies

Retained copies are the set of backup copies that have rolled off from the active copies and have not yet reached their retention periods. A nearly unlimited number of retained copies for each data set can be maintained.

retired version

In DFSMSHsm, a specially marked backup version that DFSMSHsm created before it deleted the non-SMS-managed original data set during data set retirement.

retirement

See *data set retirement*.

reverse restore

A FlashCopy option that reverses a FlashCopy relationship by copying over modified tracks from the original target volume to the original source volume. The background copy process must complete before you can perform reverse restore.

REXX

Restructured extended executor.

RLS

Record level sharing.

roll off

When a new backup, dump, or generation data set copy is created, the oldest copy becomes a candidate for deletion, to maintain a customer-specified limit.

RPL

Request parameter list. Part of VSAM control block.

RRDS

Relative record data set.

SAF

System authorization facility.

SAQ

Supplement for additional quantities.

SCP

System control programming.

SDSP

Small data set packing.

secondary address space

Synonym for DFSMSHsm secondary address space.

secondary space management

See *automatic secondary space management*.

selection data set

In aggregate backup and recovery processing, a sequential data set or a member of a partitioned data set used to define the data sets that compose the input to the aggregate backup function. It contains any include, exclude, accompany, or allocate lists.

sequential data set

A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape.

similar device

A (DASD) device with the same number of bytes per track and tracks per cylinder.

single-cluster control data set

A control data set (MCDS or BCDS) that is represented by a single VSAM key-sequenced data set (KSDS) cluster. This can be a key-range or non-key-range cluster.

single-file format

In DFSMSHsm, a 3480 or 3490 format, or the equivalent, consisting of one standard-label data set that spans up to 255 tape volumes.

single DFSMSHsm-host environment

Any environment in which a single DFSMSHsm host has exclusive use of a set of control data sets.

small data set packing (SDSP)

In DFSMSHsm, the process used to migrate data sets that contain equal to or less than a specified amount of actual data. The data sets are written as one or more records into a VSAM data set on a migration level 1 volume.

small-data-set-packing data set

In DFSMSHsm, a VSAM key-sequenced data set allocated on a migration level 1 volume and containing small data sets that have migrated.

SMF

System Management Facilities.

SMP

System Modification Program.

SMP/E

System Modification Program Extended.

SMS

Storage Management Subsystem.

SMS class

A list of attributes that SMS applies to data sets having similar allocation (data class), performance (storage class), or availability (management class) needs.

SMS-managed data set

A data set that has been assigned a storage class.

SnapShot

A function of the IBM RAMAC Virtual Array (RVA) that allows an instantaneous copy to be made of data sets using DFSMS software.

space manager

Synonym for storage administrator.

space management

In DFSMSHsm, the process of managing aged data sets on DFSMSHsm-managed and migration volumes. The three types of space management are: migration, deletion, and retirement.

spanning data sets

Data sets that span one or more tape volumes. The tape volumes that are spanned are part of a connected set. See *connected set*.

specific scratch pool

A group of empty tapes with unique serial numbers that are known to DFSMSHsm as a result of being defined to DFSMSHsm with the ADDVOL command.

spill backup volume

A volume owned by DFSMSHsm to which all but the latest backup version of a data set are moved when more space is needed on a DASD daily backup volume or all valid versions are moved when a tape backup volume is recycled.

spill process

A DFSMSHsm process that moves all but the latest backup version of a data set from a DASD daily backup volume to a spill backup volume.

SSF

Software Support Facility.

SSI

Subsystem interface.

SSM

Secondary space management.

SSSA

Subsystem option block extension for SMS.

storage administrator

In DFSMSHsm, the person who is authorized through the DFSMSHsm AUTH command to enter DFSMSHsm system programmer and storage administrator commands, who can affect the authority of other DFSMSHsm users, and who controls the manner in which DFSMSHsm manages DASD space.

storage class

A named list of data set storage service attributes that identifies performance and availability requirements. SMS uses these attributes to control data placement.

storage group

(1) A named list of DASD volumes used for allocation of new SMS-managed data sets, or a dummy storage group, or a VIO storage group. (2) A list of real DASD volumes, or a list of serial numbers of volumes that no longer reside on a system but that end users continue to refer to in their JCL.

storage hierarchy

An arrangement in which data may be stored in several types of storage devices that have different characteristics such as capacity and speed of access.

Storage Management Subsystem (SMS)

A z/OS DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

suballocated file

A VSAM file that occupies a portion of an already defined data space. The data space may contain other files. See *unique file*.

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.

SVC

Supervisor call instruction.

sysplex

A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads. This term is derived from *system complex*. See *base sysplex* and *parallel sysplex*.

sysplex timer

An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides.

system-managed storage

Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, and space to applications.

take back

Process performed by a demoted host to take back the level functions that have been taken over by a promoted host.

TCB

Task control block.

threshold of occupancy

A limit of occupied space on a volume managed by DFSMSHsm.

time sharing option (TSO)

An option on the operating system for a System/370 that provides interactive time sharing from remote terminals.

TIOT

Task input/output table.

TMP

Terminal monitoring program.

TOD

Time of day.

trace

(1) A record of the execution of a computer program that exhibits the sequence in which the instructions were executed. (2) To record a series of events as they occur.

TSN

Time sharing option.

TSO/E

Time sharing option/extended.

TTOC

Tape table of contents.

TVT

Tape volume table.

TS1120

IBM System Storage TS1120 (3592 Model E05, 3592-2 and 3592-2E).

TS1130

IBM System Storage TS1130 (3592 Model E06, 3592-3E).

TS1140

IBM System Storage TS1140 (3592 Model E07, 3592-4E).

UCB

Unit control block.

UID

The DFSMSHsm authorized-user identification in 1 to 7 characters.

undirected recall

In DFSMSHsm, moving a migrated data set from a level 1 or level 2 volume to a level 0 volume without specifying the target volume or unit where the volume can be allocated. Undirected recall can be automatic or by command.

unique file

A VSAM file that occupies a data space of its own. The data space is defined at the same time as the file and cannot contain any other file. See *suballocated file*.

unlike device

A DASD device with a different number of bytes per track and tracks per cylinder, or both.

user catalog

An optional catalog used in the same way as the master catalog and pointed to by the master catalog. Use of user catalogs lessens the contention for the master catalog and facilitates volume portability.

user exit

A programming service provided by an IBM software product that may be requested during the execution of an application program that transfers control back to the application program upon the later occurrence of a user-specified event.

virtual DASD

In DFSMSHsm, this refers to the 3850 Mass Storage System (MSS).

virtual storage access method (VSAM)

An access method for direct or sequential processing of fixed-length and variable-length records on disk devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry sequence), or by relative-record number.

virtual storage constraint relief (VSCR)

A function that increases the amount of storage available for the user's application program.

volume

(1) A certain portion of data, together with its data carrier, that can be handled conveniently as a unit. (ISO) (2) A data carrier that is mounted and demounted as a unit, for example, a reel of magnetic tape, a disk pack. (ISO) (3) That portion of a single unit of storage that is accessible to a single read/write mechanism, for example, a drum, a disk pack, or part of a disk storage module. (4) A storage medium that is mounted and demounted as a unit; for example, magnetic tape or diskette.

volume pool

In DFSMSHsm, a set of related primary volumes. When a data set is recalled, if the original volume that it was on is in a defined volume pool, the data set can be recalled to one of the volumes in the pool.

volume serial number

An identification number in a volume label that is assigned when a volume is prepared for use in the system.

volume table of contents (VTOC)

(1) A table on a direct access volume that describes each data set on the volume. (2) An area on a disk or diskette that describes the location, size, and other characteristics of each file and library on the disk or diskette.

VSA

VSAM data set allocation control block.

VSAM

Virtual storage access method.

VSAM record-level sharing (VSAM RLS)

An extension to VSAM that provides direct record-level sharing of VSAM data sets from multiple address spaces across multiple systems. Record-level sharing uses the z/OS coupling facility to provide cross-system locking, local buffer invalidation, and cross-system data caching.

VSAM RLS

See *VSAM record-level sharing (VSAM RLS)*.

VSAM sphere

(1) The base cluster of a VSAM data set and its associated alternate indexes. (2) A sphere containing the following eight components: base cluster, base data object, base index object, base path, alternate index, alternate index data object, alternate index index object, and alternate index path.

VSCR

Virtual storage constraint relief.

VSR

Volume statistics record.

VTOC

Volume table of contents.

VTOC copy data set

A data set that contains a copy of the VTOC entry for each data set that DFSMSHsm backs up or dumps.

VTs

Virtual tape server.

VVDS

VSAM volume data set.

WWFSR

ABACKUP or ARECOVER backup and recovery record.

XCF

Cross-system coupling facility.

XRC

Extended remote copy.

z/Architecture

An IBM architecture for mainframe computers and peripherals. The IBM Z® family of servers uses the z/Architecture. It is the successor to the S/390®, zSeries, and 9672 family of servers. See *Enterprise Systems Architecture/390 (ESA/390)*.

Index

Special Characters

GDG ELEMENTS ON PRIMARY attribute [40](#)

A

ABA (aggregate backup assist) [357](#)

ABACKUP

command [324](#)

examples [349](#)

HOLD command, parameter of [597](#)

invoking [347](#)

issuing with the EXECUTE parameter [350](#)

optional parameters [347](#)

parameters

EXECUTE [347](#)

FILTEROUTPUTDATASET [348](#)

MOVE [348](#)

OPTIMIZE [344](#), [347](#), [348](#)

UNIT [347](#)

VERIFY [347](#)

RELEASE command, parameter of [600](#)

required parameters [347](#)

syntax [347](#)

using with VERIFY parameter [349](#)

ABACKUP command

examples [695](#)

optional parameters

FILTEROUTPUTDATASET [693](#)

LIST [693](#)

MOVE [695](#)

NOSTACK [695](#)

OPTIMIZE [694](#)

PROCESSIONLY [694](#)

SKIP [694](#)

STACK [695](#)

UNIT [695](#)

overview [691](#)

required parameters

EXECUTE [692](#)

VERIFY [692](#)

syntax [692](#)

ABARs

aggregate

copying to another device type [368](#)

ABARS (aggregate backup and recovery support)

account field for aggregate groups [353](#), [413](#)

activity log, backing up and recovering [401](#)

activity record

creating [381](#)

expiring [354](#)

backup error exit, ARCBEXT [342](#)

CPU processing time, tracking [353](#), [413](#)

defining aggregate group and management class to

DFSMS [321](#)

DFSMSdss [321](#)

expiration date exit, ARCEEXT [342](#)

ABARS (aggregate backup and recovery support) (*continued*)
functions

held, querying [409](#)

held, releasing [409](#)

holding [408](#)

general description [12](#)

ICF catalog and catalog alias processing [356](#), [410](#)

instruction data set, recovering [400](#)

interaction with other products [319](#)

ML2 data set exit, ARCM2EXT [342](#)

overview [319](#)

RACF [321](#)

ABARSACTLOGMSGLVL parameter, SETSYS command [344](#)

ABARSBUFFERS parameter, SETSYS command [344](#)

ABARSDELETEACTIVITY parameter, SETSYS command [353](#)

ABARSOPTIMIZE parameter, SETSYS command [344](#)

ABARSPROCNAME parameter, SETSYS command [343](#), [384](#)

ABARSTAPES parameter, SETSYS command [344](#), [385](#)

abbreviations

A3590UNITNAME (ALTERNATE3590UNITNAME) [1227](#)

ABE (AUTOBACKUPEND) [810](#)

AFM (AVAILABLEFORMOUNT) [810](#)

AINC (APPLYINCREMENTAL) [998](#)

AMIG (AUTOMIGRATION) [1030](#)

AREC (AUTORECALL) [698](#)

ARUSE (AUTOREUSE) [810](#)

AUNITNAME (ALTERNATEUNITNAME) [1227](#)

AVOL (ALTERNATEVOLUME) [920](#)

AVOLS (ALTERNATEVOLUMES) [1227](#), [1235](#)

BCDS (BACKUPCONTROLDATASET) [920](#), [1243](#)

CDS (CONTROLDATASETS) [982](#)

CP (COPYPOOL) [865](#), [873](#)

CQUEUE (COMMONQUEUE) [896](#), [982](#), [1030](#)

DAVOLUMES (DISASTERALTERNATEVOLUMES) [1235](#)

DBA (DATABASEAUTHORITY) [755](#)

DBU (DELETEIFBACKEDUP) [698](#)

DCLASS (DUMPCCLASS) [698](#), [810](#), [865](#), [867](#), [873](#)

DCONTENTS (DUMPCONTENTS) [920](#)

DCYCLE (DUMPCYCLE) [810](#)

DSNAME (DATASETNAME) [1053](#)

DSRESTORE (DATASETRESTORE) [810](#)

DVOL (DUMPVOLUME) [920](#)

EXVOLS (EXCESSIVEVOLUMES) [920](#)

FODS (FILTEROUTPUTDATASET) [692](#)

FRDUMP (FASTREPLICATIONDUMP) [810](#)

FREND (FASTREPLICATIONEND) [865](#), [866](#)

FRML1 (FROMMIGRATIONLEVEL1) [1044](#)

FRRECOV (FASTREPLICATIONRECOVER) [810](#)

L0 (LEVEL0) [692](#)

MCDS (MIGRATIONCONTROLDATASET) [1243](#)

MIGCC (MIGRATIONCLEANUPCYCLE) [810](#)

ML1 (MIGRATIONLEVEL1) [692](#), [698](#)

ML2 (MIGRATIONLEVEL2) [692](#), [810](#), [878](#), [920](#), [965](#),
[982](#), [1227](#)

MVOL (MIGRATIONVOLUME) [920](#)

NOAMIG (NOAUTOMIGRATION) [698](#)

NOARUSE (NOAUTOREUSE) [810](#)

abbreviations (*continued*)

NODSI (NODATASETINFORMATION) [920](#)
NODSRESTORE (NODATASETRESTORE) [810](#)
OCDS (OFFLINECONTROLDATASET) [1243](#)
OVOLS (ORIGINALVOLUMES) [1227](#), [1235](#)
PONLY (PROCESSIONLY) [692](#)
PSMCYCLE (PRIMARYSPMGMTCYCLE) [810](#)
RENTGT (RENAMETARGET) [724](#)
RETPD (RETENTIONPERIOD) [810](#)
RTAKE (RECALLTAKEAWAY) [920](#)
SDSP (SMALLDATASETPACKING) [748](#)
SG [780](#)
SGROUP [780](#)
SRCVOL [998](#)
SSMCYCLE (SECONDARYSPMGMTCYCLE) [810](#)
ST (SWITCHTAPES) [810](#)
STRGRP [780](#)
VOLPOOL (VOLUMEPOOL) [810](#), [982](#)

ABEND (abnormal end)

restarting DFSMSHsm after [614](#)

ABR record, how to create [733](#)

Access Method Services

considerations [152](#)

accessibility

contact IBM [1413](#)

accessing a data set when profile cannot be recovered [462](#)

ACCOMPANY

data set [323](#), [332](#)

keyword [332](#)

ACCOMPANYEXCLUDE

data set [323](#)

keyword [333](#)

accounting information, ABARS CPU time [353](#), [413](#)

active backup copies [184](#)

activity log

aggregate backup output [325](#), [352](#)

aggregate recovery output [411](#)

SETSYS command

ACTLOGTYPE parameter [343](#)

ACTIVITY parameter, ARECOVER command [393](#)

additional considerations for

availability management [445](#)

backup [445](#)

dump [455](#)

recall [86](#)

ADDVOL

adding aggregate recovery volumes to DFSMSHsm [383](#)

defining dump volumes to DFSMSHsm [177](#)

level 0 volumes for automatic backup, non-SMS [304](#)

level 0 volumes for space management, non-SMS [102](#)

level 0 volumes to be dumped, specifying [305](#)

migration level 2 volumes [55](#)

optional parameters

AUTOBACKUP [304](#)

AUTODUMP [305](#)

AUTOMIGRATION [103](#)

AUTORECALL [103](#)

BACKUPDEVICECATEGORY [103](#)

DAILY [192](#)

DELETEBYAGE [103](#)

DELETEIFBACKEDUP [103](#)

DRAIN [535](#)

DUMPCCLASS [177](#)

MIGRATE [103](#)

ADDVOL (*continued*)

optional parameters (*continued*)

THRESHOLD [103](#)

PRIMARY, pool configuration in JES3 [106](#)

required parameters

BACKUP [192](#)

DUMP [177](#)

PRIMARY [106](#), [304](#)

ADDVOL command

optional parameters

AUTOBACKUP [701](#)

AUTODUMP [702](#)

AUTOMIGRATION [702](#)

AUTORECALL [703](#)

BACKUPDEVICECATEGORY [703](#)

DAILY [704](#)

DELETEBYAGE [704](#)

DELETEIFBACKEDUP [704](#)

DENSITY [705](#)

DRAIN [706](#)

DUMPCCLASS [706](#)

MIGRATE [704](#)

MIGRATIONLEVEL1 [707](#)

MIGRATIONLEVEL2 [707](#)

NOAUTOBACKUP [701](#)

NOAUTODUMP [702](#)

NOAUTOMIGRATION [702](#)

NOAUTORECALL [703](#)

NODRAIN [706](#)

NOOVERFLOW [707](#)

NOSMALLDATASETPACKING [708](#)

OVERFLOW [707](#)

SMALLDATASETPACKING [708](#)

SPIII [704](#)

THRESHOLD [708](#)

TRACKMANAGEDTHRESHOLD [710](#)

overview [697](#)

required parameters

BACKUP [700](#)

DUMP [700](#)

MIGRATION [700](#)

PRIMARY [700](#)

UNIT [700](#)

volser [700](#)

syntax [698](#)

ADDVOL ommand

examples [711](#)

ADMIN OR USER COMMAND BACKUP attribute [185](#)

age of records for migration cleanup, specifying [54](#)

aggregate backup

aggregate group, example [325](#)

assist [357](#)

checklist [327](#)

comprehensive ABACKUP command authority [341](#)

creating selection data sets [329](#)

data file, description [324](#)

data sets dynamically allocated during [345](#)

defining SETSYS parameters [343](#)

determining unit type for control and data files [345](#)

ensuring that the volumes are online [343](#)

identifying applications to be backed up [322](#)

input for recovery [325](#)

installation-wide exits [342](#)

instruction/activity log file, description [325](#)

- aggregate backup (*continued*)
 - minimizing data set contention [345](#)
 - multiple ML2 data sets with single tape mount [351](#)
 - of data sets [347](#)
 - output [351](#)
 - overview [322](#)
 - performing [346](#)
 - performing actual backup of data sets [350](#)
 - preparing for [326](#)
 - RACF checking [340](#)
 - restricted ABACKUP command authority [341](#)
 - selection data set [357](#)
 - sequential data striping [431](#)
 - SMS requirements [328](#)
 - specifying applications to be backed up [322](#)
 - tape files created [351](#)
 - tape volumes, RACF [340](#)
 - using NetView file transfer program [427](#)
 - verification [347](#)
- aggregate backup and recovery information [1341](#)
- aggregate backup and recovery record
 - creating [381](#)
 - description of [381](#)
 - expiring [354](#)
 - recording CPU time and accounting codes [413](#)
- aggregate group
 - AGGREGATE GROUP NAME parameter [347](#)
 - Aggregate Group Name, ISMF panel parameter [336](#)
 - backup, using ISMF panels [413](#)
 - defining a group [336](#)
 - example [337](#)
 - overview [322](#)
 - panel parameter [336](#)
- AGGREGATE parameter, ARECOVER command [393](#)
- aggregate recovery
 - ABARSPROcname parameter, SETSYS command [384](#)
 - ARECOVERUNITNAME parameter, SETSYS command [385](#)
 - checklist [375](#)
 - comprehensive ARECOVER command authority [377](#)
 - data set name conflict resolution [402](#)
 - data sets dynamically allocated during [410](#)
 - defining SETSYS parameters [384](#)
 - determining site constraints [379](#)
 - example
 - instruction data set [380](#)
 - recovered data sets [413](#)
 - HOLD command, parameter of [597](#)
 - installation-wide exits [387](#), [398](#)
 - MAXABARSADDRESSSPACE parameter, SETSYS command [384](#)
 - model entity [387](#)
 - model entity (MENTITY) parameter, ARECOVER command [394](#)
 - output [373](#), [411](#)
 - overview [372](#)
 - performing
 - recovery of data sets [399](#)
 - verification [398](#)
 - preparing for [374](#)
 - RACF checking during [376](#)
 - recovered data sets [373](#)
 - recovery
 - of data sets [392](#)
- aggregate recovery (*continued*)
 - recovery (*continued*)
 - site, example [374](#)
 - RELEASE command, parameter of [600](#)
 - restart data set [399](#)
 - restricted ARECOVER command authority [377](#)
 - restrictions to adding volumes to DFSMSHsm [383](#)
 - retrieving the instruction data set [379](#)
 - secondary address space [384](#)
 - sequential data striping [432](#)
 - SETSYS command examples [386](#)
 - SMS requirements [378](#)
- Aggregate Stacking Procedure [360](#)
- ALL
 - BDELETE command parameter [798](#)
- ALL parameter
 - DEFINE command [383](#)
 - HOLD command [596](#)
 - RECYCLE command [539](#)
 - RELEASE command [600](#)
 - TAPECOPY command [438](#)
 - UPDATE command [516](#)
- ALLOCATE
 - data set [323](#), [333](#)
 - keyword [333](#)
 - list [323](#)
 - processing VSAM data sets in list [389](#)
- ALLOCATEEXCLUDE
 - data set [323](#)
 - keyword [334](#)
- allocation
 - ALLOCATE keyword [333](#)
 - dynamic data set, during aggregate backup processing [345](#)
 - dynamic data set, during aggregate recovery processing [410](#)
 - recalled data sets to volumes, non-SMS-managed [120](#)
- ALLOCATION/MIGRATION THRESHOLD attribute [32](#)
- ALLOWPPRCP option
 - Fast Replication Backup [267](#)
 - Fast Replication Recovery [279](#)
- ALTERDS [308](#)
- ALTERDS command
 - examples [717](#)
 - for DFSMSHsm [715](#)
 - optional parameters
 - FREQUENCY [716](#)
 - SYSFREQUENCY [716](#)
 - SYSVERSIONS [717](#)
 - VERSIONS [717](#)
 - required parameters
 - name [716](#)
- alternate tape volume, replacing an original with [440](#)
- alternate tapes, sending off site [441](#)
- ALTERNATEVOLUMES parameter, TAPECOPY command [439](#)
- ALTERPRI [606](#)
- ALTERPRI command
 - examples [721](#)
 - for DFSMSHsm [719](#)
- always-call
 - environment, definition [618](#)
 - RACF indicator with [619](#)
 - support [618](#)
 - support for migrated data sets and backup copies [623](#)

- ANYSTORAGEVOLUME subparameter, RECALL parameter, SETSYS command
 - defining volumes eligible for recall selection [101](#)
 - general pool [106](#)
 - pool configuration [106](#)
 - pool configuration in JES3 [106](#)
- application programs, requesting DFSMSHsm service [558](#)
- ARC0500I [1339](#)
- ARC0525I [1339](#)
- ARC0805I [1339](#)
- ARC0806I [1339](#)
- ARC0807I [1339](#)
- ARC0808I [1339](#)
- ARCADEXT, data set deletion exit [115](#)
- ARCBDEXT, data set backup exit [204](#)
- ARCBEXT, ABARS backup error exit [342](#)
- ARCBJRNL
 - journal copy utility [508](#)
 - return codes [512](#)
- ARCCATGP, RACF logon group [86](#)
- ARCCBEXT, CDS backup exit [499](#)
- ARCCREXT
 - in ABARS ARECOVER support [410](#)
 - in aggregate recovery [373](#), [387](#)
 - in RACF facility class authorization checking [407](#)
 - installation exit [406](#)
- ARCEDEXT, ABARS expiration date exit [342](#)
- ARCIMPRT
 - import utility [508](#)
 - return codes [512](#)
- ARCINBAK program, inline backup [447](#)
- ARCM2EXT, ABARS ML2 data set exit [342](#)
- ARCMDEXT, data set migration exit [96](#), [118](#)
- ARCMMEXT, second-level migration data set exit [63](#)
- ARCPEDIT program
 - printing DFSMSHsm log information [607](#)
- ARCPRLOG program
 - printing DFSMSHsm log information [607](#)
 - sample list [612](#)
- ARCSAEXT, space management and backup exit [66](#)
- ARCTDEXT, tape data set exit [63](#)
- ARCTVEXT tape volume exit [437](#), [441](#)
- ARECOVER
 - and the duplex tape option [437](#)
 - catalog considerations [390](#)
 - conflict resolution, order of processing [406](#)
 - examples [397](#)
 - examples with
 - EXECUTE parameter [399](#)
 - PREPARE parameter [398](#)
 - VERIFY parameter [398](#)
 - invoking command [392](#)
 - optional parameters
 - BYPASS [394](#)
 - DATASETCONFLICT [394](#)
 - DATE [394](#)
 - MENTITY [387](#), [394](#)
 - MIGRATEDDATA [395](#)
 - ML1 [395](#)
 - ML2 [395](#)
 - NOBACKUPMIGRATED [395](#)
 - NOSTACK [348](#), [394](#)
 - PERCENTUTILIZED [395](#)
 - RECOVERNEWNAMEALL [395](#)

- ARECOVER (*continued*)
 - optional parameters (*continued*)
 - RECOVERNEWNAMELEVEL [395](#)
 - RENAMESOURCE [394](#)
 - RENAMETARGET [394](#)
 - REPLACE [394](#)
 - SOURCELEVEL [395](#)
 - STACK [348](#), [394](#)
 - TARGETUNIT [385](#), [396](#)
 - TGTGDS [386](#), [392](#), [396](#)
 - UNIT [394](#)
 - VERSION [394](#)
 - performing verification [392](#)
 - required parameters
 - AGGREGATE [393](#)
 - DATASETNAME [393](#)
 - EXECUTE [392](#)
 - PREPARE [398](#)
 - VERIFY [392](#)
 - VOLUMES [393](#)
 - XMIT [393](#)
 - syntax [392](#)
- ARECOVER command
 - creating an ABR record [733](#)
 - examples [733](#)
 - optional parameters
 - ACTIVITY [727](#)
 - DATASETCONFLICT [727](#)
 - DATE [728](#)
 - INSTRUCTION [728](#)
 - MENTITY [728](#)
 - MIGRATEDDATA [728](#)
 - NOBACKUPMIGRATED [729](#), [730](#)
 - NOSTACK [731](#)
 - PERCENTUTILIZED [730](#)
 - RECOVERNEWNAMEALL [730](#)
 - RECOVERNEWNAMELEVEL [730](#)
 - STACK [731](#)
 - TARGETUNIT [731](#)
 - TGTGDS [732](#)
 - UNIT [732](#)
 - VERSION [728](#)
 - VOLCOUNT [733](#)
 - overview [723](#)
 - required parameters
 - AGGREGATE [725](#)
 - DATASETNAME [725](#)
 - EXECUTE [726](#)
 - PREPARE [726](#)
 - VERIFY [726](#)
 - VOLUMES [725](#)
 - XMIT [725](#)
 - syntax [724](#)
- ARECOVERPERCENTUTILIZED parameter, SETSYS command [385](#)
- ARECOVERTGTGDS parameter, SETSYS command [386](#)
- ARECOVERUNITNAME parameter, SETSYS command [385](#)
- ARPOOL parameter, DEFINE command [382](#)
- assistive technologies [1413](#)
- assumptions [20](#)
- attention key
 - TSO [686](#)
- attributes
 - # GDG ELEMENTS ON PRIMARY [40](#)

attributes (*continued*)

- ADMIN OR USER COMMAND BACKUP [185](#)
- ALLOCATION/MIGRATION THRESHOLD [32](#)
- AUTO BACKUP
 - management class [186](#)
 - storage group [178](#)
- AUTO DUMP [166](#)
- AUTO MIGRATE [32](#)
- AUTOBACKUP [103](#)
- AUTODUMP [103](#)
- AUTOMIGRATION [103](#)
- AUTORECALL [103](#)
- BACKUP FREQUENCY [180](#)
- BACKUP SYSTEM/SYSPLEX NAME [178](#)
- BACKUPDEVICECATEGORY [103](#)
- COMMAND OR AUTO MIGRATE [37](#)
- DUMP CLASS [166](#)
- DUMP SYSTEM/SYSPLEX NAME [166](#)
- EXPIRE AFTER DATE/DAYS [36](#)
- EXPIRE AFTER DAYS NON-USAGE [36](#)
- GDG management, specifying [40](#)
- GUARANTEED BACKUP FREQUENCY [180](#)
- LEVEL 1 DAYS NON-USAGE [37](#)
- LEVEL 2 DAYS NON-USAGE [38](#)
- MIGRATE SYSTEM/SYSPLEX NAME [35](#)
- migration, specifying [37](#)
- NUMBER OF BACKUP VERSIONS (DATA SET DELETED) [181](#)
- NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) [181](#)
- of primary volumes [103](#)
- PARTIAL RELEASE [36](#)
- PRIMARY DAYS NON-USAGE [37](#)
- recall [104](#)
- RETAIN DAYS EXTRA BACKUP VERSION [181](#)
- RETAIN DAYS ONLY BACKUP VERSION [181](#)
- ROLLED-OFF GDS ACTION [40](#)

AUDIT

- ensuring that CDSs are accurate [519](#)
- parameters
 - COMMONQUEUE [134](#)
 - DATASETCONTROLS [517](#)
 - HOLD [597](#)
 - NOFIX [517](#)
 - RELEASE [600](#)
 - REPORT [517](#)
 - SYSOUT [517](#)

AUDIT command

- for DFSMSHsm [737](#), [1255](#), [1256](#), [1280](#)
- output [1282](#), [1284](#), [1286](#), [1288](#), [1290–1292](#)
- parameters
 - ABARSCONTROLS [743](#)
 - ALL [743](#)
 - BACKUPCONTROLDATASET [743](#)
 - BACKUPTYPE [744](#)
 - BACKUPVOLUMES [744](#)
 - COMMONQUEUE [745](#)
 - COPYPOOLCONTROLS [746](#)
 - DATASETCONTROLS BACKUP [746](#)
 - DATASETCONTROLS MIGRATION [746](#)
 - DATASET NAMES [746](#), [748](#)
 - DIRECTORYCONTROLS VOLUMES [747](#)
 - FILECONTROLS BACKUP [747](#)
 - FIX [750](#)

AUDIT command (*continued*)

parameters (*continued*)

- LEVELS [746](#), [748](#)
- MASTERCATALOG [748](#)
- MEDIACONTROLS SMALLDATASETPACKING [748](#)
- MEDIACONTROLS VOLUMES [748](#)
- MIGRATIONCONTROLDATASET [743](#)
- NOFIX [750](#)
- OFFLINECONTROLDATASET [743](#)
- OUTDATASET [751](#)
- REPORT [752](#)
- RESUME [746](#), [748](#)
- SERIALIZATION CONTINUOUS [752](#)
- SERIALIZATION DYNAMIC [752](#)
- SYSOUT [751](#)
- TERMINAL [751](#)
- USERCATALOG [748](#)
- VOLUMECONTROLS BACKUP [749](#)
- VOLUMECONTROLS MIGRATION [749](#)
- VOLUMECONTROLS RECOVERABLE [749](#)
- VOLUMES [744](#)

AUTH command

examples [757](#)

for DFSMSHsm [755](#)

required parameters

- userid DATABASEAUTHORITY [756](#)
- userid REVOKE [756](#)

authority

- comprehensive ABACKUP command [341](#)
- comprehensive ARECOVER command [377](#)
- needed to perform DFSMSHsm RACF functions [620](#)
- restricted ABACKUP command [341](#)
- restricted ARECOVER command [377](#)

authorization

checking

- during HBDELETE processing [240](#)
- during profile recovery [462](#)
- recovery of RACF-indicated data sets [460](#)

AUTO BACKUP attribute

- management class [186](#)
- storage group [178](#)

AUTO DUMP attribute [166](#)

AUTO MIGRATE attribute [32](#)

AUTO subparameter

- BACKUP parameter
 - HOLD command [597](#)
 - RELEASE command [600](#)
- DUMP parameter
 - HOLD command [597](#)
 - RELEASE command [601](#)
- MIGRATION parameter
 - HOLD command [599](#)
 - RELEASE command [602](#)

AUTOBACKUP subparameter, PRIMARY parameter, ADDVOL command [304](#)

AUTOBACKUPSTART parameter, SETSYS command [188](#)

AUTODUMP subparameter, PRIMARY parameter, ADDVOL command [305](#)

automatic

- availability management of non-SMS-managed volumes [306](#)
- availability management operation [193](#)
- backup
 - example [220](#)

automatic (*continued*)

backup (*continued*)

preventing [449](#)

specifying when to start [188](#)

specifying whether to perform [186](#)

class selection, function of [11](#)

dump processing

specifying when to start [168](#)

full-volume dump, example system [165](#)

functions, running concurrently [615](#)

hold, release [603](#)

incremental backup, example system [165](#)

primary space management, example system [23](#)

recall

data sets migrated for extent reduction [85](#)

during data set allocation [85](#)

during non-VSAM data set opening, non-SMS-managed [121](#)

during VSAM data set opening, non-SMS-managed [121](#)

for command data set scratching, non-SMS-managed [121](#)

for command deletion if VSAM AIX clusters [85](#)

for data set renaming [85](#)

SMS-managed [83](#)

with DFSMSdss [85](#)

restart of primary space management [162](#)

restart of secondary space management [162](#)

reuse of dump tapes, controlling [171](#)

SMS-managed volumes [165](#)

tasks for availability management [166](#)

automatic dynamic volume expansion

concept [32](#)

automatic recall

concept [32](#)

automatic space management

candidate data sets for backup, determining, non-SMS-managed volume [112](#)

candidate data sets for backup, determining, SMS-managed storage [68](#)

candidate data sets for extent reduction, determining, non-SMS-managed volume [112](#)

candidate data sets for extent reduction, determining, SMS-managed storage [69](#)

candidate data sets for migration, determining, non-SMS-managed volume [112](#)

candidate data sets for migration, determining, SMS-managed storage [67](#)

candidates for both migration and extent reduction, migrate, SMS-managed storage [71](#)

candidates for extent reduction, migrate, SMS-managed storage [71](#)

compaction during [75](#)

defining a new start window [164](#)

deleting expired data sets, SMS-managed storage [66](#)

deleting temporary data sets, SMS-managed storage [66](#)

determining candidate data sets for backup, non-SMS-managed volume [112](#)

determining candidate data sets for backup, SMS-managed storage [68](#)

determining candidate data sets for extent reduction, non-SMS-managed volume [112](#)

determining candidate data sets for extent reduction, SMS-managed storage [69](#)

automatic space management (*continued*)

determining candidate data sets for migration, non-SMS-managed volume [112](#)

determining candidate data sets for migration, SMS-managed storage [67](#)

DFSMSHsm-managed volumes, SMS-managed storage [65](#)

example management classes

GDS management class [41](#)

large management class [42](#)

no-migration management class [42](#)

standard management class [41](#)

functions [7](#)

general description [56](#)

migrate candidates for both migration and extent reduction, SMS-managed storage [71](#)

migrate candidates for extent reduction, SMS-managed storage [71](#)

migrate data sets, SMS-managed storage [72](#)

operation [56](#)

out of space condition [77](#)

Phase 1

processing example, SMS-managed storage [69](#)

SMS-managed storage [65](#)

Phase 2

SMS-managed storage [70](#)

priority order for migration [72](#)

processing for non-SMS-managed storage [109](#)

processing for SMS-managed storage [56](#)

purpose [7](#)

releasing unused space, SMS-managed storage [69](#)

sort data sets in priority order for migration, SMS-managed storage [72](#)

specifying the number of secondary space management tasks [50](#)

specifying when to start primary space management [46](#)

specifying when to start secondary space management [46](#)

tasks [32](#)

VSAM data sets [75](#)

zEDC compression during [75](#)

AUTOMIGRATION

parameter, HOLD command, preventing processing [149](#)

subparameter, ADDVOL command [103](#)

AUTORECALL subparameter, ADDVOL command [103](#)

AUTOREUSE subparameter, DUMPCLASS parameter, DEFINE command [171](#)

auxiliary host [593](#)

availability management

automatic operation [193](#)

automatic, non-SMS-managed volumes [306](#)

automatic, SMS-managed volumes [165](#)

command [221](#)

command, example system [165](#)

command, for non-SMS-managed storage [308](#)

considerations

extended and compressed VSAM data sets [464](#)

extended SAM data sets [464](#)

sequential data striping [464](#)

defining a new start window [487](#)

dump tasks [166](#)

functions [7](#)

non-SMS-managed storage [303](#)

other considerations [445](#)

availability management (*continued*)

- procedures [485](#)
- purpose [7](#)
- security considerations [460](#)
- SMS-managed storage [165](#)
- tasks for automatic [166](#)

B

BACKDS command

- deleting expired backup versions of UNIX files [770](#)
- examples [771](#)
- for DFSMSHsm [759](#), [775](#)
- optional parameters
 - CC [762](#)
 - CHANGEDONLY [764](#)
 - DELETE [764](#)
 - EXCLUDE [764](#), [766](#), [1004](#)
 - NEWNAME [767](#)
 - RECURSE [769](#)
 - TARGET [770](#)
 - UNIT [770](#)
 - VOLUME [770](#)
- required parameters
 - dsname [761](#)
 - filepath [761](#)
 - UNIT parameter [309](#)
 - VOLUME parameter [309](#)

BACKFILE command

- for DFSMSHsm [773](#), [777](#)

backing up

- and recovering the ABARS activity log [401](#)
- applications [324](#)

backup

- additional considerations [445](#)
- all changed data sets on a volume, non-SMS-managed [309](#)
- all data sets on a volume, SMS-managed [230](#)
- all eligible changed data sets
 - in one or more storage groups [232](#)
 - on a volume [232](#)
- all eligible changed UNIX files
 - in a directory [232](#)
- all primary volumes, non-SMS-managed [309](#)
- automatic
 - example [220](#)
 - example system [165](#)
 - specifying storage for [178](#)
 - specifying whether to perform [186](#)
 - when to start, specifying time [188](#)
- cleanup process [219](#)
- conditions for a data set, changing [308](#)
- control data set (CDS)
 - examples [519](#)
 - how to [493](#)
 - manual [505](#)
 - processing [199](#)
 - protecting tapes [499](#)
 - recovery and multiple backup data sets, using UPDATEC [513](#)
 - specifying DASD for [498](#)
 - specifying device type [498](#)
 - specifying how many backup versions to keep [497](#)
 - specifying tape for [498](#)

backup (*continued*)

control data set (CDS) (*continued*)

- when accessing in RLS mode [200](#), [494](#), [498](#), [504](#)

copies

- always-call support [623](#)
- of DFSMSHsm-owned tape volumes [435](#)
- of SMS-managed data sets at recovery site [433](#)

current, definition [114](#)

cycle

- defining when to start [192](#)
- specifying [192](#)

daily volume selection

- DASD [209](#)
- DASD and tape [212](#)
- initial DASD [209](#)
- initial DASD or tape [213](#)
- subsequent DASD [211](#)
- subsequent DASD or tape [215](#)

daily, selecting [207](#)

DASD spill volume [218](#)

data sets

- all changed, in one or more storage groups [230](#)
- all changed, on a volume [232](#)
- by command, specifying who can [185](#)

data sets for CDSs

- changing names [503](#)
- specifying names [497](#)

determining candidate data sets, non-SMS-managed volumes [112](#)

DFSMSHsm-managed volumes from storage groups [201](#)

disaster, facilities [436](#)

eligible data sets [203](#)

expired versions of non-SMS-managed data sets, deleting [310](#)

frequency

- non-SMS-managed, specifying [303](#)
- relationship to maximum number of backup versions [181](#)
- SMS-managed, guaranteed backup [180](#)
- SMS-managed, specifying [180](#)

from DASD, RACF considerations [624](#)

ICF catalog, procedure [487](#)

individual data set [222](#)

individual data set, non-SMS-managed [309](#)

initial DASD daily volume selection [209](#)

initial selection, DASD or tape daily [213](#)

inline backup

- about [447](#)
- invoking [447](#)
- non-SMS-managed [309](#)
- SMS-managed [230](#)

interrupting and restarting [485](#)

management classes, example [186](#)

migrated data sets [200](#)

migrated data sets, specifying first qualifier [53](#)

name of

- cartridge-type single-file data sets [206](#)
- dump copy data sets [197](#)
- versions of data sets [204](#)
- VTOC copy data sets [202](#)

obtaining information [560](#)

overview

- backup (*continued*)
 - overview (*continued*)
 - non-SMS-managed [15](#)
 - SMS-managed [11](#)
 - preventing [449](#)
 - procedures [485](#)
 - processing example [199](#)
 - profile
 - authorization checking during recovery [460](#)
 - deleting [207](#)
 - parameters used in creating [207](#)
 - RACF discrete, for cataloged data sets [206](#)
 - transferring from one backup version to another [240](#)
 - RACF-indicated data sets [625](#)
 - recovering from a damaged migration level 1 volume [554](#)
 - redefining the cycle [446](#)
 - relationship of maximum number of versions to frequency [181](#)
 - releasing [220](#)
 - SDSP [556](#)
 - selecting volumes [216](#)
 - specified data sets, delete [239](#)
 - specifying
 - a storage group priority for autobackup processing [179](#)
 - first qualifier for names of versions [189](#)
 - how groups of data sets are to be managed [178](#)
 - how long to keep versions, SMS-managed [181](#)
 - kind of device to use [189](#)
 - maximum number of tasks in each processor [190](#)
 - maximum number of versions to keep, SMS-managed [181](#)
 - number of versions to keep, non-SMS-managed [303](#)
 - only changed data sets [190](#)
 - volumes [304](#)
 - whether to perform [189](#)
 - spill [215](#)
 - spill volume selection
 - initial [217](#)
 - subsequent [218](#)
 - storage groups with automatic backup attribute [201](#)
 - subsequent selection, DASD or tape daily [215](#)
 - switching backup tapes [227](#)
 - tape daily volume selection
 - initial [211](#)
 - subsequent [212](#)
 - tape spill volume [218](#)
 - tasks [177](#)
 - uncataloged data sets [307](#)
 - UNIX files
 - within a directory [230](#), [231](#)
 - versions
 - moving [200](#)
 - of specific data sets, non-SMS-managed, deleting [311](#)
 - volume
 - aggregate backup, RACF protection [340](#)
 - contention [445](#)
 - defining to DFSMSHsm [192](#)
 - deleting expired backup versions from [233](#)

- backup (*continued*)
 - volume (*continued*)
 - initial selection, spill [217](#)
 - not managed by DFSMSHsm [309](#)
 - recovery combined with full-volume restore [295](#)
 - releasing a backup [220](#)
 - restore directly with DFSMSdss [458](#)
 - selection [216](#)
 - spill [294](#)
 - subsequent selection, spill [218](#)
 - wait for spill backup volume [217](#)
 - VSAM data sets [204](#)
 - VTOC copy data set, create [202](#)
 - wait for spill volume [217](#)
 - backup data sets, changing names [503](#)
 - BACKUP FREQUENCY attribute [180](#)
 - BACKUP parameter
 - ADDVOL command [192](#)
 - DEFINE command [192](#)
 - INUSE subparameter [191](#)
 - QUERY command [571](#)
 - RECYCLE command [539](#)
 - RELEASE command [600](#)
 - SETSYS command [189](#)
 - TAPECOPY command [438](#)
 - BACKUP SYSTEM/SYSPLEX NAME attribute [178](#)
 - backup tape, missing [649](#)
 - backup tape, overwritten [637](#)
 - backup version
 - creating while data set open for update [191](#)
 - journal [496](#), [507](#)
 - maximum number per data set [303](#)
 - BACKUPCONTROLDATASET parameter, UPDATE command [516](#)
 - BACKUPCOPIES subparameter, CDSVERSIONBACKUP parameter, SETSYS command [497](#)
 - BACKUPDEVICECATEGORY parameter, ADDVOL command [103](#)
 - BACKUPDEVICECATEGORY subparameter, CDSVERSIONBACKUP parameter, SETSYS command [498](#)
 - BACKUPPREFIX parameter, SETSYS command [189](#)
 - BACKUPVOLUME parameter, LIST command [560](#)
 - BACKVOL
 - optional parameters
 - DUMP [222](#)
 - FREQUENCY [310](#)
 - INCREMENTAL [232](#)
 - RETENTIONPERIOD [222](#)
 - TOTAL [230](#)
 - VERSIONS [310](#)
 - PRIMARY parameter
 - interrupting and restarting backup processing [486](#)
 - interrupting and restarting dump processing [490](#)
 - redefining the backup cycle [446](#)
 - required parameters
 - CONTROLDATASETS [505](#)
 - PRIMARY [309](#), [310](#)
 - VOLUME [222](#)
 - BACKVOL command
 - backing up
 - non-SMS-managed volumes [789](#)
 - SMS-managed volumes [779](#)

BACKVOL command (*continued*)

- dumping
 - non-SMS-managed volumes [789](#)
 - SMS-managed volumes [779](#)
- examples [788](#), [795](#)
- overview [779](#)
- syntax

- non-SMS-managed volumes [790](#)
- SMS-managed volumes [780](#)

BACKVOL command with SMS

- optional parameters (control data sets)
 - BACKUPDEVICECATEGORY [783](#)
 - DATAMOVER [783](#)
 - NULLJOURNALONLY [784](#)
- optional parameters (volumes)
 - BACKUPDEVICECATEGORY [784](#)
 - DUMP [785](#)
 - DUMPClass [785](#)
 - INCREMENTAL [785](#)
 - MAXSTACK [786](#)
 - MINSTACK [786](#)
 - RETENTIONPERIOD [786](#)
 - STACK [786](#)
 - TERMINAL [788](#)
 - TOTAL [785](#)
- required parameters
 - CONTROLDATASETS [781](#)
 - STORAGEGROUP [781](#)
 - VOLUMES [782](#)

BACKVOL command, non-SMS

- optional parameters (control data sets)
 - BACKUPDEVICECATEGORY [792](#)
 - DATAMOVER [792](#)
 - NULLJOURNALONLY [792](#)
- optional parameters (volumes)
 - BACKUPDEVICECATEGORY [793](#)
 - DUMP [793](#)
 - DUMPClass [794](#)
 - FREQUENCY [794](#)
 - INCREMENTAL [793](#)
 - RETENTIONPERIOD [794](#)
 - STACK [795](#)
 - TERMINAL [795](#)
 - TOTAL [793](#)
 - UNIT [795](#)
- required parameters
 - CONTROLDATASETS [791](#)
 - PRIMARY [791](#)
 - VOLUMES [791](#)

- summary of parameters [790](#)

base cluster [253](#)

base data component [253](#)

base index component [253](#)

BCDS scan, EXPIREBV processing

- determining starting and ending point [234](#), [235](#)

BCDSBACKUPDSN subparameter, CDSVERSIONBACKUP parameter, SETSYS command [497](#)

BDELETE command

- backup volume cleanup process [219](#)
- examples [799](#)
- FROMVOLUME parameter [311](#)
- optional parameters

BDELETE command (*continued*)

- optional parameters (*continued*)

FROMVOLUME [799](#)

VERSIONS [799](#)

overview [797](#)

required parameters

ALL [798](#)

dsname [797](#)

pathname [798](#)

syntax [797](#)

VERSIONS parameter [311](#)

bypass enqueue processing between dump and DFP

functions [458](#)

BYPASS subparameter, DATASETCONFLICT parameter, ARECOVER command [394](#), [402](#)

C

CANCEL

- DFSMSshm queued requests [604](#)

CANCEL command

- examples [803](#)

- for DFSMSshm [801](#)

optional parameters

DATASETNAME [802](#)

FILENAME [802](#)

REQUEST [802](#)

SASINDEX [803](#)

TCBADDRESS [803](#)

USERID [802](#)

syntax [801](#)

canceling a TSO request [686](#)

candidate data sets

- extent reduction, migrate, SMS-managed storage [71](#)

- for backup, determining, SMS-managed storage [68](#)

- for both migration and extent reduction, migrate, SMS-managed storage [71](#)

- for extent reduction, determining, SMS-managed storage [69](#)

- for migration, determining, SMS-managed storage [67](#)

capacity utilization

- specifying [543](#)

cartridge-type tape volumes

- making from reel-type or multiple-file-format tapes [542](#)

case 1: damaged CDS, full journal [629](#)

case 10: correcting errors within the common recall queue [647](#)

case 11: recovering a deleted ML1 data set without a backup [649](#)

case 12: recovering from a large-scale cyber event [650](#)

case 2: damaged journal, undamaged CDS [632](#)

case 3: full journal, undamaged CDS [632](#)

case 4: structurally damaged CDS, missing journal records [633](#)

case 5: time sequence gaps in CDS records, journal records also missing [633](#)

case 6: overwritten migration tape [634](#)

case 7: overwritten backup tape [637](#)

case 8: Damaged ML1 volumes [641](#)

case 9: reestablish access to previously deleted migrated data sets (no backup exists, ml2 only) [646](#)

catalog

- considerations

- ARECOVER processing [390](#)

- catalog (*continued*)
 - considerations (*continued*)
 - migration processing [78](#)
 - inline backup of data sets [447](#)
 - RACF maintenance group [86](#)
 - catalog information data set [263](#)
 - cataloged data sets, backing up RACF discrete profiles for [206](#)
 - CATALOGEDDATA subparameter, NONSMSVERSIONS parameter, EXPIREBV command [310](#)
 - categories of DFSMSHsm-owned tape volumes, copying [438](#)
 - CDS (control data set)
 - accuracy, ensuring [519](#)
 - backing up
 - determining data mover being used [505](#)
 - manually [505](#)
 - specifying DASD [498](#)
 - specifying tape [498](#)
 - specifying the data mover [494](#)
 - using concurrent copy [494](#)
 - when accessing in RLS mode [200](#), [494](#), [498](#), [504](#)
 - backup exit ARCCBEXT [499](#)
 - backup tapes, protecting [499](#)
 - determining if accurate [517](#)
 - device type for backup, specifying [498](#)
 - displaying the CDS serialization technique [505](#)
 - displaying the key boundaries of non-key-range multicenter CDSs [505](#)
 - dumping [456](#)
 - maintaining [493](#)
 - management class support [495](#)
 - MASH configurations [526](#)
 - migration
 - recovering using multiple DASD backup data sets [519](#)
 - recovering using multiple tape backup data sets [524](#)
 - multicenter considerations [502](#)
 - multiple backup data sets
 - DASD [499](#)
 - description [497](#)
 - tape [501](#)
 - names of backup data sets
 - changing [503](#)
 - specifying [497](#)
 - obtaining information from [559](#)
 - performance considerations [525](#)
 - protecting backup tapes [499](#)
 - recovered, updating [516](#)
 - recovery
 - description [507](#)
 - examples [519](#)
 - multiple backup data sets, using UPDATEC [513](#)
 - recovery functions
 - enhanced, implementing using JCL [508](#)
 - enhanced, requirements for using [508](#)
 - repairing errors [517](#)
 - specifying how to back up [493](#)
 - tasks for maintaining [493](#)
 - VSAM considerations [516](#)
 - CDS damaged, full journal [629](#)
 - CDS damaged, missing journal records [633](#)
 - CDS records with time sequence gaps, journal records missing [633](#)
 - CDS support [1112](#), [1113](#)
 - CDS undamaged, full journal [632](#)
 - CDS undamaged, journal damaged [632](#)
 - CDSVERSIONBACKUP parameter, SETSYS command
 - recovering the CDSs using UPDATEC [513](#)
 - changed data set on volumes, non-SMS-managed, backing up all [309](#)
 - CHANGEDONLY subparameter, INCREMENTALBACKUP parameter, SETSYS command [191](#)
 - changes
 - in storage group, recall problems with GUARANTEED SPACE VSAM data sets [153](#)
 - in storage group, recover problems with GUARANTEED SPACE VSAM data sets [451](#)
 - in storage-class-to-data set mapping, recall problems with GUARANTEED SPACE VSAM data sets [153](#)
 - in storage-class-to-data set mapping, recover problems with GUARANTEED SPACE VSAM data sets [452](#)
 - changing
 - backup conditions for a data set [308](#)
 - names of backup data sets [503](#)
 - characteristics of dump classes, specifying [171](#)
 - characteristics of recovered profile [462](#)
 - CHECKFIRST parameter
 - RECYCLE command [541](#)
 - checking integrity age, non-SMS-managed data sets [111](#)
 - checking, authorization during profile recovery [462](#)
 - checklists, aggregate backup [327](#)
 - checkpointed data sets
 - and extent reduction [69](#), [179](#)
 - and migration eligibility [67](#), [179](#)
 - DFSMSHsm patch to force migration [67](#), [179](#)
 - CIDS [263](#)
 - classes
 - dump, specifying characteristics [171](#)
 - cleanup
 - backup volume [218](#)
 - processing [218](#)
 - closing and processing the activity logs [602](#)
 - Cloud password [156](#)
 - cloud storage
 - integrating [58](#)
 - coexistence considerations
 - DFSMSHsm ABACKUP output data sets [354](#)
 - DFSMSHsm aggregate groups and management classes [339](#)
 - command
 - ABACKUP authority
 - comprehensive [341](#)
 - restricted [341](#)
 - access with RACF FACILITY class [625](#)
 - ARECOVER authority
 - comprehensive [377](#)
 - restricted [377](#)
 - availability management
 - example system [165](#)
 - non-SMS-managed storage [308](#)
 - SMS-managed storage [221](#)
 - migrating
 - individual data sets [89](#)
 - SMS-managed volume [91](#)
 - storage group [92](#)
 - recalling

- command (*continued*)
 - recalling (*continued*)
 - direct access data sets [93](#)
 - SMS-managed data sets [92](#)
 - recovery and restore [240](#)
 - space management
 - all primary volumes, non-SMS-managed [124](#)
 - non-SMS-managed data [123](#)
 - SMS-managed data [88](#)
 - syntax diagrams [liv](#)
 - transitioning
 - individual data sets [90](#)
 - SMS-managed volume [91](#)
 - storage group [92](#)
 - volume space management, non-SMS-managed [124](#)
- COMMAND OR AUTO MIGRATE attribute [37](#)
- command space management
 - concept [32](#)
 - tasks [88](#)
- commands
 - ABACKUP [324](#), [691](#)
 - ADDVOL
 - for backup volumes [192](#)
 - for dump volumes [177](#)
 - for ML2 volumes [55](#)
 - for primary volumes [102](#)
 - ALTERDS [308](#)
 - ARECOVER [723](#)
 - AUDIT [517](#)
 - BACKDS [222](#), [309](#)
 - backup
 - preventing processing [449](#)
 - BACKVOL [230](#), [309](#), [779](#)
 - BDELETE [797](#)
 - COPYPOOL [562](#)
 - DELETE [89](#)
 - deleting migrated data sets [88](#)
 - DFSMSHsm commands
 - ALTERDS [715](#)
 - ALTERPRI [719](#)
 - AUDIT [737](#), [1255](#), [1256](#), [1280](#)
 - AUTH [755](#)
 - BACKDS [759](#), [775](#)
 - BACKFILE [773](#), [777](#)
 - CANCEL [801](#)
 - DEFINE [809](#)
 - DELETE [839](#)
 - DELVOL [841](#)
 - DISPLAY [849](#)
 - FIXCDS [863](#)
 - FRBACKUP [865](#)
 - FRDELETE [873](#)
 - FRRECOV [883](#)
 - HSEND CMD [913](#)
 - HSENDU [917](#)
 - LIST [1341](#)
 - LOG [953](#)
 - MIGRATE [955](#)
 - PATCH [979](#)
 - QUERY [981](#)
 - STOP [1223](#)
 - SWAPLOG [1225](#)

- commands (*continued*)
 - DFSMSHsm commands (*continued*)
 - TRAP [1241](#)
 - UPDATEC [1243](#), [1253](#)
 - UPDTCDS [1253](#)
 - environment configuration
 - Java hbackup [1250](#)
 - examples
 - ABACKUP [695](#)
 - ADDVOL [711](#)
 - ALTERDS [717](#)
 - ALTERPRI [721](#)
 - ARECOVER [733](#)
 - AUTH [757](#)
 - BACKDS [771](#)
 - BACKVOL [788](#), [795](#)
 - BDELETE [799](#)
 - CANCEL [803](#)
 - DDELETE [806](#)
 - DEFINE [836](#)
 - DELETE [840](#)
 - DELVOL [845](#)
 - EXPIREBV [859](#)
 - FRBACKUP [870](#)
 - FRDELETE [874](#)
 - FREEVOL [881](#)
 - FRRECOV [892](#)
 - HOLD [909](#)
 - HSEND CMD [915](#)
 - Java hbackup [1251](#)
 - LIST [947](#)
 - LOG [953](#)
 - MIGRATE [964](#), [971](#)
 - ONLYIF [975](#)
 - RECALL [994](#)
 - RECOVER [1011](#)
 - RECYCLE [1025](#)
 - RELEASE [1039](#)
 - REPORT [1051](#)
 - SETMIG [1055](#)
 - SETSYS [1211](#)
 - STOP [1224](#)
 - SWAPLOG [1226](#)
 - TAPECOPY [1233](#)
 - TAPEREP [1238](#)
 - UPDATEC [1244](#)
 - UPDTCDS [1254](#)
 - EXPIREBV [851](#)
 - FRBACKUP [261](#), [283](#)
 - FRDELETE [283](#)
 - FREEVOL [877](#)
 - FRRECOV [270](#), [283](#)
 - general information
 - abbreviating commands [685](#)
 - comments [686](#)
 - data set naming conventions [686](#)
 - delimiters [686](#)
 - keyword parameters [685](#)
 - line continuations [686](#)
 - operator commands, listed [681](#)
 - positional parameters [684](#)
 - providing protection for [684](#), [755](#)
 - sequence of commands [687](#)
 - storage administrator commands, listed [682](#)

- commands (*continued*)
 - general information (*continued*)
 - submission of commands [687](#)
 - system programmer commands, listed [683](#)
 - TSO attention key [686](#)
 - user commands, listed [681](#)
 - using DFSMSHsm commands [684](#)
 - without [689](#)
 - without RACF [688](#)
 - HBACKDS [222](#), [447](#)
 - HDELETE [89](#)
 - HOLD [449](#), [596](#), [895](#)
 - HRECALL [92](#)
 - Java hbackup command [1247](#)
 - LIST [559](#), [919](#)
 - MIGRATE [89](#), [90](#), [532](#)
 - ONLYIF [973](#)
 - QUERY [570](#), [1389](#)
 - RECALL [92](#), [991](#)
 - RECOVER [997](#)
 - RECYCLE [539](#), [1015](#)
 - RELEASE [1029](#)
 - REPORT [1043](#), [1401](#)
 - SETMIG [1053](#)
 - SETSYS [1057](#)
 - storage administrator mode [1245](#)
 - TAPECOPY [436](#), [1227](#)
 - TAPEREP [440](#), [1235](#)
 - to operate DFSMSHsm [678](#)
 - UPDATEC [516](#)
- commandsc
 - HOLD [283](#)
- common recall queue
 - connecting to list structure [131](#)
 - deleting the structure [133](#)
 - disconnecting from [132](#)
 - failures
 - coupling facility link [134](#)
 - coupling facility structure [133](#)
 - full list structure [134](#)
 - primary address space [134](#)
 - z/OS image [134](#)
 - populating [131](#)
 - processing
 - recall requests [132](#)
 - shutdown [133](#), [596](#)
 - selecting requests on [132](#)
- common recall queue (CRQ) [647](#)
- Common Recall Queue (CRQ) [719](#), [745](#), [901](#), [985](#), [1033](#), [1117](#), [1119](#)
- common volumes, example [19](#)
- COMMONQUEUE [131](#), [134](#)
- compaction
 - during migration [75](#)
 - specifying utilization on a 3480X [542](#)
- complete volume, restoring from a dump copy [298](#)
- comprehensive ABACKUP command authority [341](#)
- comprehensive ARECOVER command authority [377](#)
- compression
 - dump data [176](#)
- concurrent copy
 - specifying for volume dumps [169](#)
 - specifying whether to use [179](#)
 - using to back up the CDSs [494](#)
- concurrent copy (*continued*)
 - with data set backup [453](#)
 - with RACF [454](#)
- conditions for a data set, backup, changing [308](#)
- configurations
 - general pool [106](#)
 - pool, commands that affect in JES3 [106](#)
- conflict resolution data set [373](#), [403](#)
- connected set processing [538](#)
- considerations
 - additional, for recall [86](#)
 - availability management
 - extended and compressed VSAM data sets [464](#)
 - extended SAM data sets [464](#)
 - sequential data striping [464](#)
 - catalog, during ARECOVER processing [390](#)
 - copy pool [288](#)
 - DFSMSHsm commands that affect pool configuration in JES3 [106](#)
 - dump [455](#)
 - FRBACKUP [261](#)
 - FRRECOV [274](#)
 - multicenter CDS [502](#)
 - multitasking, for SDSP data sets [147](#)
 - sequential data striping
 - aggregate backup processing [431](#)
 - aggregate recovery processing [432](#)
 - volume pooling [107](#)
 - VSAM, for control data sets [516](#)
- consolidating valid data on tape volumes [537](#)
- contact
 - z/OS [1413](#)
- contention
 - backup volume [445](#)
 - migration needs volume [137](#)
 - multiple-processing-unit environment [139](#)
 - recycle needs volume [138](#)
 - SDSP data sets [142](#)
 - single-processing-unit environment [137](#)
- control file
 - aggregate backup output [351](#)
 - input for recovery [373](#)
- conversion of data sets to GUARANTEED SPACE
 - recall problems [154](#)
 - recover problems with VSAM data sets [452](#)
- conversion of non-SMS-managed volume to SMS-managed [10](#)
- CONVERT parameter
 - in direct-to-tape environment [528](#)
 - MIGRATE command
 - non-SMS-managed data sets [528](#)
 - SMS-managed data sets [527](#)
 - with BDAM data sets [528](#)
- converting partial tapes to full tapes [440](#)
- copies
 - of DFSMSHsm-owned tape volumes [436](#)
- Copies, ISMF panel parameter [336](#)
- copy pool
 - considerations [274](#), [288](#)
 - decreasing volumes [289](#)
 - deleting unneeded volumes [283](#)
 - increasing volumes [289](#)
 - information, obtaining [284](#), [562](#)

- copy pool (*continued*)
 - recovery of [270](#)
 - setup [257](#)
 - using [257](#)
- copy pool backup storage group type [259](#)
- copy pool construct [257](#)
- copying
 - ABARs aggregate [368](#)
 - categories of DFSMSHsm-owned tape volumes [438](#)
 - individual volumes, disaster backup [439](#)
- CPU processing time, tracking for ABARS [353](#), [413](#)
- create a VTOC copy data set for the volume [202](#)
- creating
 - ABARS activity record [381](#)
 - backup profiles, parameters used [207](#)
 - selection data sets [329](#)
- creation
 - retained backup copies [184](#)
- current backup version, definition [114](#)
- cycle
 - backup [192](#)
 - dump [170](#)
 - redefining backup [446](#)
 - start date
 - backup [192](#)
 - dump [170](#)
- CYCLESTARTDATE subparameter
 - BACKUP parameter, DEFINE command [192](#)
 - DUMPCYCLE parameter, DEFINE command [170](#)

D

- daily
 - backup volume selection
 - DASD [209](#)
 - DASD, initial [209](#)
 - DASD, subsequent [211](#)
 - tape [211](#)
 - tape, subsequent [212](#)
 - initial DASD or tape backup volume selection [213](#)
 - limit
 - not reached, DASD or tape daily backup volume selection [214](#)
 - reached or exceeded, DASD or tape daily backup volume selection [213](#)
 - subsequent DASD or tape backup volume selection [215](#)
- DAILY parameter, RECYCLE command [539](#)
- damaged CDS, full journal [629](#)
- damaged CDS, missing journal records [633](#)
- damaged journal, undamaged CDS [632](#)
- damaged ML1 volume [641](#)
- damaged ML1 volume, recovering backup versions from [554](#)
- DAOPTION parameter, RECALL command [93](#), [125](#)
- DASD (direct access storage device)
 - and tape daily backup volume, selecting [212](#)
 - backing up control data sets to multiple backup data sets [499](#)
 - changing from DASD to tape [504](#)
 - changing from multiple tape backup data sets [503](#)
 - choosing for multiple backup data sets [498](#)
 - daily backup volume selection [209](#)
 - daily backup volume selection, initial [209](#)
 - daily backup volume selection, subsequent [211](#)

- DASD (direct access storage device) (*continued*)
 - I/O buffering technique to use for dump by DFSMSdss [168](#)
 - migration level 2 volume, moving data sets to other migration level 2 volumes [534](#)
 - migration volume, preventing further migration to [535](#) or tape, initial daily backup volume selection [213](#) or tape, subsequent daily backup volume selection [215](#)
 - specifying for CDS backup [498](#)
 - spill backup volume, subsequent selection [218](#)
- data class, use of [11](#)
- data file
 - aggregate backup output [351](#)
 - description [324](#)
 - input for recovery [373](#)
- data flow in DFSMSHsm [9](#)
- data mover
 - DFSMSdss, specifying for CDS backup [494](#)
 - DFSMSHsm, specifying for CDS backup [496](#)
- data on dump volumes becomes invalid, specifying when [174](#)
- data set backup
 - inline backup [230](#)
 - specifying tasks [224](#)
 - switching tapes [227](#)
 - taking a tape away [228](#)
 - targeting to device types [228](#)
 - using concurrent copy [228](#), [453](#)
 - using the data set backup function [224](#)
- data set naming conventions [686](#)
- data sets
 - accessing when profile cannot be recovered [462](#)
 - allocating recalled to volumes, non-SMS-managed [120](#)
 - always open, backing up [191](#)
 - automatic recall during opening VSAM, non-SMS-managed [121](#)
 - automatic recall for command scratching, non-SMS-managed [121](#)
 - backing up
 - individual, non-SMS-managed [309](#)
 - migrated [200](#)
 - serialization during [191](#)
 - SMS-managed [222](#)
 - backup
 - conditions for, changing [308](#)
 - uncataloged [307](#)
 - versions of specific data sets [239](#)
 - VSAM [204](#)
 - VTOC copy, create [202](#)
 - changed
 - backing up all eligible in one or more storage groups [232](#)
 - backing up all eligible on a volume, SMS-managed [232](#)
 - backing up all, non-SMS managed [309](#) only, specifying whether to back up [190](#)
 - command
 - migration [89](#)
 - recall of direct access [93](#)
 - recall of SMS-managed [92](#)
 - transitioning [90](#)
 - contention, SDSP [142](#)
 - control

data sets (*continued*)

control (*continued*)

- backing up [497](#)
- DASD backup data sets [499](#)
- multiple backup data sets [497](#)
- multiple tape backup data sets [501](#)
- recovering, examples [519](#)
- recovery [507](#)
- recovery and multiple backup data sets, using UPDATEC [513](#)

controlling migration, non-SMS-managed [124](#)

deleting

- Deleting empty dump containers [195](#)
- expired dump copies [195](#), [239](#)
- from migration volumes with the MIGRATE command [532](#)
- migrated [88](#)
- migrated data set fails, JES3 [122](#)

deletion

- ADDVOL command [103](#)
- description [89](#)
- exit ARCADEXT [114](#)
- non-SMS-managed volumes [114](#)
- partitioned data set [89](#)
- priority [135](#)

determining

- candidates for backup, non-SMS-managed volumes [112](#)
- candidates for extent reduction, non-SMS-managed volumes [112](#)
- candidates for migration, non-SMS-managed volumes [112](#)

dump VTOC copy, write [198](#)

dynamically allocated during aggregate backup processing [345](#)

dynamically allocated during aggregate recovery processing [410](#)

excess dump VTOC copy

- deleting [198](#)
- uncataloging [198](#)

expired backup versions

- deleting from backup volumes [233](#)
- non-SMS-managed, deleting [310](#)
- non-SMS-managed, processing [310](#)

from dump tapes, controlling restoring [172](#)

generation, deferred [149](#), [460](#)

inline backup [230](#)

journal

- DASD backup data sets [499](#)
- multiple backup data sets [497](#)
- multiple tape backup data sets [501](#)
- recovering, examples [519](#)
- replacing using multiple DASD backup data sets [522](#)

migrated

- deleting [88](#)
- profile backup for [207](#)
- recovering [551](#)
- uncataloging [625](#)

migrating [532](#)

migration exit ARCMDEXT [118](#)

migration of VSAM, non-SMS-managed volumes [115](#)

data sets (*continued*)

moving

- between level 0 volumes, example [528](#)
- from a DASD ML2 volume to other ML2 volumes [534](#)
- from DASD backup volumes [535](#)
- from migration level 1 volumes [530](#), [533](#)
- from ML1 volumes [530](#)
- from one level 0 volume to other level 0 volumes [527](#)
- journal [518](#)
- with FREEVOL [531](#)

name

- backup versions [204](#)
- backup VTOC copy [202](#)
- conflict resolution, aggregate recovery [402](#)
- dump copy [197](#)
- dump VTOC copy [198](#)
- format of SDSP [556](#)
- migrated data sets [73](#)
- name of backup copy data sets [206](#)
- name of dump copy data sets [197](#)

non-SMS-managed

volumes

- retirement [103](#), [114](#)

obtaining information [560](#)

- on a volume, backing up [230](#)

password protection on DASD [151](#)

pools, definition [19](#)

preventing processing during volume space management [149](#)

priority order

- non-SMS-managed

[113](#)

- SMS-managed [72](#)

RACF

- discrete profile [618](#)
- discrete profiles for, backing up [206](#)
- generic profile [617](#)
- protection [617](#)

RACF-indicated

- backing up [625](#)
- deleting [150](#)
- recalling [151](#)
- recovering or restoring [461](#)

recall

- during opening non-VSAM, non-SMS-managed [121](#)

- non-SMS-managed [119](#)

recalling to a specific volume, non-SMS-managed [124](#)

recovered aggregate [373](#)

recovered, maintaining password protection for [463](#)

recovery

- description [241](#)
- of direct access [248](#)
- recovery of extended format [248](#)
- VSAM [314](#)

restore

- directly with DFSMSdss [458](#)
- from a DFSMSShsm dump copy, non-SMS-managed [314](#)
- from a DFSMSShsm dump copy, SMS-managed [251](#)
- individual from dump tapes, controlling [172](#)
- of a retired data set [315](#)

data sets *(continued)*

- SDSP
 - managing [555](#)
 - removing [557](#)
 - security [625](#)
- sometimes open, backing up [191](#)
- space management with discrete data set profiles [150](#)
- specific
 - delete backup versions [239](#)
 - recovering, non-SMS-managed [311](#)
- specifying whether to back up [190](#)
- temporary, deleting [202](#)
- unmatched
 - backup [202](#)
 - space management [66](#)
- VSAM, restore of [253](#)
- with discrete data set profiles, space management of [150](#)

data sets, reestablishing access to deleted migrated [646](#)

data-set-changed indicator

- resetting [203](#)
- setting only when backup serialized [203](#)
- specifying whether to reset [173](#)

DATASET parameter

- ARECOVER command [393](#)
- LIST command [560](#)

DATASETCONFLICT parameter, ARECOVER command [387](#), [394](#)

DATASETCONTROL parameter, AUDIT command [517](#)

DATASETNAME parameter

- ARECOVER command [394](#)
- MIGRATE command [528](#)
- on DFSMSHsm ARECOVER command [725](#)

DATASETRESTORE subparameter, DUMPCLASS parameter, DEFINE command [172](#)

date

- hold volumes [122](#)
- prevent migration [117](#)

DATE parameter, ARECOVER command [394](#)

DATE parameter, RECOVER command [241](#)

date-last-referenced field for recovery [459](#), [497](#)

date-last-referenced field, maintenance and use for space management [78](#)

date-of-scratch field, resetting, recovery processing [249](#)

DAY subparameter, DUMPCLASS parameter, DEFINE command [172](#)

day to dump data to a dump class, specifying [172](#)

days on which dump is to occur, specifying [170](#)

DAYS parameter, MIGRATE command [532](#)

DAYS parameter, SETSYS command [101](#)

DDELETE command

- examples [806](#)
- optional parameters
 - LASTCOPY [806](#)
- required parameters
 - DATE and TIME [806](#)
 - storagegroup [805](#)
 - volumes [806](#)
- syntax [805](#)

DDELETE Command [805](#)

DEBUG parameter, SETSYS command [149](#), [587](#)

default management class [40](#)

deferred generation data sets [149](#), [460](#)

DEFINE

- example [384](#)
- issuing the command [382](#)
- optional parameters
 - ALL [383](#)
 - ARPOOL [382](#)
 - AUTOREUSE [171](#)
 - BACKUP [192](#)
 - CYCLESTARTDATE [170](#)
 - DATASETRESTORE [172](#)
 - DAY [172](#)
 - DISPOSITION [173](#)
 - DUMPCLASS [171](#)
 - DUMPCYCLE [170](#)
 - FREQUENCY [172](#)
 - LOVOLS [383](#)
 - ML1VOLS [383](#)
 - POOL [105](#)
 - PRIMARYSPMGMTCYCLE [46](#)
 - RESET [173](#)
 - RETENTIONPERIOD [174](#)
 - SECONDARYSPMGMTCYCLE [46](#)
 - STACK [169](#), [175](#)
 - TAPEEXPIRATIONDATE [174](#)
 - UNIT [174](#)
 - VOLUMEPOOL [107](#)
 - VTOCOPIES [176](#)
- pool configuration in JES3 [107](#)

DEFINE command

- examples [836](#)
- for DFSMSHsm [809](#)
- optional parameters
 - ARPOOL [813](#)
 - BACKUP [814](#)
 - CYCLESTARTDATE [814](#)
 - DUMPCLASS [815](#)
 - DUMPCLASS AUTOREUSE [816](#)
 - DUMPCLASS CLOUD | TAPE [817](#)
 - DUMPCLASS DATASETRESTORE [818](#)
 - DUMPCLASS DAY [819](#)
 - DUMPCLASS DISABLE [820](#)
 - DUMPCLASS DISPOSITION [820](#)
 - DUMPCLASS FREQUENCY [822](#)
 - DUMPCLASS HWCOMPRESS [823](#)
 - DUMPCLASS MAXSTACK [825](#)
 - DUMPCLASS MINSTACK [825](#)
 - DUMPCLASS NOAUTOREUSE [816](#)
 - DUMPCLASS NODATASETRESTORE [818](#)
 - DUMPCLASS NORESET [824](#)
 - DUMPCLASS RESET [824](#)
 - DUMPCLASS RETENTIONPERIOD [824](#)
 - DUMPCLASS STACK [825](#)
 - DUMPCLASS TAPEEXPIRATIONDATE [827](#)
 - DUMPCLASS UNASSIGNTAPEUNIT [828](#)
 - DUMPCLASS UNIT [827](#)
 - DUMPCLASS VTOCOPIES [828](#)
 - DUMPCLASS ZCOMPRESS [829](#)
 - DUMPCYCLE [829](#)
 - FASTREPLICATIONDUMP [815](#)
 - MIGRATIONCLEANUPCYCLE [830](#)
 - MIGRATIONLEVEL2 [831](#)
 - POOL [832](#)
 - PRIMARYSPMGMTCYCLE [832](#)
 - RECOVERRESET [823](#)

- DEFINE command (*continued*)
 - optional parameters (*continued*)
 - SECONDARYSPMGMTCYCLE [830](#)
 - SWITCHTAPES [833](#)
 - TAPE CLOUD [835](#)
 - VOLUMEPOOL [834](#)
- DEFINE DUMPCLASS command
 - using [277](#)
- defining
 - aggregate group [336](#)
 - automatic backup window [487](#)
 - automatic dump start window [490](#)
 - automatic primary space management window [164](#)
 - automatic secondary space management window [164](#)
 - backup volumes to DFSMSHsm [192](#)
 - dump volumes to DFSMSHsm [177](#)
 - new processing window [164](#)
 - parameters of the recycle process [537](#)
 - pools of volumes [105](#)
 - primary volumes to DFSMSHsm [102](#)
 - volume pool configuration in JES3 [107](#)
 - when backup cycle starts [192](#)
- definition
 - current backup version [114](#)
 - data set pool [19](#)
- DELETE [89](#)
- DELETE command
 - examples [840](#)
 - for DFSMSHsm [839](#)
 - optional parameters
 - PURGE [840](#)
 - required parameters
 - dsname [839](#)
- DELETEBYAGE subparameter, ADDVOL command [103](#)
- DELETEBYAGE subparameter, VOLUME parameter, MIGRATE command [532](#)
- deleted migrated data sets [646](#), [649](#)
- DELETEIFBACKEDUP subparameter
 - ADDVOL command [103](#)
 - NONSMSVERSIONS parameter, EXPIREBV command [310](#)
 - VOLUME parameter, MIGRATE command [532](#)
- deleting
 - backup profiles [207](#)
 - backup versions of specific data sets, non-SMS-managed [311](#)
 - data set [89](#)
 - excess dump VTOC copy data sets [198](#)
 - expired backup versions from backup volumes, SMS-managed [233](#)
 - expired backup versions of non-SMS-managed data sets [310](#)
 - expired data sets, non-SMS-managed volumes [111](#)
 - expired data sets, SMS-managed storage [66](#)
 - list and utility data sets, non-SMS-managed volumes [111](#)
 - list data sets, specifying age [100](#)
 - migrated data sets [88](#)
 - non-SMS-managed data sets from migration volumes with the MIGRATE command [532](#)
 - RACF-indicated data sets [150](#)
 - tape volumes from the HSMABR RACF tape volume set [340](#)
 - temporary data sets [202](#)
- deleting (*continued*)
 - temporary data sets, SMS-managed storage [66](#)
 - utility data sets, specifying age [100](#)
- DELVOL
 - JES3 considerations [537](#)
 - optional parameters
 - PURGE [536](#)
 - REASSIGN [536](#)
 - UNASSIGN [536](#)
 - pool configuration in JES3 [537](#)
 - PURGE parameter, redefining the backup cycle [446](#)
- DELVOL command
 - examples [845](#)
 - for DFSMSHsm [841](#)
 - optional parameters
 - COPYPOOLCOPY [843](#)
 - LASTCOPY [843](#)
 - MARKFULL [843](#)
 - PURGE [843](#)
 - REASSIGN [843](#)
 - UNASSIGN [843](#)
 - required parameters
 - BACKUP [842](#)
 - DUMP [842](#)
 - MIGRATION [842](#)
 - PRIMARY [842](#)
 - volser [842](#)
- demount mounted tapes for command data set backup [900](#)
- description, ISMF panel parameter [336](#)
- device used for backup, specifying [189](#)
- devices
 - supported by DFSMSHsm [16](#)
 - used for backup [16](#)
 - used for migration level 1 [16](#)
 - used for migration level 2 [16](#)
- DFSMSdfp
 - family membership [7](#)
- DFSMSdss
 - automatic recall with [85](#)
 - DASD I/O buffering technique to use for dump [168](#)
 - invoking RESTORE function [459](#)
 - restore data sets or volumes [458](#)
 - specifying as data mover for CDS backup [494](#)
- DFSMSHsm
 - address space, obtaining information about [570](#)
 - allowing all functions to run [600](#)
 - Altering the priority of DFSMSHsm queued requests [606](#)
 - cancel queued request [604](#)
 - coexistence considerations
 - ABACKUP output data sets [354](#)
 - aggregate groups and management classes [339](#)
 - commands
 - ALTERDS [715](#)
 - ALTERPRI [719](#)
 - AUDIT [737](#), [1255](#), [1256](#), [1280](#)
 - AUTH [755](#)
 - BACKDS [759](#), [775](#)
 - BACKFILE [773](#), [777](#)
 - CANCEL [801](#)
 - DEFINE [809](#)
 - DELETE [839](#)
 - DELVOL [841](#)
 - DISPLAY [849](#)

DFSMSHsm (continued)

commands (continued)

- [FIXCDS 863](#)
- [FRBACKUP 865](#)
- [FRDELETE 873](#)
- [FRRECOV 883](#)
- [HSEND CMD 913](#)
- [HSENDU 917](#)
- [LIST 1341](#)
- [LOG 953](#)
- [MIGRATE 955](#)
- [PATCH 979](#)
- [QUERY 981](#)
- [STOP 1223](#)
- [SWAPLOG 1225](#)
- [TRAP 1241](#)
- [UPDATEC 1243](#)
- [UPDTCDS 1253](#)

data flow 9

devices supported 16

functions

- for non-SMS-managed volumes [14](#)
- for SMS-managed volumes [10](#)

overview 9

partitioning 593

RACF authority needed to perform 620

log information, printing 607

managing spanning data sets 545, 546, 548

media, managing 527

member of DFSMSDfp family 7

objective in using 10

operations, obtaining statistical information 574

partitioning functions 593

preventing functions from running 596

previously prevented functions, allowing to run 600

processing, determining results without moving data sets 587

RACF concepts 620

service request from application programs 558

starter set 8

stopping support activities 595

tape security when using with DFSMSrmm 592

tips 545, 546, 548

Updating the DFSMSHsm control data sets 616

using with DFSMSrmm 591

DFSMSHsm commands

ALTERDS 715

ALTERPRI 719

AUDIT 737, 1255, 1256, 1280

AUTH 755

BACKDS 759, 775

BACKFILE 773, 777

CANCEL 801

DEFINE

optional parameters 813

syntax 810

DELETE 839

DELVOL 841

DISPLAY 849

FIXCDS 863

FRBACKUP 865

FRDELETE 873

FRRECOV 883

HSEND CMD 913

DFSMSHsm commands (continued)

HSENDU 917

LIST 1341

LOG 953

MIGRATE 955

PATCH 979

QUERY 981

STOP 1223

SWAPLOG 1225

TRAP 1241

UPDATEC 1243

UPDTCDS 1253

DFSMSHsm fast replication

data set recovery

selecting a copy method 1142

FlashCopy relationships

selecting type 1142

volume pairing messages (ARC1809I)

suppressing 1142

DFSMSHsm fast replication, data set recovery

defining dump classes 277

limiting all recovery requests 1150

limiting recovery requests 281

limiting tape recovery requests 1150

performance 277

recovering data sets 271

selecting recovery method 276

DFSMSHsm fast replication, tape support

deleting unneeded dump copies 283

dumping backups to tape 264

DFSMSHsm fast replication, volume recovery

limiting volume recovery requests 1151

DFSMSHsm-managed

automatic space management, SMS-managed storage 65

volumes

dumping 196

obtaining information 567, 568

DFSMSHsm-owned

tape volumes

copying categories 438

data set name of backup tape 75

data set name of migration tape 74

data set name of TAPECOPY of backup tape 206

data set name of TAPECOPY of migration tape 206

making backup copies 435

making copies 436

protecting 558

DFSMSrmm

invocation by tape volume exit ARCTVEXT 439

retaining DFSMSHsm tape volumes

by expiration date 591

by vital record processing 591

tape security when using with DFSMSHsm 592

using with DFSMSHsm 591

DFSMSrmm Report Generator. 566

directory

backing up all eligible changed UNIX files in a 232

groups

backing up all eligible changed UNIX files in a 232

disaster

alternate volumes 442

backup 435

backup copies of individual volumes, making 439

- disaster (*continued*)
 - backup facilities [436](#)
 - recovery [391](#)
 - recovery, planning [435](#)
- disaster backup
 - making copies of individual volumes [439](#)
- discrete
 - profile data set [618](#)
 - profiles, RACF, recovering [461](#)
- DISPLAY
 - EXPIREBV command [233](#)
- DISPLAY command
 - for DFSMSHsm [849](#)
- displaying SMS class definitions during ABACKUP and ARECOVER [409](#)
- DISPOSITION subparameter, DUMPCLASS parameter, DEFINE command [173](#)
- DRAIN parameter, ADDVOL command [535](#)
- DSCOMMAND parameter [449](#)
- dump
 - additional considerations [455](#)
 - automatic full-volume, example system [165](#)
 - class
 - day to dump data to, specifying [172](#)
 - how often to dump to, specifying [172](#)
 - obtaining information [564](#)
 - specifying characteristics [171](#)
 - specifying, SMS-managed [166](#)
 - command [222](#)
 - compress parameter [457](#)
 - control data sets [456](#)
 - controlling automatic reuse of tapes [171](#)
 - copy
 - data set restore from [251](#)
 - list dump volumes for [458](#)
 - restoring a complete volume from [298](#)
 - stacking [169](#), [175](#)
 - cycle [170](#)
 - data on volumes becomes invalid, specifying when [174](#)
 - data-set-changed indicator, specifying whether to reset [173](#)
 - day to dump data to a dump class, specifying [172](#)
 - days on which to dump, specifying [170](#)
 - DFSMSHsm-managed volumes [196](#)
 - DUMP parameter
 - ADDVOL command [177](#)
 - BACKVOL command [222](#)
 - HOLD command [597](#)
 - RELEASE command [601](#)
 - excess VTOC copy data sets, deleting [198](#)
 - excess VTOC copy data sets, uncataloging [198](#)
 - expired copies, deleting [195](#)
 - function
 - overview, non-SMS-managed [14](#)
 - overview, SMS-managed [11](#)
 - how often to dump to dump classes, specifying [172](#)
 - kind of tape units to use, specifying [174](#)
 - maximum number of tasks [169](#)
 - name
 - data sets [197](#)
 - dump copy data sets [197](#)
 - VTOC copy data set [198](#)
 - number of VTOC copies to keep, specifying [175](#)

- dump (*continued*)
 - of migrated data sets, specifying first qualifier [53](#)
 - owned DFSMSHsm volumes [455](#)
 - preventing [456](#)
 - processing
 - determining if volumes can be stacked [193](#)
 - determining tape dump volumes available for output [570](#)
 - example [193](#)
 - restoring individual data sets from dump tapes, controlling [172](#)
 - selecting tape volumes [198](#)
 - specifying volumes, non-SMS-managed [305](#)
 - stacking multiple copies on tape [169](#), [175](#)
 - start window, specifying [168](#)
 - tapes, returning to use [555](#)
 - tasks [166](#)
 - varying invocation [457](#)
 - volume
 - defining to DFSMSHsm [177](#)
 - in one or more storage groups [222](#)
 - list for a dump copy [458](#)
 - non-SMS-managed [308](#)
 - obtaining information [564](#)
 - serialization [197](#)
 - specifying concurrent copy with [169](#)
 - VTOC copies, example [175](#)
 - VTOC copy data set, write [198](#)
 - what to do with volumes, specifying [173](#)
- DUMP CLASS attribute [166](#)
- DUMP SYSTEM/SYSPLEX NAME attribute [166](#)
- DUMPCLASS parameter, DEFINE command [171](#)
- DUMPCLASS parameter, LIST command [564](#)
- DUMPCLASS subparameter, DUMP parameter, ADDVOL command [177](#)
- DUMPCLASS subparameter, FROMDUMP parameter, RECOVER command [252](#)
- DUMPCYCLE parameter, DEFINE command [170](#)
- dumping
 - a volume by command [222](#)
 - volumes in one or more storage groups [222](#)
- DUMPIO parameter, SETSYS command [168](#)
- DUMPVOLUME parameter, LIST command [564](#)
- DUMPVOLUME subparameter, FROMDUMP parameter, RECOVER command [252](#)
- DUPLEX parameter, SETSYS command [436](#)
- duplex tapes
 - and ARECOVER processing [437](#)
 - creating off site [441](#)
 - discussion of function [436](#)
 - specifying with SETSYS command [436](#)
- dynamic data set allocation
 - during aggregate backup processing [345](#)
 - during aggregate recovery processing [410](#)

E

- edit-log data set, printing DFSMSHsm log information [608](#)
- effects of profile modeling on profile recovery [462](#)
- eligibility for level 1 to level 2 migration, specifying for non-SMS-managed data sets [101](#)
- eligible data sets, back up [203](#)
- eligible dump copies, recovery processing [245](#)

- EMERGENCY parameter, SETSYS command, preventing processing of data sets [149](#)
- ENCRYPT parameter
 - DEFINE command [176](#)
- encryption
 - dump data [176](#)
- end BCDS scan, EXPIREBV processing [238](#)
- end of level 0 volume processing [79](#)
- ending
 - a list function [907](#)
 - an audit function [899](#)
 - an expire backup function [905](#)
 - the recycle function [908](#)
 - the report function [909](#)
- ENDKEY parameter, EXPIREBV command [233](#)
- enhanced CDS recovery functions
 - implementing using JCL [508](#)
 - requirements for using [508](#)
- EOD parameter, HOLD command [408](#)
- error
 - in control data sets, repairing [517](#)
 - obtaining system condition information when one occurs [586](#)
 - recovering from journal I/O [518](#)
 - spill backup volume selection [218](#)
 - spill backup volume selection, subsequent [218](#)
- ERRORALTERNATE parameter
 - MARKFULL subparameter [437](#)
- errors
 - codes used in AUDIT reports [1295](#)
 - unexpected during AUDIT [1337](#)
- errors within the common recall queue [647](#)
- event-driven migration
 - concept [24](#)
 - interval [24](#), [25](#)
 - on-demand [24](#), [27](#)
- examples
 - automatic backup [220](#)
 - backing up cataloged and uncataloged data sets with the same name [307](#)
 - backup processing [199](#)
 - CDS backup and recovery [519](#)
 - common volumes [19](#)
 - DASD storage [18](#)
 - dump processing [193](#)
 - dump VTOC copies [175](#)
 - JES3 pooling [107](#)
 - management classes, backup [186](#)
 - management classes, space management [41](#)
 - migration level 1 volumes [19](#)
 - moving data sets between level 0 volumes
 - non-SMS-managed [530](#)
 - SMS-managed [529](#)
 - processors [18](#)
 - recovering the control and journal data sets
 - journal from DASD [522](#)
 - MCDS from DASD [519](#)
 - MCDS when latest MCDS backup version is unusable [524](#)
 - recovering the control data sets [519](#)
 - storage groups [18](#)
 - system [17](#)
 - system, space management [23](#)
 - tape devices [19](#)

- examples (*continued*)
 - unmanaged volumes [19](#)
 - volume pool [19](#)
- excess dump VTOC copy data sets, deleting [198](#)
- EXCLUDE
 - data set [323](#)
 - keyword [332](#)
- EXECUTE parameter
 - ABACKUP command [347](#)
 - ARECOVER command [393](#)
 - EXPIREBV command [233](#)
 - RECYCLE command [539](#)
- exit
 - ABARS backup error, ARCBEXT [342](#)
 - ABARS expiration date, ARCEEXT [342](#)
 - ABARS migration level 2 data set, ARCM2EXT [342](#)
 - called during level 1 to level 2 migration [63](#)
 - CDS backup, ARCCBEXT [499](#)
 - data set deletion, ARCADEXT [115](#)
 - data set migration, ARCMDEXT [118](#)
 - second-level migration data set, ARCMMEXT [63](#), [90](#)
 - space management and backup, ARCSAEXT [66](#)
 - tape data set, ARCTDEXT [63](#), [90](#)
 - tape volume, ARCTVEXT [441](#)
- EXITOFF parameter, SETSYS command, inactivate DFSMSshm exits [149](#)
- EXITON parameter, SETSYS command, activate DFSMSshm exits [149](#)
- EXITS parameter, SETSYS command, preventing processing of data sets [149](#)
- expiration attributes, specifying [36](#)
- EXPIRE AFTER DATE/DAYS attribute [36](#)
- EXPIRE AFTER DAYS NON-USAGE attribute [36](#)
- EXPIREBV
 - BCDS scan [234](#)
 - ending message [239](#)
 - optional parameters
 - CATALOGEDDATA [310](#)
 - DELETEIFBACKEDUP [310](#)
 - ENDKEY [233](#)
 - EXECUTE [233](#)
 - NONSMSVERSIONS [310](#)
 - RESUME [233](#)
 - STARTKEY [233](#)
 - UNCATALOGEDDATA [310](#)
 - overview [851](#)
 - parameter
 - HOLD command [598](#)
 - RELEASE command [602](#)
 - processing [233](#)
 - required parameters
 - DISPLAY [233](#)
 - set up output destination [234](#)
 - starting message [234](#)
 - syntax [852](#)
- EXPIREBV command
 - examples [859](#)
 - optional parameters
 - ABARSVERSIONS [853](#)
 - ENDKEY [857](#)
 - NONSMSVERSIONS [854](#)
 - OUTDATASET [858](#)
 - RESUME [855](#)
 - RETAINDAYSEXTRABACKUPS [857](#)

- EXPIREBV command (*continued*)
 - optional parameters (*continued*)
 - RETAINDAYSONLYBACKUP [857](#)
 - STARTKEY [855](#)
 - SYSOUT [858](#)
 - UNIXPATH [855–857](#)
 - required parameters
 - DISPLAY [852](#)
 - EXECUTE [852](#)
- expired
 - backup versions
 - non-SMS-managed data sets, deleting [310](#)
 - non-SMS-managed data sets, processing [310](#)
 - SMS-managed [236](#)
 - SMS-managed, deleting from backup volumes [233](#)
 - data sets
 - deleting, non-SMS-managed volumes [111](#)
 - deleting, SMS-managed storage [66](#)
 - specifying how to scratch [49](#)
 - dump copies, deleting [195](#)
- EXPIREDDATASETS parameter, SETSYS command [49](#)
- expiring ABARS activity records [354](#)
- Extended Format Data Set [997](#), [998](#)
- extent reduction
 - daily statistics report output [49](#)
 - determining
 - candidate data sets, non-SMS-managed volumes [112](#)
 - candidates for, SMS-managed storage [69](#)
 - non-SMS-managed data sets [117](#)
 - performing on non-SMS-managed volume [117](#)
 - specifying how to do [49](#)

F

- fast replication
 - catalogs, maintaining [289](#)
 - considerations
 - copy pool [288](#)
 - FRBACKUP [261](#)
 - FRRECOV [274](#)
 - copy pool
 - decreasing volumes [289](#)
 - increasing volumes [289](#)
 - recovery of [270](#)
 - setup [257](#)
 - using [257](#)
 - copy pool backup storage group
 - decreasing [290](#)
 - setup [259](#)
 - using [259](#)
 - creating [261](#)
 - data set recovery
 - performance considerations [277](#)
 - defining dump classes [277](#)
 - holding [283](#)
 - information, obtaining [284](#)
 - limiting all recovery requests [1150](#)
 - limiting recovery requests [281](#)
 - limiting tape recovery requests [1150](#)
 - parallelism
 - FRBACKUP [269](#)
 - FRRECOV [281](#)

- fast replication (*continued*)
 - preparing for [260](#)
 - recovering [270](#)
 - releasing [283](#)
 - selecting a copy method for recovery [1142](#)
 - selecting a FlashCopy relationship to establish [1142](#)
 - selecting recovery method [276](#)
 - serialization [262](#)
 - setup
 - copy pool backup storage group type [259](#)
 - copy pool construct [257](#)
 - suppressing volume paring messages (ARC1809I) [1142](#)
 - unneeded backup versions, deleting [283](#)
 - using
 - copy pool backup storage group type [259](#)
 - copy pool construct [257](#)
 - extend storage group type [260](#)
 - multiple copy pools [290](#)
 - multivolume data sets [291](#)
 - overflow storage group type [260](#)
 - withdrawing relationship [282](#)
- Fast Replication Backup
 - ALLOWPPRCP option [267](#)
 - using Metro Mirror Primary Volume [267](#)
- fast replication messages
 - message data sets [291](#)
- Fast Replication Recovery
 - ALLOWPPRCP option [279](#)
 - using Metro Mirror Primary Volume [279](#)
- fast subsequent migration
 - how migration cleanup affects [96](#)
 - performing [69](#)
 - recalling data sets [87](#)
 - reconnecting data sets [94](#)
 - specifying [94](#)
 - using ARCMDEXT [96](#)
- filtering, generation data sets [326](#)
- FILTEROUTPUTDATASET parameter, ABACKUP command [348](#)
- first qualifier for names of backup versions, specifying [189](#)
- first qualifier for names of migrated data sets, specifying [53](#)
- FIXCDS
 - backup volume cleanup process [219](#)
 - ensuring that CDSs are accurate [519](#)
- FIXCDS command
 - for DFSMSHsm [863](#)
- FlashCopy
 - FRRECOV
 - specifying Metro Mirror primary volume [886](#)
 - specifying XRC primary volume [887](#)
- FlashCopy consistency groups [269](#)
- FlashCopy relationship, withdrawing [282](#)
- FORCENONSMS parameter
 - RECALL command [92](#), [93](#), [125](#)
 - RECOVER command
 - from a dump copy [253](#)
 - with BDAM data set [248](#)
 - with extended sequential data set [248](#)
- format
 - of names of list data sets [101](#)
 - of names of utility data sets [100](#)
- forward recovery [295](#)
- FRBACKUP [261](#), [282](#), [283](#)
- FRBACKUP command

FRBACKUP command *(continued)*

examples [870](#)

for DFSMSHsm [865](#)

optional parameters

BACKUPSTORAGEGROUP [866](#)

DATE [865](#), [868](#)

DUMP [865](#), [866](#)

DUMPClass [865](#), [867](#)

DUMPEND [865](#), [866](#)

DUMPNLY [865](#), [867](#), [868](#)

EXECUTE [867](#)

FASTREPLICATIONEND [865](#), [866](#)

FORCE [865](#), [869](#)

GENERATION [865](#), [868](#)

MESSAGEDATASET [869](#)

NOVTOCENQ [870](#)

PREPARE [865](#), [867](#)

RETURNCONTROL [865](#), [866](#)

TERMINAL [865](#)

TOKEN [865](#), [868](#), [870](#)

VERSION [865](#), [868](#)

WITHDRAW [867](#)

required parameters

COPYPOOL [866](#)

FRDELETE command

examples [874](#)

for DFSMSHsm [873](#)

optional parameters

BOTH [873](#), [874](#)

DASDONLY [873](#), [874](#)

DUMPClass [873](#), [874](#)

DUMPNLY [873](#), [874](#)

required parameters

ALL [873](#)

COPYPOOL [873](#)

TOKEN [873](#)

VERSIONS [873](#)

FREEVOL

moving all data sets from level 1 [533](#)

moving backup volumes [535](#)

using on migration level 1 volumes [531](#)

FREEVOL command

examples [881](#)

optional parameters

AGE [879](#)

RETAINNEWESTVERSION [880](#)

TARGETLEVEL [879](#)

overview [877](#)

required parameters

BACKUPVOLUME [878](#)

MIGRATIONVOLUME [878](#)

ML1BACKUPVERSIONS [878](#)

syntax [878](#)

FREQUENCY

parameter

ALTERDS command [308](#)

BACKVOL command [310](#)

DEFINE command [172](#)

SETSYS command [303](#)

frequency of backup, non-SMS-managed, specifying [303](#)

frequency of backup, SMS-managed, specifying [180](#)

FROMDUMP

parameter

RECOVER command [252](#)

FROMVOLUME

parameter

BDELETE command [311](#)

RECOVER command [311](#)

recovering a data set [241](#)

recovering a VSAM data set [314](#)

FRRECOV

AVAILABLEFORMOUNT [822](#)

FlashCopy

specifying Metro Mirror primary volume [886](#)

specifying XRC primary volume [887](#)

FRRECOV command

examples [892](#)

for DFSMSHsm [883](#)

optional parameters

ALLOWXRCP [887](#)

APLOWPPRC [886](#)

DATE [888](#)

FASTREPLICATION [888](#)

FORCE [889](#)

FROMCOPYPOOL [889](#)

FROMDASD [889](#)

FROMDUMP [873](#), [889](#)

GENERATION [888](#)

MESSAGEDATASET [890](#)

NEWNAME [891](#)

NOCOPYPOOLBACKUP [891](#)

PARTIALOK [889](#)

REPLACE [891](#)

RESUME [889](#)

TOKEN [888](#)

VERIFY [892](#)

VERSION [888](#)

required parameters

COPYPOOL [885](#)

DSNAME [886](#)

TOVOLUME [886](#)

using [271](#)

FSR (function statistics record)

and ABARS [325](#)

recording ABARS CPU time and accounting codes [413](#)

FTP (NetView file transfer program) [427](#)

full journal, damaged CDS [629](#)

full journal, undamaged CDS [632](#)

full tapes, converting partial tapes to [440](#)

full-volume dump

automatic, example system [165](#)

invocation compress parameter [457](#)

procedures [489](#)

full-volume restore

combined with volume recovery [295](#)

using APPLYINCREMENTAL subparameter [295](#)

FULLDSNAME parameter

RECYCLE command [540](#)

function

availability management [7](#)

backup

non-SMS-managed

[15](#)

SMS-managed [11](#)

dump

non-SMS-managed

[14](#)

SMS-managed [11](#)

function (*continued*)

- enhanced CDS recovery
 - implementing using JCL [508](#)
 - requirements for use [508](#)
- non-SMS-managed volumes, DFSMSHsm [14](#)
- overview of DFSMSHsm [9](#)
- space management
 - non-SMS-managed [16](#)
 - SMS-managed [13](#)

G

GDG

- management attributes, specifying [40](#)

GDS management class, space management [41](#)

generation data sets

- and the ARECOVER TGTGDS command [387](#)
- and the SETSYS ARECOVERTGTGDS command [387](#)
- deferred [149](#), [460](#)
- filtering [326](#)
- processing [387](#)

GENERATION parameter, RECOVER command [241](#)

generic data set profile [617](#)

glossary [1419](#)

grouping data sets

- with management classes [11](#)
- with storage groups [11](#)

GUARANTEED BACKUP FREQUENCY attribute [180](#)

GUARANTEED SPACE attribute and VSAM data sets, availability management [451](#)

GUARANTEED SPACE attribute, space management [152](#)

GUARANTEED SPACE VSAM data sets

recalling

- changes in storage-class-to-data set mapping [153](#)
- changes in the storage group [153](#)
- problems [153](#)
- problems with conversion of data sets [154](#)

recovering

- changes in storage-class-to-data set mapping [452](#)
- changes in the storage group [451](#)
- problems with conversion of data sets [452](#)

H

HALTERDS command, security checking [309](#)

HARDCOPY parameter, RELEASE command [602](#)

hbackup command

- for DFSMSHsm [1247](#)

HBDELETE processing, authorization checking [240](#)

health checker for DFSMSHsm [627](#)

HFS data set

- eligibility to migrate to an SDSP [76](#)
- support during RECOVER [253](#)

HOLD

optional parameters

- ABACKUP [597](#)
- ALL [596](#)
- ARECOVER [597](#)
- AUDIT [597](#)
- BACKUP [597](#)
- COMMONQUEUE [131](#), [597](#)

HOLD (*continued*)

optional parameters (*continued*)

- DUMP [597](#)
- EOD [408](#)
- EXPIREBV [598](#)
- FRBACKUP [283](#)
- FRRECOV [283](#)
- LIST [598](#)
- LOG [598](#)
- MIGRATION [599](#)
- RECALL [599](#)
- RECOVER [599](#)
- RECYCLE [600](#)
- REPORT [600](#)
- TAPE [599](#)
- TAPEDATASET [599](#)
- TSO [599](#)

preventing

- backup processing [449](#)
- dump processing [456](#)
- recovery processing [459](#)
- restore processing [459](#)

HOLD command

examples [909](#)

optional parameters

- ABACKUP [897](#)
- ALL [898](#)
- ARECOVER [898](#)
- AUDIT [899](#)
- AUTOEXPIRATION [899](#)
- AUTOMIGRATION [900](#)
- BACKUP [900](#)
- BACKUP(DSCOMMAND) [900](#)
- COMMONQUEUE [901](#)
- DUMP [902](#)
- ENDOFDATASET [903](#), [904](#)
- ENDOFVOLUME [904](#)
- EXPIREBV [905](#)
- FRBACKUP [902](#), [905](#)
- FRRECOV [905](#)
- LIST [907](#)
- LOG [907](#)
- MIGRATION [907](#)
- RECALL [907](#)
- RECOVER [908](#)
- RECYCLE [908](#)
- REPORT [909](#)
- TAPECOPY [909](#)
- TAPEREP [909](#)

overview [895](#)

syntax [896](#)

hold-volumes date description [122](#)

holding ABARS functions [408](#)

HOST

parameter

- LIST command [565](#)
- startup procedure [595](#)

host modes [593](#)

how to obtain

information about

- backup volumes [560](#)
- conditions in the system when an error occurs [586](#)
- data sets [560](#)
- DFSMSHsm address space [570](#)

- how to obtain (*continued*)
 - information about (*continued*)
 - DFSMSHsm operations [574](#)
 - DFSMSHsm-managed volumes [568](#)
 - dump classes [564](#)
 - dump volumes [564](#)
 - migration and DFSMSHsm-managed volumes [567](#)
 - processor serialization [565](#)
 - tape volumes [569](#)
 - user authorization [570](#)
 - information from
 - DFSMSHsm [559](#)
 - the control data sets [559](#)
- HSEND CMD command
 - examples [915](#)
 - for DFSMSHsm [913](#)
 - optional parameters
 - NOWAIT [914](#)
 - WAIT [914](#)
 - required parameters
 - command [913](#)
- HSENDU command
 - for DFSMSHsm [917](#)
- HSMABR
 - aggregate backup RACF [340](#)
 - RACF tape volume set [340](#)
- HSMEDIT procedure, printing DFSMSHsm log information [608](#)
- HSMLOG procedure, printing DFSMSHsm log information [608](#)
- HWCOMPRESS parameter
 - DEFINE command [176](#)

I

- ICF catalog, ABARS processing [356](#), [410](#)
- ICFRU Forward Recovery Utility, procedure for using [298](#)
- ICHRIN03, RACF started-procedures table, printing DFSMSHsm log information [608](#)
- IDCAMS
 - data set [373](#)
 - EXPORT, use of, for CDS backup [494](#)
- identify expired backup versions, EXPIREBV processing, SMS-managed [236](#)
- identifying migration level 2 volumes to DFSMSHsm [55](#)
- IDRC [1190](#)
- IGG.CATLOCK FACILITY class profile [488](#)
- implementing enhanced CDS recovery using JCL [508](#)
- import utility ARCIMPRT [508](#)
- Improved Data Recording Capability (IDRC) [1190](#)
- INCLUDE
 - data set [323](#)
 - keyword [332](#)
 - list, processing migrated data sets in [389](#)
- incremental backup, automatic, example system [165](#)
- INCREMENTAL parameter, BACKVOL command [232](#)
- incremental volume recovery [300](#)
- INCREMENTALBACKUP parameter, SETSYS command [190](#)
- INDATASET parameter, TAPECOPY command [440](#)
- initial selection
 - DASD daily backup volume [209](#)
 - DASD or tape daily backup volume [213](#)
 - spill backup volume [217](#)
 - tape daily backup volume [211](#)
- inline backup, how to use [447](#)

- installation exits
 - aggregate backup [342](#)
 - aggregate recovery [387](#), [398](#)
- instruction data set
 - creating [334](#)
 - example [335](#), [380](#)
 - overview [323](#)
- INSTRUCTION parameter, ARECOVER command [393](#)
- instruction/activity log file
 - aggregate backup output [352](#)
 - description [325](#)
 - input for recovery [373](#)
- integrated catalog facility catalog
 - backing up, procedure [487](#)
 - recovering, procedure [487](#)
 - recovery [314](#)
 - restoring from a dump copy, procedure [489](#)
- interrupting
 - ABACKUP command processing [597](#)
 - all DFSMSHsm processing [596](#)
 - and restarting
 - automatic dump processing [489](#)
 - backup processing [485](#)
 - primary space management [161](#)
 - secondary space management [161](#)
 - ARECOVER command processing [597](#)
 - AUDIT command processing [597](#)
 - backup processing [597](#)
 - DFSMSHsm functions [596](#)
 - dump processing [597](#)
 - EXPIREBV command processing [598](#)
 - LIST command processing [598](#)
 - long-running commands [680](#)
 - migration processing [599](#)
 - recycle processing [600](#)
 - REPORT command processing [600](#)
- interval migration
 - concept [25](#)
 - disabling [25](#)
 - enabling [25](#)
 - non-SMS-managed volumes [119](#)
 - notes, usage [26](#)
 - other considerations [26](#)
 - requirements for [25](#)
 - scenario [81](#)
 - SMS-managed volumes [25](#), [27](#)
- INTERVALMIGRATION parameter [48](#)
- introduction [7](#)
- ISMF (Interactive Storage Management Facility)
 - aggregate group name, panel parameter [336](#)
 - copies, panel parameter [336](#)
 - description, panel parameter [336](#)
 - example of aggregate group panel [337](#)
 - output data set prefix, panel parameter [336](#)
 - using
 - for aggregate group backup [413](#)
 - for aggregate group recovery [418](#)
 - for aggregate groups [336](#)
 - for selection data sets [329](#)

J

- java hbackup command

java hbackup command (*continued*)

 syntax [1247](#)

Java hbackup command

 environment configuration [1250](#)

 examples [1251](#)

 for Java commands [1247](#)

 optional parameters [1247](#)

 required parameters

 directory_name [1247](#)

JES2

 pools in systems running under [105](#)

JES3

 considerations, DELVOL command [537](#)

 deletion of migrated data sets fails, non-SMS-managed [122](#)

 DFSMSHsm commands that affect pool configuration [106](#)

 DFSMSHsm migration window [117](#)

 directed recall request [117](#)

 example of pooling [107](#)

 migration considerations, non-SMS-managed volumes [116](#)

 parameter, SETSYS command [106](#)

 pools in systems running under [106](#)

 prevent migration request [117](#)

 recall considerations, non-SMS-managed data sets [102](#), [121](#)

 specifying pools [106](#)

 volume pooling [107](#)

journal copy utility ARCBJRNL [508](#)

journal damaged, CDS undamaged [632](#)

journal data set

 DASD backup data sets [499](#)

 disabled [522](#)

 moving [518](#)

 multiple

 backup data sets [497](#)

 tape backup data sets [501](#)

 non-intrusive backup [494](#), [496](#), [507](#), [525](#)

 nulling [503](#)

 quiesced backup [496](#)

 recovering, examples [519](#)

 replacing, using multiple DASD backup data sets [522](#)

journal full, undamaged CDS [632](#)

journal I/O error, recovering from [518](#)

JOURNAL parameter, UPDATEC command [516](#)

journal records missing, damaged CDS [633](#)

journal records missing, time sequence gaps in CDS records [633](#)

JRNBACKUPDSN subparameter, CDSVERSIONBACKUP parameter, SETSYS command [497](#)

K

key range volumes, ML2, making available [551](#)

keyboard

 navigation [1413](#)

 PF keys [1413](#)

 shortcut keys [1413](#)

L

LOVOLS parameter, DEFINE command [383](#)

large-scale cyber event [650](#)

last backup date, resetting [203](#)

last-backup date, resetting, recovery processing [249](#)

last-processed key [234](#)

LBACKUP dates in catalog [433](#)

level 0 volumes

 moving all data sets to other level 0 volumes [527](#)

LEVEL 1 DAYS NON-USAGE attribute [37](#)

LEVEL 2 DAYS NON-USAGE attribute [38](#)

level-1-to-level-2 migration

 command [530](#)

 example [64](#)

 exits called [63](#)

 process [61](#)

 specifying when non-SMS-managed data sets become eligible [101](#)

 to a DASD level 2 volume [62](#)

 to a tape level 2 volume [62](#)

limit

 daily

 not reached, DASD or tape daily backup volume selection [214](#)

 reached, DASD or tape daily backup volume selection [213](#)

 on size of DFSMSHsm DASD copies [9](#)

LIMIT parameter, RECYCLE command [539](#)

LIST

 optional parameters

 OUTDATASET [559](#)

 SELECT [560](#)

 SYSOUT [559](#)

 TERMINAL [559](#)

 required parameters

 BACKUPVOLUME [560](#)

 DATASETNAME [560](#)

 DUMPCCLASS [564](#)

 DUMPVOLUME [564](#)

 HOST [565](#)

list and utility data sets, deleting, non-SMS-managed volumes [111](#)

LIST command

 examples [947](#)

 for DFSMSHsm [1341](#)

 optional parameters

 ALLCOMPLETE [940](#)

 ALLDUMPS [931](#)

 ALLVERS [932](#)

 ALLVOLS [932](#)

 BACKUPCONTENTS [931](#)

 BACKUPCONTROLDATASET [933](#)

 BOTH [933](#)

 DATASETINFORMATION [933](#)

 DATASETS [932](#)

 DATE [933](#)

 DUMPCONTENTS [934](#)

 DUMPVOLS [932](#)

 FAILED [940](#)

 FASTREPLICATIONVOLS [932](#)

 FCFRRINCOMPLETE [940](#)

 GEN [932](#)

 GENERATION [932](#)

 INCLUDEPRIMARY [932](#), [934](#)

 MIGRATIONCONTROLDATASET [933](#)

 NODATASETINFORMATION [933](#)

LIST command (*continued*)

optional parameters (*continued*)

NONE [940](#)
NONRECOVERABLE [940](#)
NOVOLS [932](#)
OUTDATASET [934](#)
PARTIAL [940](#)
RECOVERABLE [940](#)
REQUIREDCOMPLETE [940](#)
RESET [935](#)
SELECT [936](#)
SELECT ACTIVE [938](#)
SELECT AGE [938](#)
SELECT ALTERNATEVOLUME [938](#)
SELECT ASSOCIATED [941](#)
SELECT AVAILABLE [939](#)
SELECT BACKUP [944](#)
SELECT BOTH [944](#)
SELECT CAPACITYMODE [939](#)
SELECT CONNECTED [939](#)
SELECT DISASTERALTERNATEVOLUMES [940](#)
SELECT DUMPCLASS [940](#)
SELECT DUMPSTATE [940](#)
SELECT EEFMT2 [941](#)
SELECT EEFMT3 [941](#)
SELECT EEFMT4 [941](#)
SELECT EFMT1 [941](#)
SELECT EFMT2 [941](#)
SELECT EFMT3 [941](#)
SELECT EFMT4 [941](#)
SELECT EMPTY [941](#)
SELECT ENCRYPTED [941](#)
SELECT EXCESSIVEVOLUMES [942](#)
SELECT EXPIRED [942](#)
SELECT FAILEDCREATE [943](#)
SELECT FAILEDRECYCLE [943](#)
SELECT FASTREPLICATIONSTATE [940](#)
SELECT FULL [941](#)
SELECT INCOMPLETERECOVERY [940](#)
SELECT LIBRARY [943](#)
SELECT MIGRATIONLEVEL1 [944](#)
SELECT MIGRATIONLEVEL2 [944](#)
SELECT MULTIPLEVOLUME [944](#)
SELECT NOALTERNATEVOLUME [938](#)
SELECT NOLIBRARY [943](#)
SELECT NOOVERFLOW [945](#)
SELECT NORETENTIONLIMIT [945](#)
SELECT NOSMALLDATASETPACKING [946](#)
SELECT NOTASSOCIATED [941](#)
SELECT NOTCONNECTED [939](#)
SELECT NOTENCRYPTED [941](#)
SELECT NOTFULL [941](#)
SELECT OVERFLOW [945](#)
SELECT RECALLTAKEAWAY [945](#)
SELECT RECYCLETAKEAWAY [941](#)
SELECT RESUMEAUDIT [945](#)
SELECT RETAINDAYS [938](#)
SELECT SMALLDATASETPACKING [946](#)
SELECT UNAVAILABLE [946](#)
SELECT UNEXPIRED [946](#)
SELECT VOLUME [944](#)
SELECT VSAM [947](#)
SUMMARY [947](#)
SYSOUT [934](#)

LIST command (*continued*)

optional parameters (*continued*)

TERMINAL [934](#)
TOKEN [932](#)
VERSION [933](#)

output

LIST AGGREGATE [1341](#)
LIST BACKUPVOLUMES [1345](#)
LIST COPYPOOL [1348](#)
LIST COPYPOOLBACKUPSTORAGEGROUP [1351](#)
LIST DATASETNAME [1352](#)
LIST DUMPCLASS [1361](#)
LIST DUMPVOLUME [1364](#)
LIST HOST [1381](#)
LIST LEVEL [1352](#)
LIST MIGRATIONVOLUME [1371](#)
LIST PRIMARYVOLUME [1371](#)
LIST TAPETABLEOFCONTENTS [1381](#)
LIST USER [1386](#)
LIST VOLUME [1371](#)

required parameters

AGGREGATE [925](#)
BACKUPVOLUME [925](#)
CLOUD [925](#)
COPYPOOL [926](#)
CPBSG [926](#)
DATASETNAME [926](#), [928](#)
DUMPCLASS [927](#)
DUMPVOLUME [927](#)
HOST [928](#)
LEVEL [926](#), [928](#)
MIGRATIONLEVEL1 [929](#)
MIGRATIONLEVEL2 [929](#)
MIGRATIONVOLUME [929](#)
PRIMARYVOLUME [929](#)
TTOC [930](#)
USER [931](#)
VOLUME [929](#)

syntax [920](#)

LIST COPYPOOL(cpname)

FCFRR [278](#)

list data sets

format of names [101](#)

specifying age for deleting [100](#)

LIST parameter

HOLD command [598](#)

RELEASE command [602](#)

LISTOFNAMES [730](#)

LOG command

examples [953](#)

for DFSMSHsm [953](#)

required parameter

data [953](#)

LOG parameter

HOLD command [598](#)

RELEASE command [602](#)

log, DFSMSHsm, printing [607](#)

LOGY data set, printing DFSMSHsm log information [608](#)

M

macro invocation of DFSMSHsm functions [558](#)

main host [593](#)

maintaining

- maintaining (*continued*)
 - control data sets [493](#)
 - password protection for recovered data sets [463](#)
- maintenance and use of the data set date-last-referenced field for space management [78](#)
- management classes
 - aggregate backup overview [322](#)
 - attributes
 - # GDG ELEMENTS ON PRIMARY [40](#)
 - ADMIN OR USER COMMAND BACKUP [185](#)
 - AUTO BACKUP [186](#)
 - BACKUP COPY TECHNIQUE [495](#)
 - BACKUP FREQUENCY [180](#)
 - COMMAND OR AUTO MIGRATE [37](#)
 - EXPIRE AFTER DATE/DAYS [36](#)
 - EXPIRE AFTER DAYS NON-USAGE [36](#)
 - LEVEL 1 DAYS NON-USAGE [37](#)
 - LEVEL 2 DAYS NON-USAGE [38](#)
 - NUMBER OF BACKUP VERSIONS [181](#)
 - PARTIAL RELEASE [36](#)
 - PRIMARY DAYS NON-USAGE [37](#)
 - RETAIN DAYS EXTRA BACKUP VERSIONS [181](#)
 - RETAIN DAYS ONLY BACKUP VERSION [181](#)
 - ROLLED-OFF GDS ACTION [40](#)
 - considerations [338](#)
 - default [40](#)
 - example
 - backup [186](#)
 - space management [41](#)
 - grouping data sets with [11](#)
 - large, space management [42](#)
 - retrieval of
 - availability management [464](#)
 - space management [154](#)
 - support for CDS backup [495](#)
 - use in backup [178](#)
 - use of [11](#)
- managing
 - DFSMSHsm media [527](#)
 - SDSP data sets [555](#)
 - storage with DFSMSHsm, tasks for [8](#)
- manual restarts
 - of automatic primary space management [162](#), [163](#)
 - of automatic secondary space management [163](#)
- MASH configurations [526](#)
- MAXABARSADDRESSSPACE parameter, SETSYS command [343](#), [384](#)
- MAXBACKUPTASKS parameter, SETSYS command [190](#)
- MAXDUMPTASKS parameter, SETSYS command [169](#)
- MAXEXTENTS parameter, SETSYS command [49](#)
- maximum number of
 - backup tasks in each processor, specifying [190](#)
 - backup versions to keep, specifying [181](#)
 - dump tasks [169](#)
- MAXINTERVALTASKS parameter, SETSYS command [51](#)
- MAXMIGRATIONTASKS parameter, SETSYS command [49](#)
- MAXRECALLTASKS parameter, SETSYS command [52](#)
- MAXRECYCLETASKS parameter, SETSYS command [538](#)
- MAXSSMTASKS parameter, SETSYS command [50](#)
- MCDSBACKUPDSN subparameter, CDSVERSIONBACKUP parameter, SETSYS command [497](#)
- media management [527](#)
- MENTITY parameter, ARECOVER command [387](#), [394](#)
- message data set

- message data set (*continued*)
 - allocation [292](#)
 - contents [293](#)
 - example [293](#)
- messages
 - ARC0026E [629](#), [632](#), [633](#)
 - ARC0734I [637](#)
 - ARC0909E [632](#)
 - ARC1102I [647](#)
 - ARC1187E [647](#), [648](#)
 - ARC1502I [648](#)
 - ARC1506E [647](#), [648](#)
- messages associated with QUERY [1389](#)
- messages, selecting for ABARS activity logs [344](#)
- Metro Mirror primary volume
 - FlashCopy operation
 - FRRECOV [886](#)
- Metro Mirror Primary Volume, using
 - Fast Replication Backup [267](#)
 - Fast Replication Recovery [279](#)
- migrate
 - data sets, SMS-managed storage [72](#)
 - to an SDSP, size eligibility for data sets [76](#)
- MIGRATE
 - command for SMS-managed data sets [89](#), [90](#)
 - deleting non-SMS-managed data sets from migration volumes [532](#)
 - parameters
 - CONVERT, non-SMS-managed data sets [528](#)
 - CONVERT, SMS-managed data sets [527](#)
 - DAYS [532](#)
 - DELETEBYAGE [532](#)
 - DELETEIFBACKEDUP [532](#)
 - MIGRATIONLEVEL1 [532](#)
 - VOLUME [532](#)
 - VOLUME, SMS-managed data sets [91](#)
 - using for migration level 1 volumes [532](#)
- migrate candidates for
 - both migration and extent reduction, SMS-managed storage [71](#)
 - extent reduction, SMS-managed storage [71](#)
- MIGRATE command
 - examples [964](#), [971](#)
 - for DFSMSHsm [955](#)
- MIGRATE command for SMS volumes and data sets
 - optional parameters
 - CONVERT [960](#)
 - DAYS [961](#)
 - KEEPMIGDATE [961](#)
 - MIGRATIONLEVEL1 [962](#)
 - MIGRATIONLEVEL2 [962](#)
 - MOVE [962](#)
 - TERMINAL [963](#)
 - required parameters
 - DATASETNAME [958](#)
 - MIGRATIONLEVEL1 [958](#)
 - STORAGEGROUP [959](#)
 - VOLUME [958](#)
- MIGRATE command, non-SMS
 - optional parameters
 - CONVERT [968](#)
 - DAYS [969](#)
 - MIGRATIONLEVEL1 [970](#)

- MIGRATE command, non-SMS (*continued*)
 - optional parameters (*continued*)
 - MIGRATIONLEVEL2 [970](#)
 - TERMINAL [970](#)
 - UNIT [970](#)
 - required parameters
 - DATASETNAME [966](#)
 - MIGRATIONLEVEL1 [966](#)
 - PRIMARY [966](#)
 - VOLUME [967](#)
- MIGRATE command, SMS
 - optional parameters
 - CLOUD [960](#)
- MIGRATE subparameter, ADDVOL command [103](#)
- MIGRATE SYSTEM/SYSPLEX NAME attribute [35](#)
- migrated data sets
 - backing up [200](#)
 - deleting [88](#)
 - GDG FORCE, caution with [89](#)
 - names [73](#)
 - profile backup [207](#)
 - recovering [551](#)
 - results with different commands [88](#)
 - uncataloging [625](#)
- migrated data sets that got deleted [646](#)
- MIGRATEDDATA parameter, ARECOVER command [395](#)
- MIGRATEPREFIX parameter [53](#)
- MIGRATEPSMEARLY
 - parameter [58](#)
- migrating
 - all data sets from migration level 1 volume, processing for non-SMS-managed data sets [534](#)
 - all data sets from migration level 1 volume, processing for SMS-managed data sets [533](#)
 - data sets [532](#)
 - data sets to migration level 2, specifying how [47](#)
- migration
 - and recall from DASD, RACF considerations [624](#)
 - attributes, specifying [37](#)
 - automatic interval space management tasks, specifying [51](#)
 - automatic primary space management tasks, specifying [49](#)
 - cleanup
 - example [60](#)
 - process [59](#)
 - specifying how to do [54](#)
 - command, of a data set [89](#)
 - control data set
 - recovering, using multiple DASD backup data sets [519](#)
 - recovering, using multiple tape backup data sets [524](#)
 - recovery and multiple backup data sets, using UPDATEC [513](#)
 - controlling, for data sets and volumes, non-SMS-managed [124](#)
 - determining candidate data sets, non-SMS-managed volumes [112](#)
 - from SDSP data sets [62](#)
 - interval
 - disabling [25](#)
 - enabling [25](#)
 - notes, usage [26](#)
 - migration (*continued*)
 - interval (*continued*)
 - other considerations [26](#)
 - requirements for [25](#)
 - scenario [81](#)
 - interval, non-SMS-managed volumes [119](#)
 - JES3 considerations, non-SMS-managed volumes [116](#)
 - level 1 volumes
 - damaged, recovering backup versions from [554](#)
 - example [19](#)
 - moving all data sets [533](#)
 - using the MIGRATE command [532](#)
 - level 2
 - key range volumes, making available [551](#)
 - specifying desired number of partial tapes [141](#)
 - minimum age [112](#)
 - name of
 - cartridge-type single-file data sets [74](#)
 - dump copy data sets [197](#)
 - non-SMS-managed volumes [115](#)
 - on-demand
 - disabling [28](#)
 - enabling [28](#)
 - multiple DFSMSHsm host environment, in a [29](#)
 - notes, usage [29](#)
 - other considerations [29](#)
 - requirements for [28](#)
 - scenario [81](#)
 - out-of-space condition [77](#)
 - preventing processing of data sets [149](#)
 - processing, determining which tape ML2 volumes are available for output [574](#)
 - RACF-protected data sets [150](#)
 - reconnecting data sets [95](#)
 - SMS-managed volumes [91](#)
 - storage group [92](#)
 - tape volume contention [136](#)
 - to a DASD ML2 volume, level 1 to level 2 migration [62](#)
 - to a tape ML2 volume, level 1 to level 2 migration [62](#)
 - to both DASD and tape for migration level 2 [73](#)
 - using the FREEVOL command [533](#)
 - volume contention
 - single-processing-unit environment [137](#)
 - volumes
 - deleting non-SMS-managed data sets with the MIGRATE command [532](#)
 - obtaining information about [567](#)
 - recovering [552](#)
 - with thresholds of occupancy, non-SMS-managed volumes [115](#)
 - without thresholds of occupancy, non-SMS-managed volumes [115](#)
- migration level one volume, damaged [641](#)
- MIGRATION parameter
 - HOLD command [599](#)
 - HOLD command, preventing processing of data sets [149](#)
 - MIGRATIONLEVEL2 subparameter [55](#)
 - MIGRATIONVOLUME parameter, FREEVOL command [533](#)
 - ML1BACKUPVERSIONS parameter, FREEVOL command [531](#)
 - RELEASE command [602](#)
- migration tape, overwritten [634](#)
- MIGRATIONAUTOCLOUD

- MIGRATIONAUTOCLOUD (*continued*)
 - parameter [58](#)
- MIGRATIONCLEANUPDAYS parameter, SETSYS command [54](#)
- MIGRATIONCONTROLDATASET parameter, UPDATEC command [516](#)
- MIGRATIONLEVEL1 parameter, MIGRATE command [532](#)
- MIGRATIONLEVEL1DAYS parameter, SETSYS command [101](#)
- MIGRATIONLEVEL2 parameter, TAPECOPY command [438](#)
- minimum migration age [112](#)
- missing journal records, damaged CDS, [633](#)
- ML1 subparameter, MIGRATEDDATA parameter, ARECOVER command [395](#)
- ML1 volume, damaged [641](#)
- ML1OVERFLOW
 - SETSYS parameter [1160](#)
- ML1VOLS parameter, DEFINE command [383](#)
- ML2 parameter, RECYCLE command [539](#)
- ML2 subparameter, MIGRATEDDATA parameter, ARECOVER command [395](#)
- ML2PARTIALSNOTASSOCIATEDGOAL parameter, SETSYS command [141](#)
- ML2RECYCLEPERCENT parameter, SETSYS command [538](#)
- ML2TAPE subparameter, TAPEMIGRATION parameter, SETSYS command [48](#)
- model entity (MENTITY) parameter, ARECOVER command [394](#)
- model entity, defining [387](#)
- modeling, profile, effects on profile recovery [462](#)
- MOVE parameter, ABACKUP command [348](#)
- moving
 - data sets [93](#)
- MSG 900 [1337](#)
- MSG 901 [1337](#)
- MSG 902 [1337](#)
- MSG 903 [1337](#)
- MSG 904 [1337](#)
- MSG 905 [1337](#)
- MSG 906 [1337](#)
- MSG 907 [1337](#)
- MSG 908 [1337](#)
- MSG 909 [1337](#)
- MSG 910 [1337](#)
- MSG 911 [1338](#)
- MSG 912 [1338](#)
- MSG 913 [1338](#)
- MSG 914 [1338](#)
- MSG 915 [1338](#)
- MSG 916 [1338](#)
- MSG 917 [1338](#)
- MSG 919 [1338](#)
- MSG 920 [1338](#)
- MSG 922 [1338](#)
- MSG 924 [1339](#)
- MSG 925 [1339](#)
- MSG 998 [1339](#)
- MSG 999 [1339](#)
- multicenter CDS considerations [502](#)
- multicenter CDS support [1112](#), [1113](#)
- multiple backup data sets
 - backing up the control data sets [497](#)
 - changing
 - from DASD to tape [504](#)
 - from tape to DASD [503](#)
 - choosing

- multiple backup data sets (*continued*)
 - choosing (*continued*)
 - DASD [498](#)
 - tape [498](#)
 - DASD, example
 - recovering the MCDS [519](#)
 - replacing the journal [522](#)
 - recovering control data sets and journal data set [519](#)
 - tape, example of recovering the MCDS [524](#)
- multiple-DFSMSHsm-host environment
 - ADDVOL [697](#)
 - AUDIT [1340](#)
 - LIST [928](#), [935](#)
 - SETSYS [1109](#), [1128](#), [1129](#)
- multiple-file tapes, making into cartridge-type single-file-format tapes [542](#)
- multitasking considerations for SDSP data sets [147](#)

N

- names of
 - backup copy data sets [206](#)
 - backup versions of data sets [204](#)
 - backup VTOC copy data sets [202](#)
 - dump copy data sets [197](#)
 - dump VTOC copy data sets [198](#)
 - list data sets, format [101](#)
 - migrated data sets [73](#)
 - migrated data sets, specifying first qualifier [53](#)
 - utility data sets, format [100](#)
- naming conventions
 - data sets [686](#)
- navigation
 - keyboard [1413](#)
- NEWNAME parameter, RECOVER command [241](#), [246](#), [252](#)
- no-migration management class, space management [42](#)
- NOBACKUPMIGRATED parameter, ARECOVER command [395](#)
- NOFIX parameter, AUDIT command [517](#)
- NOMIGRATION parameter, SETMIG command, preventing processing of data sets [149](#)
- non-intrusive journal backup, using [496](#), [507](#)
- non-SMS-managed
 - data sets
 - deleting expired backup versions [310](#)
 - extent reduction [117](#)
 - processing, migrating all data sets from ML1 volume [534](#)
 - recall [119](#)
- storage
 - automatic space management processing [109](#)
 - availability management [303](#)
 - command availability management [308](#)
 - command space management [123](#)
 - space management [99](#)
 - subtasks for availability management [303](#)
 - subtasks for space management [99](#)
- volumes
 - automatic space management [306](#)
 - backing up all primary [309](#)
 - command volume space management [124](#)
 - data set deletion [114](#)
 - data set retirement [114](#)
 - deleting expired data sets [111](#)
 - deleting list and utility data sets [111](#)

- non-SMS-managed (*continued*)
 - volumes (*continued*)
 - DFSMSHsm functions for [14](#)
 - interval migration [119](#)
 - migration [115](#)
 - migration with thresholds of occupancy [115](#)
 - migration without thresholds of occupancy [115](#)
 - specifying recall characteristics [101](#)
- non-VSAM data set
 - recovering from a DFSMSHsm backup version [245](#)
 - restoring from a dump copy [247](#)
- NONE subparameter, TAPEMIGRATION parameter, SETSYS command [48](#)
- NONE(ROUTETOTAPE) subparameter, TAPEMIGRATION parameter, SETSYS command [48](#)
- NONSMSVERSIONS parameter, EXPIREBV command [310](#)
- NOOVERFLOW
 - ADDVOL parameter [707](#)
- NOSPILL parameter, SETSYS command, cleanup processing [219](#)
- NOSTACK parameter
 - ABACKUP command [348](#)
 - ARECOVER DATASETNAME command [394](#)
 - SETSYS ABARSTAPES command [344](#), [385](#)
- nulling the journal [503](#)
- NUMBER OF BACKUP VERSIONS
 - attributes
 - relationship to RETAIN DAYS EXTRA BACKUP VERSIONS [182](#)
 - relationship to RETAIN DAYS ONLY BACKUP VERSIONS [182](#)

O

- objective in using DFSMSHsm [10](#)
- OCDS TTOC record handling [550](#)
- offline control data set
 - recovery and multiple backup data sets, using UPDATEC [513](#)
- OFFLINECONTROLDATASET parameter, UPDATEC command [516](#)
- on-demand migration
 - concept [27](#)
 - disabling [28](#)
 - enabling [28](#)
 - multiple DFSMSHsm host environment, in a [29](#)
 - notes, usage [29](#)
 - other considerations [29](#)
 - requirements for [28](#)
 - scenario [81](#)
- ONLYIF command
 - examples [975](#)
 - optional parameters
 - BEGIN [975](#)
 - END [975](#)
 - overview [973](#)
 - required parameters
 - HSMHOST [974](#)
 - HMSYM [974](#)
 - SYSNAME [974](#)
 - syntax [973](#)
- operating procedures for DFSMSHsm [593](#)
- OPTIMIZE parameter, ABACKUP command [344](#), [347](#), [348](#)
- optimum

- optimum (*continued*)
 - DASD blocking option [385](#)
- order of data set migration
 - automatic space management, non-SMS-managed [113](#)
 - automatic space management, SMS-managed [72](#)
 - interval migration [25](#), [27](#)
- orderly shutdown [596](#)
- organization
 - of this manual [8](#)
- ORIGINAL subparameter, INCREMENTALBACKUP parameter, SETSYS command [190](#)
- ORIGINALVOLUMES parameter
 - TAPECOPY command [439](#)
 - TAPEREPL command [440](#)
- OUTDATASET parameter
 - LIST command [559](#)
 - RECYCLE command [540](#)
- output data set prefix, ISMF panel parameter [336](#)
- output from REPORT DAILY [1401](#)
- output from REPORT VOLUME [1405](#)
- OVERFLOW
 - ADDVOL parameter [707](#)
- overflow volumes
 - migration [47](#)
- overview
 - backup and recovery functions
 - non-SMS-managed [15](#)
 - SMS-managed [11](#)
 - dump and restore functions
 - non-SMS-managed [14](#)
 - SMS-managed [11](#)
 - of DFSMSHsm functions [9](#)
 - space management functions
 - non-SMS-managed [16](#)
 - SMS-managed [13](#)
- overwritten
 - backup tape [637](#)
 - migration tape [634](#)
- owned dumping volume [455](#)

P

- parameter
 - ABARS [1389](#)
 - ACTIVE [1389](#)
 - ARPOOL [1391](#)
 - AUTOPROGRESS [1391](#)
 - BACKUP [1391](#)
 - CDSVERSIONBACKUP [1392](#)
 - COMMONQUEUE [1392](#)
 - CONTROLDATASETS [1392](#)
 - COPYPOOL [1392](#)
 - CSALIMITS [1392](#)
 - DATASETNAME [1392](#)
 - IMAGE [1393](#)
 - MIGRATIONLEVEL2 [1393](#)
 - NETWORK [1393](#)
 - POOL [1394](#)
 - REQUEST [1394](#)
 - RETAIN [1394](#)
 - SECURITY [1394](#)
 - SETSYS [1394](#)
 - SPACE [1397](#)

parameter (*continued*)

STARTUP [1397](#)
STATISTICS [1397](#)
TRAPS [1398](#)
USER [1398](#)
VOLUMEPOOL [1398](#)
WAITING [1398](#)

parameters

* [786](#), [794](#), [813](#), [1000](#), [1001](#)
ABACKUP [896](#), [897](#), [1031](#)
ABARS [982](#), [983](#)
ABARSACTLOGMSGVLV [1094](#)
ABARSACTLOGTYPE [1094](#)
ABARSBUFFERS [1095](#)
ABARSCONTROLS [743](#)
ABARSDELETEACTIVITY [1095](#)
ABARSKIP [1095](#)
ABARSOPTIMIZE [1096](#)
ABARSPROCNAME [1096](#)
ABARSTAPES [1096](#)
ABARSUNITNAME [1097](#)
ABARSVERSIONS [853](#)
ABARSVOLCOUNT [1097](#)
absolutepath [855](#)
ACCEPTPSCBUSERID [1098](#)
ACTIVE [938](#), [982](#), [983](#), [1126](#)
ACTIVITY [724](#), [727](#)
ACTLOGMSGVLV [1098](#)
ACTLOGTYPE [1099](#)
AGE [878](#), [879](#), [938](#)
AGGREGATE [724](#), [725](#), [925](#), [1030](#), [1031](#)
AGNAME [852](#), [853](#)
ALL [743](#), [752](#), [813](#), [873](#), [898](#), [982](#), [985](#), [1031](#), [1048](#),
[1122](#), [1208](#), [1227](#), [1228](#), [1235](#), [1243](#)
ALLCOMPLETE [940](#)
ALLDUMPS [931](#)
ALLOWPPRCP [886](#)
ALLOWXRC [887](#)
ALLVOLS [932](#)
ALTERNATE3590UNITNAME
A3590UNITNAME [1227](#)
ALTERNATEUNITNAME
AUNITNAME [1227](#)
ALTERNATEVOLUME [938](#)
ALTERNATEVOLUMES
AVOLS [1227](#)
ANY [1104](#), [1186](#), [1192](#)
ANYSSTORAGEVOLUME [1173](#)
APPLYINCREMENTAL [1005](#)
ARECOVER [898](#), [1031](#)
ARECOVERML2UNIT [1100](#)
ARECOVERPERCENTUTILIZED [1100](#)
ARECOVERTGTDGS [1100](#)
ARECOVERUNITNAME [1100](#)
ARPOOL [810](#), [813](#), [982](#), [983](#)
ASSOCIATED [941](#)
AUDIT [899](#), [1032](#)
AUTO [896](#), [900](#), [902](#), [907](#), [1037](#)
AUTOBACKUP [701](#)
AUTOBACKUPSTART [1101](#)
AUTODUMP [702](#)
AUTODUMPSTART [1102](#)
AUTOEXPIRATION [899](#), [1032](#)
AUTOMIGRATION [702](#), [900](#), [1032](#)

parameters (*continued*)

AUTOMIGRATIONSTART [1103](#)
AUTOPROGRESS [982](#), [984](#)
AUTORECALL [703](#)
AUTOREUSE [816](#)
AVAILABLE [939](#)
BACKUP [700](#), [746](#), [747](#), [810](#), [814](#), [841](#), [842](#), [900](#), [944](#),
[982](#), [985](#), [1017](#), [1032](#), [1048](#), [1104](#), [1176](#), [1227](#), [1228](#),
[1235](#)
BACKUPCONTENTS [931](#)
BACKUPCONTROLDATASET [743](#), [933](#), [1243](#)
BACKUPCOPIES [1110](#)
BACKUPDEVICECATEGORY [703](#), [783](#), [784](#), [790](#), [792](#),
[793](#), [1110](#)
BACKUPPREFIX [1107](#)
BACKUPSTORAGEGROUP [866](#)
BACKUPTAPES [1191](#)
BACKUPTYPE [744](#)
BACKUPVOLUME [878](#), [925](#)
BACKUPVOLUMES [744](#)
BCDSBACKUPDSN [1112](#)
BDELETE command
ALL [798](#)
BEGIN [973](#), [975](#)
BOTH [873](#), [933](#), [944](#), [1178](#)
BYPASS [724](#), [727](#)
CAPACITYMODE [939](#), [1198](#)
CATALOGEDDATA [854](#)
CC [762](#), [1206](#)
CDSVERSIONBACKUP [982](#), [985](#), [1109](#)
CHANGEDONLY [764](#), [1144](#)
CHECKFIRST [1019](#)
CLASSTRANSITIONS [1114](#)
CLEANUP [1154](#)
CLOUD [925](#), [1114](#), [1115](#)
CLOUDMIGRATE [1208](#)
COMMANDMIGRATION [1053](#), [1054](#)
COMMONQUEUE [745](#), [901](#), [985](#), [1033](#), [1117](#), [1119](#)
COMPACT [1122](#)
COMPACTPERCENT [1125](#)
CONNECT [1117](#), [1120](#)
CONNECTED [939](#)
CONTROL [755](#), [756](#)
CONTROLDATASETS [780](#), [781](#), [790](#), [791](#), [982](#), [986](#),
[1243](#)
CONVERSION [1125](#)
CONVERT [960](#), [965](#), [968](#)
COPYPOOL [866](#), [873](#), [885](#), [926](#), [986](#)
COPYPOOLCONTROLS [746](#)
COPYPOOLCOPY [843](#)
CPBSG [926](#)
CSALIMITS [982](#), [986](#), [1126](#)
CYCLESTARTDATE [810](#), [814](#)
DAILY [704](#), [743](#), [982](#), [985](#), [1017](#), [1044](#), [1045](#)
DAOPTION [991](#), [992](#), [1001](#)
DASD [703](#), [780](#), [783](#), [784](#), [792](#), [793](#), [879](#), [929](#), [1094](#),
[1099](#), [1104](#), [1110](#), [1130](#), [1186](#)
DASDBACKUP [1123](#), [1209](#)
DASDMIGRATE [1123](#), [1210](#)
DASDONLY [873](#)
DASDSELECTIONSIZE [1130](#)
DATABASEAUTHORITY [755](#), [756](#)
DATAMOVER [780](#), [783](#), [792](#), [1112](#)
DATASETCONFLICT [727](#)

parameters (*continued*)

[DATASETCONTROLS 746](#)
[DATASETINFORMATION 933](#)
[DATASETNAME 725, 802, 896, 926, 928, 958, 966, 982, 986, 1031, 1053, 1054](#)
[DATASETNAMES 746, 748](#)
[DATASETRESTORE 818](#)
[DATASETS 932](#)
[DATE 728, 865, 868, 888, 933, 998, 1002](#)
[DAY 819](#)
[DAYS 961, 969, 1128](#)
[DEBUG 1128](#)
[DEFERMOUNT 1129](#)
[DELAY 1104, 1105](#)
[DELETE 764, 770, 1046](#)
[DELETEBYAGE 698, 704, 967](#)
[DELETEIFBACKEDUP 704, 854, 965, 967](#)
[DEMOUNTDELAY 1130](#)
[DENSITY 705, 1110, 1129](#)
[DFDSSOPTION 991, 993, 1003](#)
[DIRECT 1192](#)
[DIRECTORYCONTROLS VOLUMES 747](#)
[DISABLE 820](#)
[DISASTERALTERNATEVOLUMES 940, 1235, 1238](#)
[DISASTERMODE 1130](#)
[DISCONNECT 1118, 1121](#)
[DISPLAY 852, 1018](#)
[DISPOSITION 820](#)
[DRAIN 706](#)
[DSBACKUP 1130](#)
[DSCOMMAND 900, 1032](#)
[DSNAME 886](#)
[DSS 783, 790, 792](#)
[DSSXMMODE 1133](#)
[DUMP 698, 700, 785, 793, 841, 842, 865, 866, 902, 1034, 1223](#)
[DUMPCCLASS 706, 780, 785, 794, 810, 815, 865, 867, 927, 940, 1005](#)
[DUMPCONTENTS 934](#)
[DUMPCYCLE 810, 829](#)
[DUMPEND 865, 866](#)
[DUMPGENERATION 1005](#)
[DUMPIO 1134](#)
[DUMPONLY 865, 867, 868, 873](#)
[DUMPSTATE 940](#)
[DUMPTAPES 1191](#)
[DUMPVOLS 932](#)
[DUMPVOLUME 927, 1005](#)
[DUPLEX 1135](#)
[EEFMT2 941](#)
[EEFMT3 941](#)
[EEFMT4 941](#)
[EFMT1 941](#)
[EFMT2 941](#)
[EFMT3 941](#)
[EFMT4 941](#)
[EMERGENCY 1136](#)
[EMPTY 941](#)
[ENCRYPTED 941](#)
[END 973, 975](#)
[ENDKEY 857](#)
[ENDOFDATASET 903, 904](#)
[ENDOFVOLUME 904](#)
[ERASEONSCRATCH 1137](#)

parameters (*continued*)

[ERRORS 752](#)
[EXCEPTIONONLY 1098](#)
[EXCESSIVEVOLUMES 942](#)
[EXCLUDE 764, 766, 1004](#)
[EXECUTE 692, 724, 726, 852, 867, 1018](#)
[EXITOFF 1138](#)
[EXITON 1139](#)
[EXITS 1141](#)
[EXPDT 1227, 1232](#)
[EXPIRATION 1195](#)
[EXPIRATIONDATE 1110](#)
[EXPIRATIONINCLUDE 1195](#)
[EXPIREBV 896, 905, 1030, 1035](#)
[EXPIRED 942](#)
[EXPIREDDATASETS 1141](#)
[FAILED 940](#)
[FAILEDCREATE 943](#)
[FAILEDRECYCLE 943](#)
[FASTREPLICATION](#)
 [DATASETRECOVERY 1142](#)
 [FCRELATION 1142](#)
 [VOLUMEPAIRMESSAGES 1142](#)
[FASTREPLICATIONDUMP 815, 902, 1034](#)
[FASTREPLICATIONEND 865, 866](#)
[FASTREPLICATIONSTATE 940](#)
[FASTREPLICATIONVOLS 932](#)
[FCFRRINCOMPLETE 940](#)
[FILECONTROLS 747](#)
[FILENAME 802, 982](#)
[FILTEROUTPUTDATASET 692, 693](#)
[FIX 750](#)
[FORCE 865, 869, 889, 1019](#)
[FORCENONSMS 991, 993, 998, 1005](#)
[FRBACKUP 905, 1035](#)
[FREQUENCY 715, 716, 794, 822, 1143](#)
[FROMANY 1044, 1047, 1048](#)
[FROMCOPYPOOL 889](#)
[FROMDASD 889](#)
[FROMDATE 1046](#)
[FROMDUMP 889, 1005](#)
[FROMMIGRATIONLEVEL1 1047, 1048](#)
[FROMMIGRATIONLEVEL2 1044, 1047, 1048](#)
[FROMPRIMARY 1047](#)
[FROMVOLUME 797, 799, 998, 1007](#)
[FRRECOV 905, 1035](#)
[FULL 941, 1094, 1098](#)
[FULLDSNAME 1022](#)
[FUNCTION 1046](#)
[FUNCTION\(BACKUP\) 1044, 1047](#)
[FUNCTION\(DELETE\) 1047](#)
[FUNCTION\(MIGRATION\) 1047](#)
[FUNCTION\(RECALL\) 1044, 1048](#)
[FUNCTION\(RECOVER\) 1048](#)
[FUNCTION\(RECYCLE\) 1048](#)
[FUNCTION\(SPILL\) 1044, 1049](#)
[FUNCTION\(TRANSITION\) 1049](#)
[GENERATION 865, 868, 888, 1002](#)
[HARDCOPY 1036](#)
[HOST 928](#)
[HSERIALIZATION 1129](#)
[HSM 716, 783, 792](#)
[HSMHOST 973, 974](#)
[HMSYM 974](#)

parameters (*continued*)

[HSM TAPE 1189](#)
[HW COMPRESS 823](#)
[IMAGE 982, 987](#)
[INACTIVE 1127](#)
[INCACHE 1178](#)
[INCLUDE PRIMARY 932, 934](#)
[INCOMPLETE RECOVERY 940](#)
[INCREMENTAL 785, 793](#)
[INCREMENTAL BACKUP 1144](#)
[INDATASET 1227, 1228, 1235, 1236](#)
[INPUT TAPE ALLOCATION 1145](#)
[INSTRUCTION 724, 728](#)
[INTERVAL MIGRATION 1145](#)
[INUSE 1104, 1105](#)
[JES2 1147](#)
[JES3 1147](#)
[JOURNAL 1147, 1244](#)
[JRN L BACKUP DSN 1114](#)
[KEEP MIG DATE 961](#)
[KEYS 831](#)
[LOVOLS 813](#)
[LAST COPY 841, 843](#)
[LEVEL 926, 928, 1053, 1054](#)
[LEVEL 0 692, 694](#)
[LEVELS 746, 748](#)
[LIBRARY 943](#)
[LIBRARY BACKUP 1198](#)
[LIBRARY MIGRATION 1198](#)
[LIKE 1173](#)
[LIMIT 1016, 1020](#)
[LIST 692, 693, 907, 1036](#)
[LOG 907, 953, 1036, 1225](#)
[MARK FULL 841, 843, 1168](#)
[MASTER CATALOG 748](#)
[MASTER SCHEDULE CANDIDATE 1118, 1121](#)
[MAX ABARS ADDRESS SPACE 1148](#)
[MAX BACKUP TASKS 1149](#)
[MAX COPY POOL TASKS 1149](#)
[MAX DS RECOVER TASKS 1150](#)
[MAX DUMP RECOVER TASKS 1151](#)
[MAX DUMP TASKS 1151](#)
[MAX EXTENTS 1152](#)
[MAX IDLE TASKS 1130](#)
[MAXIMUM 1127](#)
[MAX INTERVAL TASKS 1152](#)
[MAX MIGRATION TASKS 1153](#)
[MAX RECALL TASKS 1153](#)
[MAX RECYCLE TASKS 1154](#)
[MAX SSMTASKS 1154](#)
[MAX STACK 786, 825](#)
[MCDS BACKUP DSN 1113](#)
[MEDIA CONTROLS 748](#)
[MENTITY 728](#)
[MESSAGE DATASET 869, 890](#)
[MIG DENSITY 1155](#)
[MIGRATE 704, 956, 967](#)
[MIGRATED DATA 728](#)
[MIGRATE PREFIX 1155](#)
[MIGRATE PSMEARLY 1158](#)
[MIGRATION 700, 746, 841, 842, 907, 1030, 1037, 1048, 1053, 1054, 1176, 1235, 1236](#)
[MIGRATION AUTO CLOUD 1156](#)
[MIGRATION CLEANUP CYCLE 810, 830](#)

parameters (*continued*)

[MIGRATION CLEANUP DAYS 1157](#)
[MIGRATION CONTROL DATASET 743, 933, 1243](#)
[MIGRATION LEVEL 1 692, 694, 707, 879, 929, 944, 956, 958, 962, 965, 966, 970](#)
[MIGRATION LEVEL 1 DAYS 1158](#)
[MIGRATION LEVEL 2](#)
[ML 2 1227](#)
[MIGRATION SUBTASKS 1159](#)
[MIGRATION TAPES 1191](#)
[MIGRATION VOLUME 878, 929](#)
[MIG UNIT NAME 1159](#)
[MIN STACK 786, 825](#)
[MINUTES 1130](#)
[ML 1 728](#)
[ML 1 BACKUP VERSIONS 878](#)
[ML 1 OVERFLOW 1160](#)
[ML 1 VOLS 813](#)
[ML 2 724, 728, 743, 1016](#)
[ML 2 PARTIALS NOT ASSOCIATED GOAL 1161](#)
[ML 2 RECYCLE PERCENT 1161](#)
[ML 2 TAPE 1192](#)
[MONITOR 1162](#)
[MONITOR \(BACKUP CONTROL DATASET\) 1163](#)
[MONITOR \(JOURNAL\) 1163](#)
[MONITOR \(MIGRATION CONTROL DATASET\) 1163](#)
[MONITOR \(NOSPACE\) 1164](#)
[MONITOR \(NOSTARTUP\) 1164](#)
[MONITOR \(NOVOLUME\) 1165](#)
[MONITOR \(OFFLINE CONTROL DATASET\) 1164](#)
[MONITOR \(SPACE\) 1164](#)
[MONITOR \(STARTUP\) 1164](#)
[MONITOR \(VOLUME\) 1165](#)
[MOUNT WAIT TIME 1165](#)
[MOVE 692, 693, 962](#)
[MULTIPLE VOLUME 944](#)
[MWE 1127](#)
[NETWORK 982, 987](#)
[NEW DIR 1007](#)
[NEW NAME 767, 891, 1007](#)
[NO 1194](#)
[NO ACCEPT PS CB USER ID 1098](#)
[NO ALTERNATE VOLUME 938](#)
[NO AUTO BACKUP 698, 701](#)
[NO AUTO DUMP 702](#)
[NO AUTO MIGRATION 698, 702](#)
[NO AUTO RECALL 703](#)
[NO AUTO REUSE 816](#)
[NO BACKUP 1104](#)
[NO BACKUP MIGRATED 729, 730](#)
[NO CC 1206](#)
[NO CONVERSION 1125](#)
[NO COPY POOL BACKUP 891](#)
[NO CSALIMITS 1126](#)
[NO DASD BACKUP 1123](#)
[NO DASD MIGRATE 1123](#)
[NO DATASET INFORMATION 933](#)
[NO DATASET RESTORE 818](#)
[NODE BUG 1128](#)
[NO DRAIN 698, 706](#)
[NO EMERGENCY 1136](#)
[NO ERASE ON SCRATCH 1137](#)
[NO FIX 750](#)
[NO FUNCTION 1046](#)

parameters (*continued*)

NOINTERVALMIGRATION [1145](#)
 NOJOURNAL [1147](#)
 NOLIBRARY [943](#)
 NOLIMIT [786, 794, 824, 1198](#)
 NOMIGRATION [1053, 1054](#)
 non-ABARS [1198](#)
 NONE [703, 940, 1122, 1169, 1192, 1208](#)
 NONPREFERREDFRVOLS [940](#)
 NONRECOVERABLE [940](#)
 NONSMSVERSIONS [854](#)
 NOOOVERFLOW [945](#)
 NOOPTIMUMDASDBLOCKING [1167](#)
 NOOVERFLOW [707](#)
 NOPARALLEL [783, 790, 792, 1110](#)
 NOPROFILEBACKUP [1172](#)
 NORACFIND [1173](#)
 NOREQUEST [1179](#)
 NORESET [824](#)
 NORETENTIONLIMIT [945](#)
 NOSCRATCH [1141](#)
 NOSKIPABPRIMARY [1183](#)
 NOSMALLDATASETPACKING [708, 946, 1183](#)
 NOSMF [1185](#)
 NOSPILL [1186](#)
 NOSTACK [692, 695, 731](#)
 NOSWAP [1187](#)
 NOSYS1DUMP [1188](#)
 NOTAPEBACKUP [1124](#)
 NOTAPEHARDWARECOMPACT [1190](#)
 NOTAPEMIGRATE [1124](#)
 NOTASSOCIATED [941](#)
 NOTCONNECTED [939](#)
 NOTENCRYPTED [941](#)
 NOTFULL [941](#)
 NOTINCACHE [1178](#)
 NOUSERUNITTABLE [1202](#)
 NOVOLS [932](#)
 NOVTOCENQ [870](#)
 NOWAIT [913, 914, 1145, 1168, 1177](#)
 NPFRV [940](#)
 NULLJOURNALONLY [780, 784, 792](#)
 OBJECTNAMES [1166](#)
 OCDSBACKUPDSN [1113](#)
 ODMNOTIFICATIONLIMIT [1167](#)
 of the recycle process, defining [537](#)
 OFF [1169](#)
 OFFLINECONTROLDATASET [743, 1243](#)
 ON [1169](#)
 ONDEMANDMIGRATION [1166](#)
 ONLYDISASTERALTERNATES [1235, 1237](#)
 OPTIMIZE [692, 694](#)
 OPTIMUMDASDBLOCKING [1167](#)
 ORIGINAL [1144](#)
 ORIGINALVOLUMES
 OVOLES [1227](#)
 OUTDATASET [751, 852, 858, 934, 1020, 1044, 1049](#)
 OUTPUTTAPEALLOCATION [1168](#)
 OVERFLOW [707, 945](#)
 PARALLEL [783, 792, 1110](#)
 PARTIAL [940](#)
 PARTIALOK [889](#)
 PARTIALTAPE [1168](#)
 PASSWORD [1195](#)

parameters (*continued*)

PDA [1169, 1225, 1226](#)
 PERCENTFULL [1198](#)
 PERCENTUTILIZED [730](#)
 PERCENTVALID [1021](#)
 PLEXNAME [1170](#)
 POOL [810, 832, 982, 987](#)
 PREFIX [1022](#)
 PREPARE [726, 867](#)
 PRIMARY [700, 791, 841, 842, 966](#)
 PRIMARYSPMGMTCYCLE [810, 832](#)
 PRIMARYSPMGMTSTART [1171](#)
 PRIMARYVOLUME [929](#)
 PRIVATEVOLUME [1173](#)
 PROCESSIONLY [692, 694](#)
 PROFILEBACKUP [1172](#)
 PROMOTE [1172, 1224](#)
 PULLSIZE [1022](#)
 PURGE [839-841, 843, 1253](#)
 RACF [1195](#)
 RACFINCLUDE [1195](#)
 RACFIND [1173](#)
 REASSIGN [841, 843](#)
 REBLOCKTOANY [1125](#)
 RECALL [896, 907, 1037, 1173](#)
 RECALLTAKEAWAY [945](#)
 RECONNECT
 ALL [1192](#)
 ML2DIRECTEDONLY [1192](#)
 NONE [1192](#)
 RECOVER [908, 1030, 1038](#)
 RECOVERABLE [940](#)
 RECOVERNEWNAMEALL [730](#)
 RECOVERNEWNAMELEVEL [730](#)
 RECOVERRESET [823](#)
 RECOVERY [1147](#)
 RECURSE [769, 856, 1008, 1009](#)
 RECYCLE [908, 1038](#)
 RECYCLEINPUTDEALLOCFREQUENCY [1174](#)
 RECYCLEOUTPUT [1176](#)
 RECYCLEPERCENT [1176](#)
 RECYCLETAKEAWAY [941](#)
 RECYCLETAKEAWAYRETRY [1177](#)
 RECYCLETAPEALLOCATION [1177](#)
 RECYCLETAPERESIDENCE [1178](#)
 REDUCED [1094, 1098](#)
 RELBLK [991, 992, 1001](#)
 RELTRK [991, 992, 1001](#)
 REMOVECOMPACTNAMES [1178](#)
 RENAMESOURCE [727](#)
 RENAMETARGET [727](#)
 REPLACE [891, 1009](#)
 REPORT [752, 909, 1038](#)
 REQUEST [802, 982, 986, 1179](#)
 REQUIREDCOMPLETE [940](#)
 RESET [824, 935, 1235](#)
 RESUME [746, 748, 855, 889](#)
 RESUMEAUDIT [945](#)
 RETAIN [982, 987](#)
 RETAINDAYS [183, 769, 938](#)
 RETAINNEWESTVERSION [880](#)
 RETAINVERSIONS [853](#)
 RETENTIONPERIOD [786, 790, 794, 824, 1110](#)
 RETPD [1227, 1232](#)

parameters (*continued*)

RETRY [1104](#), [1105](#)
RETURNCONTROL [865](#), [866](#)
REUSE [1168](#)
REVOKE [755](#), [756](#)
RMM [1039](#)
ROUTETOTAPE
 ANY [1192](#)
 unittype [1192](#)
RSA [891](#)
SAMETRK [991](#), [992](#), [998](#), [1001](#)
SASINDEX [803](#)
SCRATCH [1141](#), [1181](#)
SCRATCHFREQUENCY [1179](#)
SCRATCHTAPE [1189](#)
SECONDARYSPMGMTCYCLE [830](#)
SECONDARYSPMGMTCYLCE [810](#)
SECONDARYSPMGMTDIRECTIVE [1179](#)
SECONDARYSPMGMTSTART [1180](#)
SECURITY [987](#)
SELECT [936](#), [1022](#)
SELECT(EXCLUDE) [1016](#)
SELECT(INCLUDE) [1016](#)
SELECTVOLUME [1181](#)
SERIALIZATION [1104](#), [1105](#)
SERIALIZATION CONTINUOUS [752](#)
SERIALIZATION DYNAMIC [752](#)
SETSYS [982](#), [988](#)
SKIP [692](#), [694](#)
SKIPABPRIMARY [1183](#)
SMALLDATASETPACKING [708](#), [748](#), [946](#), [1183](#)
SMF [1185](#)
SOURCELEVEL [728](#)
SOURCENAMES [1185](#)
SOURCEVOLUME [998](#), [1005](#)
SPACE [982](#), [988](#)
SPECIFIC [1181](#)
SPEED [1147](#)
SPILL [698](#), [704](#), [743](#), [879](#), [982](#), [985](#), [1186](#)
STACK [692](#), [695](#), [724](#), [731](#), [780](#), [786](#), [790](#), [795](#), [825](#)
STARTKEY [852](#), [855](#)
STARTUP [982](#), [988](#)
STATISTICS [982](#), [988](#)
STORAGEGROUP [780](#), [781](#), [959](#)
STORAGEGROUPAFFINITY [1187](#)
SUMMARY [947](#), [1044](#), [1050](#)
SWAP [1187](#)
SWITCHTAPES [810](#), [833](#)
SYS1DUMP [1188](#)
SYSFREQUENCY [715](#), [716](#)
SYSNAME [974](#)
SYSOUT [751](#), [858](#), [934](#), [1049](#), [1094](#), [1099](#), [1187](#)
SYSVERSIONS [715](#), [717](#)
TAPE
 ANY [1192](#)
 unittype [1192](#)
TAPEBACKUP [1124](#), [1210](#)
TAPECOPY [909](#), [1030](#), [1039](#), [1194](#)
TAPEDATASET [896](#), [908](#), [1038](#)
TAPEDATASETORDER [1188](#)
TAPEDELETION [1189](#)
TAPEEXPIRATIONDATE [827](#)
TAPEHARDWARECOMPACT [1190](#)
TAPEINPUTPROMPT [1191](#)

parameters (*continued*)

TAPELIST [1022](#)
TAPEMAXRECALLTASKS [1192](#)
TAPEMIGRATE [1124](#), [1211](#)
TAPEMIGRATION [1192](#)
TAPEMOVEMENT [1154](#)
TAPEOUTPUTPROMPT [1194](#)
TAPERECALLLIMITS [1195](#)
TAPEREREPL [896](#), [909](#), [1039](#)
TAPESECURITY [1195](#)
TAPESPANSIZE [1197](#)
TAPETABLEOFCONTENTS [930](#)
TAPETABLEOFCONTENTS(volser) [930](#)
TAPETYPE [1198](#)
TAPEUTILIZATION [1198](#)
TARGET [770](#)
TARGETLEVEL [879](#)
TARGETUNIT [724](#), [731](#)
TARGETVOLUME [1009](#)
TCBADDRESS [803](#)
TERMINAL [751](#), [788](#), [795](#), [865](#), [934](#), [963](#), [970](#), [1010](#)
TGTGDS [732](#)
THRESHOLD [708](#)
TIME [1002](#)
TOANY [1044](#), [1047](#)
TODATE [1050](#)
TOKEN [865](#), [868](#), [870](#), [873](#), [888](#)
TOMIGRATIONLEVEL1 [1047](#)
TOMIGRATIONLEVEL2 [1047](#)
TOTAL [785](#), [790](#), [793](#), [1022](#)
TOVOLUME [886](#), [1010](#)
TRACKMANAGEDTHRESHOLD [710](#)
TRAPS [982](#), [988](#)
TSO [907](#), [1037](#)
UNASSIGN [841](#), [843](#)
UNASSIGNED [982](#), [985](#)
UNASSIGNTAPEUNIT [828](#)
UNAVAILABLE [946](#)
UNCATALOGEDDATA [854](#)
UNEXPIRED [946](#)
UNIT [692](#), [695](#), [700](#), [724](#), [732](#), [760](#), [770](#), [795](#), [827](#),
[970](#), [991](#), [993](#), [998](#), [1011](#)
UNITNAME [1110](#), [1201](#)
UNITTYPE [1198](#)
UNIXPATH [855](#), [857](#)
UNLIKE [1173](#)
USECYLINDERMANAGEDSPACE [1202](#)
used in creating backup profiles [207](#)
USER [755](#), [756](#), [931](#), [982](#), [986](#)
USERCATALOG [748](#)
USERID [802](#)
userid DATABASEAUTHORITY [756](#)
userid REVOKE [756](#)
USERSERIALIZATION [1129](#)
USERTAPE [692](#), [694](#)
USERUNITTABLE [1202](#)
VERIFY [692](#), [726](#), [892](#), [1018](#)
VERSION [728](#), [865](#), [868](#), [888](#), [933](#), [1002](#)
VERSIONS [715](#), [717](#), [797](#), [799](#), [873](#), [1205](#)
VOLCOUNT [733](#), [1206](#)
VOLUME [760](#), [770](#), [929](#), [944](#), [958](#), [965](#), [967](#), [991](#), [994](#),
[1016](#), [1053](#), [1054](#)
VOLUMECONTROLS BACKUP [749](#)
VOLUMECONTROLS MIGRATION [749](#)

- parameters (*continued*)
 - VOLUMECONTROLS RECOVERABLE [749](#)
 - VOLUMEDUMP [1206](#)
 - VOLUMEPOOL [810](#), [834](#), [982](#), [988](#)
 - VOLUMES [725](#), [744](#), [748](#), [749](#), [782](#), [791](#), [831](#), [832](#), [834](#), [1050](#)
 - VSAM [947](#)
 - VTOCCOPIES [828](#)
 - WAIT [913](#), [914](#), [1145](#), [1168](#), [1177](#)
 - WAITING [982](#), [988](#)
 - WITHDRAW [867](#)
 - XMIT [725](#)
 - YES [1194](#)
 - ZCOMPRESS [829](#), [1207](#)
- partial
 - release, specifying [36](#)
 - tapes
 - converting to full tapes [440](#)
 - specifying desired number of [141](#)
- PARTIAL RELEASE attribute [36](#)
- partitioned data set
 - deleting [89](#)
- partitioning DFSMSHsm activities [593](#)
- PATCH command
 - for DFSMSHsm [979](#)
- PDA (problem determination aid)
 - specifying use of [586](#)
 - using [586](#)
- PDA parameter, SETSYS command [586](#)
- PERCENTUTILIZED parameter, ARECOVER command [395](#)
- PERCENTVALID parameter, RECYCLE command
 - calculation for [541](#)
 - CHECKFIRST option [541](#)
- performance considerations
 - fast replication [277](#)
- phase 1
 - backing up the control data sets [199](#)
 - DFSMSHsm-managed volume space management, SMS-managed storage [65](#)
 - processing example, SMS-managed storage [69](#)
- phase 2
 - moving backup versions [200](#)
 - processing example, SMS-managed storage [79](#)
 - SMS-managed storage [70](#)
- phase 3
 - backing up migrated data sets [200](#)
- phase 4
 - backing up DFSMSHsm-managed volumes from storage groups with the automatic backup attribute [201](#)
- physical data set restore
 - HRECOVER considerations [244](#)
- pool
 - data set
 - definition [19](#)
 - of volumes
 - configuration in JES3 [106](#)
 - defining [105](#)
 - general, configuration [106](#)
 - general, definition [106](#)
 - in systems running under JES2 [105](#)
 - requirements and restrictions [105](#)
- pool configuration in JES3 [107](#)
- POOL parameter, DEFINE command [105](#)
- pooling
 - pooling (*continued*)
 - examples of JES3 [107](#)
 - PREFIX parameter
 - RECYCLE command [540](#)
 - PREPARE parameter, ARECOVER command [393](#), [398](#)
 - preparing for aggregate recovery [374](#)
 - prevent-migration date, description [117](#)
 - prevented functions, allowing to run [600](#)
 - preventing
 - aggregate backup [897](#)
 - aggregate recovery [898](#)
 - all DFSMSHsm functions [898](#)
 - all recover functions [908](#)
 - automatic backup [900](#)
 - automatic volume space management [899](#), [900](#)
 - backups [900](#)
 - data set backups by command [900](#)
 - data set backups to DASD [900](#)
 - data set backups to tape [900](#)
 - deletion of migrated data sets [907](#)
 - dumps [902](#)
 - logging [907](#)
 - recall of migrated data sets [907](#)
 - space management [899](#), [900](#), [907](#)
 - tape copy processes [909](#)
 - tape recover functions [908](#)
 - tape replacement processes [909](#)
 - prevention of
 - ABACKUP command processing [597](#)
 - all DFSMSHsm processing [596](#)
 - ARECOVER command processing [597](#)
 - AUDIT command processing [597](#)
 - backup processing [449](#), [597](#)
 - DFSMSHsm functions from running [596](#)
 - dump processing [456](#), [597](#)
 - entries in the DFSMSHsm log [598](#)
 - EXPIREBV command processing [598](#)
 - fast replication processing [288](#)
 - further migration to a DASD migration volume [535](#)
 - LIST command processing [598](#)
 - migration processing [599](#)
 - processing of data sets during volume space management [149](#)
 - recall processing [599](#)
 - recovery processing [459](#), [599](#)
 - recycle processing [600](#)
 - REPORT command processing [600](#)
 - restore processing [459](#)
- PRIMARY DAYS NON-USAGE attribute [37](#)
- PRIMARY parameter
 - ADDVOL command [304](#)
 - ADDVOL, pool configuration in JES3 [106](#)
 - BACKVOL command [309](#)
- primary space management
 - automatic restarts of [162](#)
 - concept [23](#)
 - defining a processing window [164](#)
 - manual restarts of [163](#)
 - scenario [81](#)
- primary volume
 - attributes [103](#)
 - defining to DFSMSHsm [102](#)
 - space management [111](#)
- printing DFSMSHsm log information [607](#)

- priority
 - of data set recoveries [249](#)
 - of recalls and deletions [135](#)
 - of volume recoveries [302](#)
- order
 - for migration, SMS-managed storage [72](#)
 - sort data sets, SMS-managed storage [72](#)
- priority of queued requests
 - altering [719](#)
- PRIVATEVOLUME subparameter, RECALL parameter, SETSYS command
 - pool configuration [106](#)
 - pool configuration in JES3 [106](#)
- problems with recalling GUARANTEED SPACE VSAM data sets, space management [153](#)
- problems with recovering GUARANTEED SPACE VSAM data sets, availability management [451](#)
- procedures
 - availability management [485](#)
 - backing up and recovering
 - ICF catalogs [487](#)
 - backup and recovery [485](#)
 - full-volume dump [489](#)
 - interrupting and restarting
 - automatic backup processing [485](#)
 - automatic dump processing [489](#)
 - sending alternate tapes off site [441](#)
 - space management [161](#)
 - stopping DFSMSHsm [595](#)
 - using ICFRU Forward Recovery Utility [298](#)
- process data set records from BCDS, EXPIREBV processing [235](#)
- process window, defining a new one
 - for primary space management [161](#)
 - for secondary space management [161](#)
- processing
 - ABARS ICF user catalog and catalog alias [356](#), [410](#)
 - cleanup [218](#)
 - example, backup [199](#)
 - for expired backup versions of non-SMS-managed data sets [310](#)
 - for non-SMS-managed data sets, migrating all data sets from migration level 1 volume [534](#)
 - for SMS-managed data sets, migrating all data sets from migration level 1 volume [533](#)
 - generation data sets [387](#)
 - migrated data sets
 - in the ALLOCATE list [389](#)
 - in the INCLUDE list [389](#)
 - order, ARECOVER conflict resolution [406](#)
 - prevent data set [149](#)
 - priority of data set recovery [249](#)
 - priority of recalls and deletions [135](#)
 - priority of volume recovery [302](#)
 - time, tracking for ABARS [353](#), [413](#)
 - unit serialization information, obtaining [565](#)
 - units that process each storage group, specifying [35](#)
- PROFILE parameter, SETSYS command [190](#)
- profile, RACF
 - authorization checking during recovery [460](#)
 - backup
 - deleting [207](#)
 - for migrated data sets [207](#)
 - parameters used in creating [207](#)

- profile, RACF (*continued*)
 - discrete
 - profile data set [618](#)
 - recovering [461](#)
 - specifying whether to back up [190](#)
 - generic [617](#)
 - modeling, effects on profile recovery [462](#)
 - recovered, characteristics [462](#)
 - recovering [460](#)
 - recovery
 - accessing a data set when profile cannot be recovered [462](#)
 - authorization checking during [462](#)
 - effects of profile modeling [462](#)
 - programming interface information [1418](#)
 - protect-all environment, RACF [619](#)
 - protection
 - password
 - data sets on DASD [151](#)
 - recovered data sets, maintaining [463](#)
 - RACF
 - data sets [617](#)
 - discrete data set profiles [618](#)
 - generic data set profiles [617](#)
 - PURGE parameter, DELETE command [89](#)
 - PURGE parameter, DELVOL command [536](#)
 - purpose of
 - availability management [7](#)
 - reblocking [87](#)
 - space management [7](#)

Q

- QUERY
 - ABARS [345](#)
 - ARPOOL [383](#)
 - COMMONQUEUE [572](#), [614](#)
 - find request numbers, user IDs, and data set names [604](#)
- QUERY command
 - for DFSMSHsm [981](#)
 - messages associated [1389](#)
 - optional parameters
 - ABARS [983](#)
 - ACTIVE [983](#)
 - ARPOOL [983](#)
 - AUTOPROGRESS [984](#)
 - BACKUP [985](#)
 - CDSVERSIONBACKUP [985](#)
 - COMMONQUEUE [985](#)
 - CONTROLDATASETS [986](#)
 - COPYPOOL [986](#)
 - CSALIMITS [986](#)
 - DATASETNAME [986](#)
 - IMAGE [987](#)
 - MIGRATIONLEVEL2 [987](#)
 - NETWORK [987](#)
 - POOL [987](#)
 - REQUEST [986](#)
 - RETAIN [987](#)
 - SECURITY [987](#)
 - SETSYS [988](#)
 - SPACE [988](#)
 - STARTUP [988](#)
 - STATISTICS [988](#)

QUERY command (*continued*)
 optional parameters (*continued*)
 TRAPS [988](#)
 USER [986](#)
 VOLUMEPOOL [988](#)
 WAITING [988](#)
 using [1389](#)

QUERY COPYPOOL
 percent complete [278](#)
querying held ABARS functions [409](#)
queued DFSMSHsm request
 altering the priority [719](#)

R

RACF (Resource Access Control Facility)
 access with RACF FACILITY class [625](#)
 ARCCATGP logon group [86](#)
 authority needed to perform DFSMSHsm functions [620](#)
 authority required for space management functions [150](#)
 authorization checking
 during HALTERDS processing [460](#)
 during HBDELETE processing [460](#)
 during recovery [460](#)
 checking during aggregate backup [340](#)
 checking during aggregate recovery [376](#)
 considerations [617](#)
 discrete data set profiles
 for cataloged data sets, backing up [206](#)
 recovering [461](#)
 specifying whether to back up [190](#)
 discrete profiles
 data set [618](#)
 data set, aggregate recovery [394](#)
 for aggregate recovery [387](#)
 facility authorization checking, ARECOVER data set
 name conflict resolution [407](#)
 generic data set profiles [617](#)
 indication [618](#)
 indicator
 definition [618](#)
 with always-call [619](#)
 model entity [387](#)
 protect-all environment [619](#)
 protecting data sets [617](#)
 RACF catalog maintenance group [86](#)
 security bypassing for DFSMSHsm commands [621](#)
 started task table [622](#)
 started-procedures table, printing DFSMSHsm log
 information [608](#)
 user ID for DFSMSHsm [622](#)
RACF and DFSMSHsm, concepts [620](#)
RACF-indicated data sets
 backing up [625](#)
 definition [618](#)
 deleting [150](#)
 protecting aggregate backup volumes [340](#)
 recalling [151](#)
 recovering or restoring [461](#)
RACF-protected data sets, migrating [150](#)
REASSIGN parameter, DELVOL command [536](#)
reblocking
 during recall [87](#)
 during recovery [249](#)

reblocking (*continued*)
 purpose [87](#)
recall
 additional considerations [86](#)
 allocating data sets to volumes, non-SMS-managed [120](#)
 attributes [104](#)
 automatic
 command data set scratching, non-SMS-managed
 [121](#)
 command deletion of VSAM AIX clusters [85](#)
 data set renaming [85](#)
 during catalog updates [86](#)
 during data set allocation [85](#)
 during non-VSAM data set opening, non-SMS-
 managed [121](#)
 during VSAM data set opening, non-SMS-managed
 [121](#)
 SMS-managed [83](#)
 with DFSMSdss [85](#)
 command
 direct access data sets [93](#)
 non-SMS-managed data set to specific volume [124](#)
 SMS-managed BDAM data set to non-SMS-
 managed volume [125](#)
 SMS-managed data set to non-SMS-managed
 volume [125](#)
 SMS-managed data sets [92](#)
 SMS-managed extended format data set to non-
 SMS-managed volume [125](#)
 data set
 migrated for extent reduction [85](#)
 SMS-managed BDAM data set to non-SMS-
 managed volume [125](#)
 SMS-managed data set to non-SMS-managed
 volume [125](#)
 SMS-managed extended format data set to non-
 SMS-managed volume [125](#)
 to a specific volume, non-SMS-managed [124](#)
 during catalog updates [86](#)
 JES3 considerations for non-SMS-managed data sets
 [102](#), [121](#)
 non-SMS-managed data sets [119](#)
 priority [135](#)
 RACF-indicated data sets [151](#)
 reblocking during [87](#)
 SMS-managed BDAM data set to non-SMS-managed
 volume [125](#)
 SMS-managed data set to non-SMS-managed volume
 [125](#)
 SMS-managed extended format data set to non-SMS-
 managed volume [125](#)
 specifying characteristics for non-SMS-managed
 volumes [101](#)
 tape take away by [137](#)
 tape volume contention [136](#)
 tasks, specifying [52](#)
 volume contention
 multiple-processing-unit environment [139](#)
 volume selection, non-SMS-managed [119](#)
RECALL command
 examples [994](#)
 optional parameters
 DAOPTION [992](#)
 DFDSSOPTION [993](#)

- RECALL command (*continued*)
 - optional parameters (*continued*)
 - FORCENONSMS [993](#)
 - UNIT [993](#)
 - VOLUME [994](#)
 - overview [991](#)
 - required parameters
 - dsname [991](#)
 - syntax [991](#)
- RECALL parameter
 - DAOPTION parameter [93](#), [125](#)
 - FORCENONSMS parameter [92](#), [93](#), [125](#)
 - HOLD command [599](#)
 - JES3 restrictions and recommendations [122](#)
 - RELEASE command [603](#)
 - SETSYS command [101](#)
- recalling
 - extent-reduced data sets [85](#)
 - problems that can occur
 - changes in the storage group [153](#)
 - changes in the storage-class-to-data set mapping [153](#)
 - conversions of data sets [154](#)
 - GUARANTEED SPACE VSAM data sets [152](#)
- reclaiming unused DASD space
 - deleting expired backup versions, non-SMS-managed [310](#)
 - deleting expired backup versions, SMS-managed [233](#)
 - deleting expired data sets, space management [49](#)
 - specifying
 - extent reduction [49](#)
 - migration level 1 space [54](#)
 - partial release [36](#)
- reconnection [48](#)
- record level sharing, accessing CDSs with [504](#)
- record, ABARS activity [381](#)
- records
 - for migration cleanup, specifying age [54](#)
- RECOVER
 - optional parameters
 - APPLYINCREMENTAL [295](#), [316](#)
 - DAOPTION [248](#)
 - DATE [241](#)
 - DUMPCCLASS [252](#)
 - DUMPVOLUME [252](#)
 - FROMDUMP [252](#)
 - FROMVOLUME [311](#)
 - GENERATION [241](#)
 - NEWNAME [241](#), [252](#)
 - REPLACE [241](#), [252](#)
 - recovering a volume [294](#)
 - required parameters
 - TOVOLUME [252](#), [311](#), [314](#)
 - UNIT [252](#), [311](#), [314](#)
- RECOVER command
 - examples [1011](#)
 - optional parameters
 - DAOPTION [1001](#)
 - DATE [1002](#)
 - DFDSSOPTION [1003](#)
 - EXCLUDE [764](#), [1004](#)
 - FORCENONSMS [1005](#)
 - FROMDUMP [1005](#)
 - FROMVOLUME [1007](#)
- RECOVER command (*continued*)
 - optional parameters (*continued*)
 - GENERATION [1002](#)
 - NEWDIR [1007](#)
 - NEWNAME [1007](#)
 - RECURSE [1008](#), [1009](#)
 - REPLACE [1009](#)
 - RSA [891](#)
 - SOURCEVOLUME [1005](#)
 - TARGETVOLUME [1009](#)
 - TERMINAL [1010](#)
 - TOVOLUME [1010](#)
 - UNIT [1011](#)
 - VERSION [1002](#)
 - overview [997](#)
 - required parameters
 - * [1000](#), [1001](#)
 - dsname [1000](#)
 - pathname [1001](#)
 - syntax [998](#)
 - recover copy pool from incremental version [279](#)
 - recover copy pool using fast reverse restore [278](#)
 - recovered
 - control data sets, updating [516](#)
 - data sets, maintaining password protection [463](#)
 - profile, characteristics [462](#)
 - recovering
 - a migrated data set [551](#)
 - a migrated volume [552](#)
 - a non-VSAM data set from a DFSMSHsm backup version [245](#)
 - a VSAM data set from a DFSMSHsm backup version [246](#)
 - ABARS instruction data set [400](#)
 - backup versions from a damaged ML1 volume [554](#)
 - damaged single-file ML2 volumes [551](#)
 - data sets to SMS-managed volumes [245](#)
 - from journal I/O error [518](#)
 - ICF catalog, procedure [487](#)
 - or restoring RACF-indicated data sets [461](#)
 - problems that can occur
 - changes in the storage group [451](#)
 - changes in the storage-class-to-data set mapping [452](#)
 - conversions of data sets [452](#)
 - GUARANTEED SPACE data sets [451](#)
 - RACF discrete profiles [461](#)
 - specific data sets, non-SMS-managed [311](#)
- RECOVERNEWNAMEALL parameter, ARECOVER command [395](#)
- RECOVERNEWNAMELEVEL parameter, ARECOVER command [395](#)
- recovery
 - aggregate [391](#)
 - command [240](#)
 - control data sets [507](#)
 - data set [241](#)
 - date-of-scratch field [249](#)
 - direct access data sets [248](#)
 - examples [519](#)
 - examples, control data sets [519](#)
 - extended format data sets [248](#)
 - fast replication [270](#)
 - from DASD, RACF considerations [624](#)
 - incremental [300](#)

recovery (*continued*)

- journal from DASD, example [522](#)
- last-backup date [249](#)
- MCDS from DASD, example [519](#)
- MCDS from tape, example [524](#)
- prevent processing [459](#)
- procedures
 - CDS function [508](#)
 - control data set [507](#)
 - DASD backup data set support [499](#)
 - multiple backup data sets for CDS and journal [497](#)
 - multiple tape backup data set support [501](#)
- profile
 - accessing a data set when DFSMSHsm cannot recover a discrete profile [462](#)
 - authorization checking during [462](#)
 - effects of profile modeling on [462](#)
- SDSP [556](#)
- UNIX file [250](#)
- use of date last referenced field [459](#)
- volume [294](#)
- VSAM data set [314](#)

RECOVERY parameter

- HOLD command [599](#)
- RELEASE command [603](#)

recycle

- eligibility [540](#)
- process, defining parameters [537](#)
- process, initiating [539](#)
- selecting tape volumes during [541](#)
- volume contention
 - multiple-processing-unit environment [139](#)
 - single-processing-unit environment [138](#)
- tape [136](#)

RECYCLE

- optional parameters
 - FULLDSNAME [540](#)
 - LIMIT [539](#)
 - OUTDATASET [540](#)
 - PERCENTVALID [539](#)
 - PREFIX [540](#)
 - SELECT [539](#)
 - TAPELIST [540](#)
- required parameters
 - ALL [539](#)
 - BACKUP [539](#)
 - DAILY [539](#)
 - DISPLAY [539](#), [540](#)
 - EXECUTE [539](#)
 - ML2 [539](#)
 - SPILL [539](#)
 - VERIFY [540](#)
 - VOLUME [539](#)

RECYCLE command

- examples [1025](#)
- optional parameters
 - CHECKFIRST [1019](#)
 - FORCE [1019](#)
 - FULLDSNAME [1022](#)
 - LIMIT [1020](#)
 - OUTDATASET [1020](#)
 - PERCENTVALID [1021](#)
 - PREFIX [1022](#)
 - PULLSIZE [1022](#)

RECYCLE command (*continued*)

optional parameters (*continued*)

- SELECT [1022](#)
- TOTAL [1022](#)
- overview [1015](#)
- required parameters
 - ALL [1017](#)
 - DISPLAY [1018](#)
 - EXECUTE [1018](#)
 - SPILL [1017](#)
 - TAPELIST [1016](#), [1022](#)
 - VERIFY [1018](#)

syntax

- generic [1016](#)
- specific [1016](#)

RECYCLE parameter

- HOLD command [600](#)

- RELEASE command [604](#)

RECYCLEOUTPUT parameter, SETSYS command [538](#)

RECYCLEPERCENT parameter, SETSYS command [538](#)

RECYCLETAPEALLOCATION parameter, SETSYS command [539](#)

redefining the backup cycle [446](#)

reel-type tapes, making into cartridge-type single-file-format tapes [542](#)

relationship of

- frequency and maximum number of backup versions [181](#)
- NUMBER OF BACKUP VERSIONS and RETAIN DAYS attributes [182](#)

RELEASE

BACKUP or ALL, interrupting and restarting backup processing [486](#)

DUMP or ALL, interrupting and restarting dump processing [490](#)

held

- backup processing [449](#)
- dump processing [456](#)
- fast replication processing [283](#)
- recovery processing [459](#)
- restore processing [459](#)

optional parameters

- ABACKUP [600](#)
- ALL [600](#)
- ARECOVER [600](#)
- AUDIT [600](#)
- BACKUP [600](#)
- COMMONQUEUE [601](#)
- DUMP [601](#)
- EXPIREBV [602](#)
- FRBACKUP [283](#)
- FRRECOV [283](#)
- HARDCOPY [602](#)
- LIST [602](#)
- LOG [602](#)
- MIGRATION [602](#)
- RECALL [603](#)
- RECOVER [603](#)
- RECYCLE [604](#)
- REPORT [604](#)
- TAPE [603](#)
- TAPEDATASET [603](#)
- TSO [603](#)

RELEASE command

RELEASE command (*continued*)

examples [1039](#)

optional parameters

ABACKUP [1031](#)

ALL [1031](#)

ARECOVER [1031](#)

AUDIT [1032](#)

AUTOEXPIRATION [1032](#)

AUTOMIGRATION [1032](#)

BACKUP [1032](#)

BACKUP(DSCOMMAND) [1032](#)

COMMONQUEUE [1033](#)

DUMP [1034](#)

EXPIREBV [1035](#)

FRBACKUP [1034](#), [1035](#)

FRRECOV [1035](#)

HARDCOPY [1036](#)

LIST [1036](#)

LOG [1036](#)

MIGRATION [1037](#)

RECALL [1037](#)

RECOVER [1038](#)

RECYCLE [1038](#)

REPORT [1038](#)

RMM [1039](#)

TAPECOPY [1039](#)

TAPEREP [1039](#)

syntax [1030](#)

releasing

automatic hold [603](#)

backup volume [220](#)

held ABARS functions [409](#)

held backup processing [449](#)

held dump processing [456](#)

held fast replication processing [283](#)

held recovery processing [459](#)

held restore processing [459](#)

unused space, SMS-managed storage [69](#)

removal of

ML1 volume that contains an SDSP [557](#)

residual dump and backup VTOC copy data sets from

ML1 volumes [535](#)

SDSPs from a ML1 volume [557](#)

volumes from DFSMSHsm management and control [536](#)

RENAMESOURCE subparameter, DATASETCONFLICT

parameter, ARECOVER command [394](#), [402](#)

RENAMETARGET subparameter, DATASETCONFLICT

parameter, ARECOVER command [394](#), [402](#)

REPLACE parameter

ARECOVER command [394](#), [402](#)

RECOVER command [241](#), [252](#)

REPORT [574](#)

REPORT command

examples [1051](#)

optional parameters

DAILY [1045](#)

DELETE [1046](#)

FROMDATE [1046](#)

FUNCTION [1046](#)

FUNCTION DELETE [1047](#)

FUNCTION MIGRATION [1047](#)

FUNCTION RECALL [1048](#)

FUNCTION RECOVER [1048](#)

FUNCTION RECYCLE [1048](#)

REPORT command (*continued*)

optional parameters (*continued*)

FUNCTION SPILL [1049](#)

FUNCTION TRANSITION [1049](#)

NOFUNCTION [1046](#)

OUTDATASET [1049](#)

SUMMARY [1050](#)

SYSOUT [1049](#)

TODATE [1050](#)

VOLUMES [1050](#)

output

REPORT DAILY [1401](#)

REPORT VOLUME [1405](#)

syntax [1044](#)

using [1401](#)

report generator

running new reports [575](#)

REPORT parameter

AUDIT command [517](#)

HOLD command [600](#)

RELEASE command [604](#)

reports from

AUDIT BACKUPCONTROLDATASET [1291](#)

AUDIT BACKUPTYPE [1286](#)

AUDIT BACKUPVOLUMES [1286](#)

AUDIT DATASET NAMES [1282](#)

AUDIT MASTERCATALOG [1288](#)

AUDIT MIGRATIONCONTROLDATASET [1290](#)

AUDIT OFFLINECONTROLDATASET [1292](#)

AUDIT USERCATALOG [1288](#)

AUDIT VOLUMES [1284](#)

LIST AGGREGATE [1341](#)

LIST BACKUPVOLUMES [1345](#)

LIST DATASETNAME

BCDS, from the [1355](#)

latest backup version, for the [1360](#)

MCDS and BCDS, from the [1359](#)

MCDS, from the [1352](#)

LIST DUMPCLASS [1361](#)

LIST DUMPVOLUME

dump volumes, for [1366](#)

LIST HOST [1381](#)

LIST LEVEL

BCDS, from the [1355](#)

latest backup version, for the [1360](#)

MCDS and BCDS, from the [1359](#)

MCDS, from the [1352](#)

LIST MIGRATIONVOLUME

BCDS, from the [1376](#), [1378](#)

MCDS, from the [1371](#)

VTOC, contents of the [1379](#)

LIST PRIMARYVOLUME

BCDS, from the [1376](#), [1378](#)

MCDS, from the [1371](#)

VTOC, contents of the [1379](#)

LIST TAPETABLEOFCONTENTS [1381](#)

LIST USER [1386](#)

LIST VOLUME

BCDS, from the [1376](#), [1378](#)

MCDS, from the [1371](#)

VTOC, contents of the [1379](#)

requirements

for using enhanced CDS recovery function [508](#)

- RESET subparameter, DUMPCLASS parameter, DEFINE command [173](#)
- restart data set [373](#), [399](#)
- RESTART keyword [614](#)
- restarting
 - after interrupted automatic dump processing [489](#)
 - after interrupted backup processing [485](#)
 - automatic primary space management when the current time is not in the process window [163](#)
 - automatic secondary space management when the current time is not in the process window [163](#)
 - DFSMSHsm after an abnormal end [614](#)
 - primary and secondary space management
 - automatic [161](#)
 - command [161](#)
- restore
 - command [240](#)
 - data sets or volumes with DFSMSdss [458](#)
 - prevent processing [459](#)
 - VSAM data sets [253](#)
- restoring
 - a complete volume from a dump copy [298](#)
 - a non-VSAM data set from a dump copy [247](#)
 - a VSAM data set from a dump copy [247](#)
 - individual data sets from dump tapes, controlling [172](#)
- restricted command authority
 - ABACKUP [341](#)
 - ARECOVER [377](#)
- RESUME parameter, EXPIREBV command [233](#)
- RETAIN DAYS EXTRA BACKUP VERSIONS
 - attribute [181](#)
 - relationship to NUMBER OF BACKUP VERSIONS [182](#)
- RETAIN DAYS ONLY BACKUP VERSIONS
 - attribute [181](#)
 - relationship to NUMBER OF BACKUP VERSIONS [182](#)
- RETAIN DAYS
 - parameters [183](#), [769](#)
- retained backup copies
 - creation [184](#)
- RETENTIONPERIOD parameter
 - BACKVOL command [222](#)
 - DEFINE command [174](#)
- retirement (delete if backed up)
 - definition [114](#)
 - handling of retired versions
 - creating [115](#)
 - handling during EXPIREBV [310](#)
 - handling during spill processing [216](#)
 - recovery [315](#)
 - restriction to non-SMS-managed volumes [99](#)
- retrieval of
 - list of management class definitions, EXPIREBV processing [234](#)
 - management classes, availability management [464](#)
 - management classes, space management [154](#)
 - storage groups, availability management [464](#)
 - storage groups, space management [154](#)
- return codes
 - ARCBJRNL [512](#)
 - ARCIMPRT [512](#)
- returning dump tapes to use [555](#)
- reuse of dump tapes, controlling automatic [171](#)
- RLS mode, accessing CDSs with [504](#)

- RLS user catalogs
 - backing up and recovering [488](#)
- ROLLED-OFF GDS ACTION attribute [40](#)
- running
 - automatic functions concurrently [615](#)

S

- SAF (system authorization facility) [617](#)
- SCRATCHFREQUENCY parameter, SETSYS command [101](#)
- SDSP (small data set packing)
 - backup and recovery [556](#)
 - data sets
 - contention [142](#)
 - format for name [556](#)
 - managing [555](#)
 - multitasking considerations [147](#)
 - reorganizing [556](#)
 - security [625](#)
 - removing [557](#)
 - selection [77](#)
 - specifying [52](#)
 - searching for eligible dump copies, recovery processing [245](#)
- SECLABEL CLASS [617](#)
- second-level migration data set exit ARCMEXT [63](#), [90](#)
- secondary address space, maximum
 - specifying frequency of backup, SMS-managed [180](#)
 - specifying GDG management attributes [40](#)
 - specifying migration tasks [51](#)
 - specifying space requirements [32](#)
- secondary host promotion
 - automatic CDS backup [493](#)
 - backup processing [199](#)
 - dump processing [195](#)
 - during DFSMSHsm stop [596](#)
 - for SSM hosts [59](#), [164](#)
- secondary space management
 - automatic restarts of [162](#)
 - concept [24](#)
 - defining a processing window [164](#)
 - manual restarts of [163](#)
- security
 - checking for HALTERDS command [309](#)
 - checking with discrete profiles [309](#)
 - considerations for automatic space management [149](#)
 - considerations for availability management [460](#)
 - small data set packing [625](#)
 - tape
 - when using DFSMSHsm with DFSMSrmm [592](#)
- SELECT parameter
 - LIST command [560](#)
 - RECYCLE command [539](#)
- selection
 - backup volumes [216](#)
 - DASD and tape daily backup volumes [212](#)
 - initial DASD and tape daily backup volumes [213](#)
 - initial spill backup volumes [217](#)
 - messages to write to ABARS activity logs [344](#)
 - SDSP for migration [77](#)
 - subsequent DASD and tape daily backup volumes [215](#)
 - subsequent spill backup volumes [218](#)
 - tape volumes during recycling [541](#)
 - volume
 - daily backup [207](#)

selection (*continued*)

volume (*continued*)

- DASD daily backup [209](#)
- initial DASD daily backup [209](#)
- initial tape daily backup [211](#)
- subsequent DASD daily backup [211](#)
- subsequent tape daily backup [212](#)
- tape daily backup [211](#)
- tape dump [198](#)
- target, incremental backup version recovery processing [242](#), [251](#)
- target, physical dump copy restore [242](#)

selection data set

- ACCOMPANY data set [323](#)
- ACCOMPANYEXCLUDE data set [323](#)
- ALLOCATE data set [323](#)
- ALLOCATEEXCLUDE data set [323](#)
- creating [329](#)
- examples [334](#)
- EXCLUDE data set [323](#)
- INCLUDE data set [323](#)
- keyword descriptions [331](#)
- keyword syntax [329](#)
- overview [322](#)

sending alternate tapes off site [441](#)

sequential data striping

- aggregate backup processing considerations [431](#)
- aggregate recovery processing considerations [432](#)
- automatic space management processing considerations [155](#)
- backup processing considerations [464](#)
- recall processing considerations [155](#)
- volume space management processing considerations [155](#)

serialization of

- dump volume [197](#)
- processor information, obtaining [565](#)

SETMIG

- using to control migration of non-SMS-managed data sets and volumes [124](#)

SETMIG command

- examples [1055](#)
- optional parameters
 - COMMANDMIGRATION [1054](#)
 - MIGRATION [1054](#)
 - NOMIGRATION [1054](#)

overview [1053](#)

required parameters

- DATASETNAME [1054](#)
- LEVEL [1054](#)
- VOLUME [1054](#)

syntax [1053](#)

SETSYS command

- ANYSTORAGEVOLUME parameter, pool configuration [106](#)
- BACKUP, interrupting and restarting backup processing [486](#)
- BACKUP, interrupting and restarting dump processing [490](#)
- BACKUP, specifying handling of data sets INUSE [191](#)
- CDSVERSIONBACKUP, multiple backup data sets, recovering the CDSs using UPDATEC [513](#)
- defining aggregate backup options [343](#)
- DFSMSshm defaults [1084](#)

SETSYS command (*continued*)

examples [1211](#)

examples for aggregate backup [344](#)

fast-path [1075](#)

JES3, defining a JES3 system [106](#)

NOEMERGENCY, interrupting and restarting backup processing [486](#)

NOEMERGENCY, interrupting and restarting dump processing [490](#)

optional parameters

- ABARSACTLOGMSGVLV [344](#), [1094](#)
- ABARSACTLOGTYPE [1094](#)
- ABARSBUFFERS [344](#), [1095](#)
- ABARSDELETEACTIVITY [353](#), [1095](#)
- ABARSKIP [1095](#)
- ABARSOPTIMIZE [344](#), [1096](#)
- ABARSPROCNAME [343](#), [384](#), [1096](#)
- ABARSTAPES [344](#), [385](#), [1096](#)
- ABARSUNITNAME [1097](#)
- ABARSVOLCOUNT [1097](#)
- ACCEPTPSCBUSERID [1098](#)
- ACTLOGMSGVLV [1098](#)
- ACTLOGTYPE [343](#), [1099](#)
- ANYSTORAGEVOLUME [101](#)
- ARECOVERML2UNIT [1100](#)
- ARECOVERPERCENTUTILIZED [385](#), [1100](#)
- ARECOVERGTGDS [386](#), [1100](#)
- ARECOVERUNITNAME [385](#), [1100](#)
- AUTOBACKUPSTART [188](#), [1101](#)
- AUTODUMPSTART [1102](#)
- AUTOMIGRATIONSTART [1103](#)
- BACKUP [189](#), [1104](#)
- BACKUPCOPIES [497](#)
- BACKUPDEVICECATEGORY [498](#)
- BACKUPPREFIX [189](#), [1107](#)
- BCDSBACKUPDSN [497](#)
- CDSVERSIONBACKUP [493](#), [1109](#)
- CDSVERSIONBACKUP BACKUPCOPIES [1110](#)
- CDSVERSIONBACKUP BACKUPDEVICECATEGORY [1110](#)
- CDSVERSIONBACKUP BCDSBACKUPDSN [1112](#)
- CDSVERSIONBACKUP DATAMOVER [1112](#)
- CDSVERSIONBACKUP JRNLBACKUPDSN [1114](#)
- CDSVERSIONBACKUP MCDSBACKUPDSN [1113](#)
- CDSVERSIONBACKUP OCDSBACKUPDSN [1113](#)
- CHANGEDONLY [191](#)
- CLASSTRANSITIONS [1114](#)
- CLOUD [1114](#), [1115](#)
- COMMONQUEUE [131–133](#), [1117](#), [1119](#)
- COMPACT [1122](#)
- COMPACT ALL [1122](#)
- COMPACT DASDBACKUP [1123](#)
- COMPACT DASDMIGRATE [1123](#)
- COMPACT NODASDBACKUP [1123](#)
- COMPACT NODASDMIGRATE [1123](#)
- COMPACT NOTAPEBACKUP [1124](#)
- COMPACT NOTAPEMIGRATE [1124](#)
- COMPACT TAPEBACKUP [1124](#)
- COMPACT TAPERMIGRATE [1124](#)
- COMPACTPERCENT [1125](#)
- CONNECT [1117](#), [1120](#)
- CONVERSION [1125](#)
- CSALIMITS [1126](#)
- CSALIMITS ACTIVE [1126](#)

SETSYS command *(continued)*

optional parameters *(continued)*

[CSALIMITS INACTIVE 1127](#)
[CSALIMITS MAXIMUM 1127](#)
[CSALIMITS MWE 1127](#)
[DAYS 101, 1128](#)
[DEBUG 587, 1128](#)
[DEFERMOUNT 1129](#)
[DENSITY 1110, 1129](#)
[DISASTERMODE 1130](#)
[DISCONNECT 1118, 1121](#)
[DSBACKUP 1130](#)
[DSBACKUP \(DASD\) 1130](#)
[DSBACKUP \(DASDSELECTIONSIZE\) 1130](#)
[DSBACKUP \(TAPE \(DEMOUNTDELAY \(MAXIDLETASKS\)\)\) 1130](#)
[DSBACKUP \(TAPE \(DEMOUNTDELAY \(MINUTES\)\)\) 1130](#)
[DSBACKUP \(TAPE \(DEMOUNTDELAY\)\) 1130](#)
[DSBACKUP \(TAPE\) 1130](#)
[DSSXMMODE 1133](#)
[DUMPIO 168, 1134](#)
[DUPLEX 436, 1135](#)
[EMERGENCY 1136](#)
[ERASEONSCRATCH 1137](#)
[EXITOFF 343, 384, 1138](#)
[EXITON 343, 384, 1139](#)
[EXITS 1141](#)
[EXPIREDDATASETS 49, 1141](#)
[FASTREPLICATION 1142](#)
[FREQUENCY 303, 1143](#)
[HSERIALIZATION 1129](#)
[INCREMENTALBACKUP 190, 1144](#)
[INPUTTAPEALLOCATION 1145](#)
[INTERVALMIGRATION 1145](#)
[ISSUEARC1902 1146](#)
[JES2 1147](#)
[JES3 1147](#)
[JOURNAL 1147](#)
[JRNLBKUPDSN 497](#)
[MASTERSCHEDULERCANDIDATE 1118, 1121](#)
[MAXABARSADDRESSSPACE 343, 384, 1148](#)
[MAXBACKUPTASKS 190, 1149](#)
[MAXCOPYPOOLTASKS 1149](#)
[MAXDSRECOVERTASKS 1150](#)
[MAXDUMPRECOVERTASKS 1151](#)
[MAXDUMPTASKS 169, 1151](#)
[MAXEXTENTS 49, 1152](#)
[MAXINTERVALTASKS 51, 1152](#)
[MAXMIGRATIONTASKS 49, 1153](#)
[MAXRECALLTASKS 52, 1153](#)
[MAXRECYCLETASKS 538, 1154](#)
[MAXSSMTASKS 50, 1154](#)
[MCDSBACKUPDSN 497](#)
[MIGDENSITY 1155](#)
[MIGRATEPREFIX 53, 1155](#)
[MIGRATEPSMEARLY 58, 1158](#)
[MIGRATIONAUTOCLD 58, 1156](#)
[MIGRATIONCLEANUPDAYS 54, 1157](#)
[MIGRATIONLEVEL1DAYS 101, 1158](#)
[MIGRATIONSUBTASKS 1159](#)
[MIGUNITNAME 1159](#)
[ML1OVERFLOW 1160](#)
[ML2PARTIALSNOTASSOCIATEDGOAL 141](#)

SETSYS command *(continued)*

optional parameters *(continued)*

[ML2RECYCLEPERCENT 538, 1161](#)
[ML2TAPE 48](#)
[MONITOR 1162](#)
[MONITOR BACKUPCONTROLDATASET 1163](#)
[MONITOR JOURNAL 1163](#)
[MONITOR MIGRATIONCONTROLDATASET 1163](#)
[MONITOR NOSPACE 1164](#)
[MONITOR NOSTARTUP 1164](#)
[MONITOR NOVOLUME 1165](#)
[MONITOR OFFLINECONTROLDATASET 1164](#)
[MONITOR SPACE 1164](#)
[MONITOR STARTUP 1164](#)
[MONITOR VOLUME 1165](#)
[MOUNTWAITTIME 1165](#)
[NOACCEPTPSCBUSERID 1098](#)
[NOBACKUP 1104](#)
[NOCDSVERSIONBACKUP 502](#)
[NOCONVERSION 1125](#)
[NOCALIMITS 1126](#)
[NODEBUG 1128](#)
[NOEMERGENCY 1136](#)
[NOERASEONSCRATCH 1137](#)
[NOINTERVALMIGRATION 1145](#)
[NOJOURNAL 1147](#)
[NONE 48, 1122, 1208](#)
[NONE\(ROUTETOTAPE\) 48](#)
[NOOPTIMUMDASDBLOCKING 1167](#)
[NOPROFILEBACKUP 1172](#)
[NORACFIND 1173](#)
[NOREQUEST 1179](#)
[NOSKIPABPRIMARY 1183](#)
[NOSMALLDATASETPACKING 1183](#)
[NOSMF 1185](#)
[NOSPILL 219, 1186](#)
[NOSWAP 1187](#)
[NOSYS1DUMP 1188](#)
[NOTAPEHARDWARECOMPACT 1190](#)
[NOUSERUNITTABLE 1202](#)
[OBJECTNAMES 1166](#)
[OCDSBACKUPDSN 497](#)
[ODMNOTIFICATIONLIMIT 1167](#)
[ONDEMANDMIGRATION 1166](#)
[OPTIMUMDASDBLOCKING 1167](#)
[ORIGINAL 190](#)
[OUTPUTTAPEALLOCATION 1168](#)
[PARTIALTAPE 1168](#)
[PDA 586, 1169](#)
[PLEXNAME 1170](#)
[PRIMARYSPMGMTSTART 46, 1171](#)
[PRIVATEVOLUME 102](#)
[PROFILEBACKUP 190, 1172](#)
[PROMOTE 1172](#)
[RACFIND 1173](#)
[RECALL 101, 1173](#)
[RECYCLEINPUTDEALLOCFREQUENCY 1174](#)
[RECYCLEOUTPUT 538, 1176](#)
[RECYCLEPERCENT 538, 1176](#)
[RECYCLETAKEAWAYRETRY 1177](#)
[RECYCLETAPEALLOCATION 539, 1177](#)
[RECYCLETAPERESIDENCE 1178](#)
[REMOVECOMPACTNAMES 1178](#)
[REQUEST 1179](#)

SETSYS command (*continued*)

optional parameters (*continued*)

- SCRATCHFREQUENCY [101](#), [1179](#)
- SECONDARYSPMGMTDIRECTIVE [1179](#)
- SECONDARYSPMGMTSTART [46](#), [1180](#)
- SELECTVOLUME [1181](#)
- SKIPABPRIMARY [1183](#)
- SMALLDATASETPACKING [52](#), [1183](#)
- SMF [1185](#)
- SOURCENAMES [1185](#)
- SPILL [190](#), [1186](#)
- STORAGEGROUPAFFINITY [58](#), [1187](#)
- SWAP [1187](#)
- SYS1DUMP [1188](#)
- SYSOUT [1099](#), [1187](#)
- TAPE [189](#)
- TAPEDATASETORDER [1188](#)
- TAPEDELETION [1189](#)
- TAPEHARDWARECOMPACT [1190](#)
- TAPEINPUTPROMPT [1191](#)
- TAPEMAXRECALLTASKS [52](#), [1192](#)
- TAPEMIGRATION [1192](#)
- TAPEOUTPUTPROMPT [1194](#)
- TAPERECALLLIMITS [1195](#)
- TAPESECURITY [1195](#)
- TAPESPANSIZE [1197](#)
- TAPE TYPE [1198](#)
- TAPEUTILIZATION [1198](#)
- UNITNAME [1110](#), [1201](#)
- USECYLINDERMANAGEDSPACE [1202](#)
- USERSERIALIZATION [1129](#)
- USERUNITTABLE [1202](#)
- VERSIONS [303](#), [1205](#)
- VOLCOUNT [1206](#)
- VOLUMEDUMP [1206](#)
- ZCOMPRESS [1207](#)
- ZCOMPRESS ALL [1208](#)
- ZCOMPRESS CLOUDMIGRATE [1208](#)
- ZCOMPRESS DASDBACKUP [1209](#)
- ZCOMPRESS DASDMIGRATE [1210](#)
- ZCOMPRESS TAPEBACKUP [1210](#)
- ZCOMPRESS TAPEMIGRATE [1211](#)

overview [1057](#)

PRIVATESTORAGEVOLUME, pool configuration in JES3 [106](#)

SMS relationship [1082](#)

UNLOAD, releasing a backup volume [220](#)

SETSYS DUPLEX command

ERRORALTERNATE parameter [437](#)

SETSYS FASTREPLICATION command

using [276](#)

SETSYS MAXDSRECOVERTASKS command

description [1150](#)

SETSYS MAXDSRECOVERTASKS(0) command

using [281](#)

SETSYS MAXDSTAPERECOVERTASKS command

description [1150](#)

shortcut keys [1413](#)

shutdown, orderly [596](#)

single-processing-unit-environment volume contention [137](#)

size

eligibility for data sets to migrate to an SDSP [76](#)

of data sets eligible for small data set packing [52](#)

SMALLDATASETPACKING

SMALLDATASETPACKING (*continued*)

parameter [52](#)

SMS (Storage Management Subsystem)

purpose [7](#)

SMS class definitions, displaying during ABACKUP and

ARECOVER [409](#)

SMS-managed

attributes [378](#)

automatic recall [83](#)

backup frequency [180](#)

class names [378](#)

data sets

command recall of [92](#)

processing, migrating all from ML1 volume [533](#)

recalling BDAM to non-SMS-managed volume [125](#)

recalling extended format to non-SMS-managed volume [125](#)

recalling to non-SMS-managed volume [125](#)

guaranteed frequency of backup [180](#)

requirements

for aggregate backup [328](#)

for aggregate recovery [378](#)

storage

availability management of [165](#)

space management of [23](#)

volumes

command migration of [91](#)

functions for [10](#)

recovering data sets to [245](#)

transitioning of [91](#)

sort candidate data sets into priority order, non-SMS-

managed data sets [113](#)

SOURCELEVEL subparameter, MIGRATEDDATA parameter,

ARECOVER command [395](#)

space management

Access Method Services considerations [152](#)

automatic processing for non-SMS-managed storage [109](#)

catalog considerations [78](#)

command

all primary volumes, non-SMS-managed [124](#)

functions supported [88](#)

non-SMS-managed [123](#)

concept [23](#)

daily

defining a new start window [487](#)

data sets with discrete data set profiles [150](#)

date-last-referenced field, maintenance and use [78](#)

deleting

expired data sets, non-SMS-managed volumes [111](#)

list and utility data sets, non-SMS-managed volumes [111](#)

dynamic volume expansion [32](#)

end of level 0 volume processing [79](#)

example differences for non-SMS-managed storage [99](#)

extent reduction, non-SMS-managed data sets [117](#)

functions

overview, non-SMS-managed

[16](#)

overview, SMS-managed [13](#)

integrity age [56](#)

interrupting

and restarting primary [161](#)

- space management (*continued*)
 - interrupting (*continued*)
 - and restarting secondary [161](#)
 - JES3 considerations, non-SMS-managed volumes [116](#)
 - maintenance and use of the date-last-referenced field [78](#)
 - manual restarts [162](#)
 - migration
 - interval [25](#)
 - on-demand [27](#)
 - migration, interval
 - disabling [25](#)
 - enabling [25](#)
 - notes, usage [26](#)
 - other considerations [26](#)
 - requirements for [25](#)
 - scenario [81](#)
 - migration, on-demand
 - disabling [28](#)
 - enabling [28](#)
 - multiple DFSMSHsm host environment, in a [29](#)
 - notes, usage [29](#)
 - other considerations [29](#)
 - requirements for [28](#)
 - scenario [81](#)
 - non-SMS-managed
 - volumes
 - pass 1 [111](#)
 - pass 2 [113](#)
 - of non-SMS-managed storage [99](#)
 - of SMS-managed storage [23](#)
 - of VSAM data sets [75](#)
 - other considerations [127](#)
 - performing on non-SMS-managed volumes [113](#)
 - preventing processing of data sets [149](#)
 - primary
 - automatic restarts of [162](#)
 - manual restarts of [163](#)
 - processing window [164](#)
 - scenario [81](#)
 - volumes, non-SMS-managed [111](#)
 - volumes, pass 1 [111](#)
 - volumes, pass 2 [113](#)
 - priority order, non-SMS-managed data sets [113](#)
 - procedures [161](#)
 - RACF authority required [150](#)
 - recall [32](#)
 - restarting
 - primary [161](#)
 - secondary [161](#)
 - when the current time is not in the process window [163](#)
 - secondary
 - automatic restarts of [162](#)
 - manual restarts of [163](#)
 - processing window [164](#)
 - specifying the number of tasks [50](#)
 - security considerations [149](#)
 - specific tapes subsequent volume selection management processing [215](#)
 - subtasks for non-SMS-managed storage [99](#)
 - volume, command, non-SMS-managed [124](#)
 - VSAM data sets, non-SMS-managed volumes [115](#)

- space management (*continued*)
 - when primary needs to be restarted [161](#)
 - when secondary needs to be restarted [161](#)
 - where it restarts [163](#)
 - writing to a cartridge=type single-file tape [78](#)
 - space management and backup exit ARCSAEXT [66](#)
 - space requirements, specifying [32](#)
 - specific data sets, delete backup versions [239](#)
 - specifying age for deleting list data sets [100](#)
 - specifying age for deleting utility data sets [100](#)
 - specifying automatic interval space management tasks [51](#)
 - specifying automatic primary space management tasks [49](#)
 - specifying automatic secondary space management tasks [50](#)
 - specifying DASD for CDS backup [498](#)
 - specifying device type for backup of the CDSs [498](#)
 - specifying DFSMSdss as data mover for CDS backup [494](#)
 - specifying DFSMSHsm as data mover for CDS backup [496](#)
 - specifying first qualifier for names of backup versions [189](#)
 - specifying first qualifier for names of migrated data sets [53](#)
 - specifying maximum number of backup tasks in each processor [190](#)
 - specifying maximum number of dump copies to stack on a volume [175](#)
 - specifying maximum number of dump tasks [169](#)
 - specifying migration attributes [37](#)
 - specifying migration tasks [49](#)
 - specifying recall characteristics for non-SMS-managed volumes [101](#)
 - specifying recall tasks [52](#)
 - specifying tape for CDS backup [498](#)
 - specifying volumes to be dumped
 - and dump classes to use, SMS-managed [166](#)
 - non-SMS-managed [305](#)
 - spill
 - backup volume
 - DASD, subsequent spill volume selection [218](#)
 - moving DASD backup volumes to spill volumes [535](#)
 - none available during initial spill volume selection [217](#)
 - tape, subsequent spill volume selection [218](#)
 - initial backup volume selection [217](#)
 - processing [215](#)
 - processing, specifying whether to do [190](#)
 - subsequent backup volume selection [218](#)
 - SPILL parameter
 - RECYCLE command [539](#)
 - SETSYS command [190](#)
 - STACK parameter
 - ABACKUP command [348](#)
 - ARECOVER DATASETNAME command [394](#)
 - DEFINE command [169, 175](#)
 - SETSYS ABARSTAPES command [344](#)
 - stacking multiple dump copies on a tape [169, 175](#)
 - standard interval migration compared to TMM [30](#)
 - standard management class, space management [41](#)
 - start window
 - automatic primary space management, defining a new [164](#)
 - automatic secondary space management, defining a new [164](#)
 - when automatic primary space management is not running [164](#)
 - when automatic secondary space management is not running [164](#)

- starter set [8](#)
- starting
 - automatic primary space management [64](#)
 - automatic secondary space management [59](#)
 - DFSMSHsm [594](#)
- STARTKEY parameter, EXPIREBV command [233](#)
- startup
 - procedure
 - HOST parameter [595](#)
 - multiple backup data sets and CDS recovery using UPDATEC [513](#)
 - window, defining for automatic dump [491](#)
- statistical information about DFSMSHsm operations, obtaining [574](#)
- STOP command
 - examples [1224](#)
 - for DFSMSHsm [1223](#)
 - optional parameters
 - DUMP [1223](#)
 - PROMOTE [1224](#)
- stop DFSMSHsm [595](#)
- stopping
 - data set processing [904](#)
 - recording in the log [907](#)
 - report function [909](#)
 - volume processing [903](#), [904](#)
- storage
 - groups
 - ALLOCATION/MIGRATION THRESHOLD attribute [32](#)
 - AUTO BACKUP attribute [178](#)
 - AUTO DUMP attribute [166](#)
 - AUTO MIGRATE attribute [32](#)
 - backing up all eligible changed data sets to one or more of [232](#)
 - backing up data sets to one or more of [230](#)
 - BACKUP SYSTEM/SYSPLEX NAME attribute [178](#)
 - control of automatic dump [166](#)
 - control of dump classes [166](#)
 - control of number of dump copies [166](#)
 - copy pool backup storage group type [259](#)
 - DUMP CLASS attribute [166](#)
 - DUMP SYSTEM/SYSPLEX NAME attribute [166](#)
 - dumping volumes to one or more of [222](#)
 - example, space management [35](#)
 - grouping data sets with [11](#)
 - GUARANTEED BACKUP FREQUENCY attribute [180](#)
 - MIGRATE SYSTEM/SYSPLEX NAME attribute [35](#)
 - retrieval of, availability management [464](#)
 - retrieval of, space management [154](#)
 - use of [11](#)
- storage administrator mode
 - for UNIX commands [1245](#)
- storage class, use of [11](#)
- storage group priority
 - use in autobackup processing [179](#)
- storage to be managed, specifying [32](#)
- STORAGEGROUPAFFINITY
 - parameter [58](#)
- subsequent selection
 - DASD daily backup volume [211](#)
 - DASD or tape daily backup volume [215](#)
 - DASD spill backup volume [218](#)
 - spill backup volume [218](#)
 - tape daily backup volume [212](#)
- subsequent selection (*continued*)
 - tape spill backup volume [218](#)
- subtasks
 - availability management of non-SMS-managed storage [303](#)
 - availability management of SMS-managed storage [166](#)
 - space management of non-SMS-managed storage [99](#)
 - space management of SMS-managed storage [32](#)
 - SYSOUT parameter
 - AUDIT command [517](#)
 - LIST command [559](#)
- summary of changes [lix](#), [lx](#)
- SWAPLOG [607](#)
- SWAPLOG command
 - examples [1226](#)
 - for DFSMSHsm [1225](#)
 - optional parameters
 - LOG [1225](#)
 - PDA [1225](#), [1226](#)
- swapping log data sets [607](#)
- switching backup tapes [227](#)
- syntax diagrams
 - how to read [liv](#)
- sysplex name support [14](#)
- system
 - non-SMS-managed [17](#)
 - running under
 - JES2, pools [105](#)
 - JES3, pools [106](#)
 - SMS-managed [17](#)
- system conditions when an error occurs, obtaining information about [586](#)

T

- tape
 - and DASD daily backup volume, selecting [212](#)
 - backing up CDSs to multiple backup data sets [501](#)
 - backup overwritten [637](#)
 - changing from DASD to tape [504](#)
 - changing to DASD backup data sets [503](#)
 - choosing, for multiple backup data sets [498](#)
 - daily backup volume selection
 - initial [211](#)
 - subsequent [212](#)
 - data set exit ARCTDEXT [63](#), [90](#)
 - data set name
 - backup tape [206](#)
 - migration tape [74](#)
 - TAPECOPY of backup tape [206](#)
 - TAPECOPY of migration tape [75](#)
 - determining unit type for aggregate backup [345](#)
 - devices, example [19](#)
 - how to access single-file format data set [74](#)
 - migration level 2
 - partial, specifying number of [141](#)
 - volume contention [136](#)
 - migration overwritten [634](#)
 - name of backup copy data sets [206](#)
 - or DASD, subsequent daily backup volume selection [215](#)
 - security
 - when using DFSMSHsm with DFSMSrmm [592](#)
 - selection

- tape (*continued*)
 - selection (*continued*)
 - with duplex tape processing [436](#)
 - with TAPECOPY processing [438](#)
 - specifying for CDS backup [498](#)
 - subsequent selection, backup [215](#)
 - take away [228](#)
 - using HMIGTAPE.DATASET to access single-file format data sets [74](#)
 - volume
 - alternate, replacing an original with [440](#)
 - backup, determining which are available for output [573](#)
 - consolidating valid data [537](#)
 - contention [136](#)
 - determining which are available for output [573](#)
 - DFSMSHsm-owned, copying categories [438](#)
 - DFSMSHsm-owned, making backup copies [435](#)
 - DFSMSHsm-owned, making copies [436](#)
 - DFSMSHsm-owned, protecting [558](#)
 - dump, determining which are available for output [570](#)
 - dump, selecting [198](#)
 - exit, ARCTVEXT [439](#)
 - migration level 2, determining which are available for output [574](#)
 - obtaining information [569](#)
 - retaining, by DFSMSrmm vital record processing [591](#)
 - retaining, by expiration date [591](#)
 - selecting during recycling [541](#)
- tape copy needed (TCN) record [436](#)
- tape data set exit ARCTDEXT [63](#), [90](#)
- tape drive
 - 3592 Model E05 considerations [545](#)
 - 3592 Model E06 considerations [546](#)
 - 3592 Model E07 considerations [548](#)
 - 3592 Model J considerations [544](#)
- tape media contents and OCDS TTOC records, handling [550](#)
- tape mount management (TMM)
 - compared to standard interval migration [30](#)
 - used with interval migration [33](#)
- TAPE subparameter, RECALL parameter
 - HOLD command [599](#)
 - RELEASE command [603](#)
- tape units used for dump volumes, specifying [174](#)
- tape utilization
 - specifying [543](#)
- tape utilization capacity for media 3 or 4 [939](#), [1198](#), [1200](#)
- TAPECOPY
 - data set name of TAPECOPY of backup tape [206](#)
 - data set name of TAPECOPY of migration tape [75](#)
 - optional parameters
 - ALTERNATEVOLUMES [439](#)
 - required parameters
 - ALL [438](#)
 - BACKUP [438](#)
 - INDATASET [440](#)
 - MIGRATIONLEVEL2 [438](#)
 - take away by recall [137](#)
- TAPECOPY command
 - examples [1233](#)
 - optional parameters
 - ALTERNATE3590UNITNAME [1231](#)
- TAPECOPY command (*continued*)
 - optional parameters (*continued*)
 - ALTERNATEUNITNAME [1230](#)
 - ALTERNATEVOLUMES [1231](#)
 - EXPDT [1232](#)
 - RETPD [1232](#)
 - required parameters
 - ALL [1228](#)
 - BACKUP [1228](#)
 - INDATASET [1228](#)
 - MIGRATIONLEVEL2 [1228](#)
 - ORIGINALVOLUMES [1228](#)
 - syntax [1227](#)
 - TAPEDATASET subparameter, RECOVER parameter
 - HOLD command [599](#)
 - RELEASE command [603](#)
 - TAPEEXPIRATIONDATE subparameter, DUMPCCLASS parameter, DEFINE command [174](#)
 - TAPELIST parameter
 - RECYCLE command [540](#)
 - TAPEMAXRECALLTASKS parameter, SETSYS command [52](#)
 - TAPEREP
 - required parameters
 - ORIGINALVOLUMES [439](#)
 - TAPEREP command
 - examples [1238](#)
 - optional parameters
 - ALTERNATEUNITNAME [1237](#)
 - ALTERNATEVOLUME [1238](#)
 - DISASTERALTERNATEVOLUMES [1238](#)
 - overview [1235](#)
 - required parameters
 - ALL [1235](#)
 - BACKUP [1235](#)
 - INDATASET [1236](#)
 - MIGRATION [1236](#)
 - ONLYDISASTERALTERNATES [1237](#)
 - ORIGINALVOLUMES [1237](#)
 - syntax [1235](#)
 - tapetype WORM media for non-ABARS functions [1198](#)
 - TAPEUTILIZATION and PERCENTFULL
 - determining capacity [1201](#)
- target tier
 - specifying [55](#)
- TARGETUNIT parameter, ARECOVER command [385](#), [396](#)
- tasks
 - automatic
 - availability management, SMS-managed storage [166](#)
 - space management, SMS-managed storage [32](#)
 - availability management, non-SMS-managed storage [303](#)
 - backup [177](#)
 - backup, maximum number in each processor, specifying [190](#)
 - maintaining control data sets [493](#)
 - managing storage with DFSMSHsm [8](#)
 - prepare for recovery of a head crash
 - steps for [663](#)
 - preventing a full journal
 - steps for [661](#)
 - preventing overwritten migration tape (backup)
 - steps for [663](#)

- tasks (*continued*)
 - preventing structural damage to CDS and missing journal records
 - steps for [662](#)
 - preventing time sequence gaps in CDS records
 - steps for [662](#)
 - reestablishing access to previously deleted, migrated data sets
 - steps for [663](#)
 - space management, non-SMS-managed storage [99](#)
- TCN (tape copy needed) record [436](#)
- temporary data set
 - deleting [202](#)
 - deleting, SMS-managed storage [66](#)
- TERMINAL parameter, LIST command [559](#)
- TGTGDS parameter, ARECOVER command [386](#), [392](#), [396](#)
- threshold of occupancy, backup volume, definition [215](#)
- THRESHOLD parameter, ADDVOL command
 - backup volume cleanup process [220](#)
- thresholds, selecting for volumes [710](#)
- time sequence gaps in CDS records, journal records missing [633](#)
- TMM (tape mount management)
 - compared to standard interval migration [30](#)
 - used with interval migration [33](#)
- TOTAL parameter, BACKVOL command [230](#)
- TOVOLUME parameter, RECOVER command [252](#), [311](#), [314](#)
- tracing
 - DFSMSHsm processing [586](#)
- trademarks [1418](#)
- TRANSITION
 - parameter of MIGRATE command [963](#)
- transitioning
 - SMS-managed volumes [91](#)
 - storage group [92](#)
- TRANSITIONONLY
 - parameter of MIGRATE command [963](#)
- TRAP command
 - for DFSMSHsm [1241](#)
- TSO attention key [686](#)
- TSO subparameter, RECALL parameter
 - HOLD command [599](#)
 - RELEASE command [603](#)

U

- UID [488](#)
- UNASSIGN parameter, DELVOL command [536](#)
- uncataloged data sets, backing up [307](#)
- UNCATALOGEDDATA subparameter, NONSMSVERSIONS parameter, EXPIREBV command [310](#)
- uncataloging
 - excess dump VTOC copy data sets [198](#)
 - migrated data sets [625](#)
- undamaged CDS, damaged journal, [632](#)
- undamaged CDS, full journal, [632](#)
- UNIT parameter
 - ABACKUP command [347](#)
 - ARECOVER command [392](#)
 - BACKDS command [309](#)
 - RECOVER command [241](#), [252](#), [311](#), [314](#)
- UNIT parameter, ARECOVER command [394](#)
- UNIT subparameter, DUMPCLASS parameter, DEFINE command [174](#)

- UNIX files
 - changed
 - backing up all eligible in a directory [232](#)
- UNLOAD parameter, SETSYS command, releasing a backup volume [220](#)
- unmanaged volumes, example [19](#)
- unmatched data sets
 - backup [202](#)
 - space management [66](#)
- unused space, releasing, SMS-managed storage [69](#)
- UPDATEC
 - optional parameters
 - JOURNAL [516](#)
 - required parameters
 - ALL [516](#)
 - BACKUPCONTROLDATASET [516](#)
 - MIGRATIONCONTROLDATASET [516](#)
 - OFFLINECONTROLDATASET [516](#)
- UPDATEC command
 - examples [1244](#)
 - for DFSMSHsm [1243](#)
 - optional parameters
 - JOURNAL [1244](#)
 - required parameters
 - ALL [1243](#)
 - OFFLINECONTROLDATASET [1243](#)
- updating recovered CDSs [516](#)
- UPDTCDS [616](#)
- UPDTCDS command
 - for DFSMSHsm [1253](#)
 - required parameters
 - COPYPOOL [1253](#)
 - DUMPEXPIRATION [1253](#)
 - VERSION [1253](#)
- UPDTCDS ommand
 - examples [1254](#)
- use of
 - concurrent copy to back up the CDSs [494](#)
 - data class [11](#)
 - DFSMSHsm with DFSMSrmm [591](#)
 - JCL to implement enhanced CDS recovery function [508](#)
 - management classes [11](#)
 - non-intrusive journal backup [494](#), [496](#), [507](#), [525](#)
 - quiesced journal backup [496](#)
 - storage class [11](#)
 - storage groups [11](#)
- user authorization information, obtaining [570](#)
- user interface
 - ISPF [1413](#)
 - TSO/E [1413](#)
- using
 - AUDIT ABARSCONTROLS [1257–1259](#)
 - AUDIT DATASETCONTROLS (BACKUP) [1260](#)
 - AUDIT DATASETCONTROLS (MIGRATION) [1261](#)
 - AUDIT DIRECTORYCONTROLS [1263](#)
 - AUDIT FILECONTROLS(BACKUP) [1266](#)
 - AUDIT MEDIACONTROLS (CLOUD) for DASD [1267](#)
 - AUDIT MEDIACONTROLS (CLOUD) for TAPE [1267](#)
 - AUDIT MEDIACONTROLS (SMALLDATASETPACKING) [1268](#)
 - AUDIT MEDIACONTROLS (VOLUMES) for DASD [1269](#)
 - AUDIT MEDIACONTROLS (VOLUMES) for TAPE [1269](#)
 - AUDIT VOLUMECONTROLS (BACKUP) [1272](#)
 - AUDIT VOLUMECONTROLS (MIGRATION) [1273](#)

using (*continued*)

- AUDIT VOLUMECONTROLS (RECOVERABLE) [1274](#)
- LIST AGGREGATE [1341](#)
- LIST BACKUPVOLUMES [1345](#)
- LIST DATASETNAME [1352](#)
- LIST DUMPCLASS [1361](#)
- LIST DUMPVOLUME [1364](#)
- LIST HOST [1381](#)
- LIST LEVEL [1352](#)
- LIST MIGRATIONVOLUME [1371](#)
- LIST PRIMARYVOLUME [1371](#)
- LIST TAPETABLEOFCONTENTS [1381](#)
- LIST USER [1386](#)
- LIST VOLUME [1371](#)

using fast reverse restore [278](#)

using incremental FlashCopy [279](#)

utility data sets

- formats of names [100](#)

- specifying age for deleting [100](#)

V

valid data on tape volumes, consolidating [537](#)

varying dump invocation [457](#)

VCAT copy data sets [218](#)

verification

- aggregate backup [349](#)

- aggregate recovery [392](#), [398](#)

VERIFY parameter

- ABACKUP command [347](#)

- ARECOVER command [392](#)

- RECYCLE command [540](#)

VERSION parameter, ARECOVER command [394](#)

VERSIONS parameter

- BACKVOL command [310](#)

- BDELETE command [311](#)

- SETSYS command [303](#)

versions, backup, moving [200](#)

Viewing DFSMSHsm Facility Class Access by using the Security Configuration Assistant (SCA) for z/OS Management Facility(z/OSMF))

- viewing DFSMSHsm facility class access by using the security configuration assistant [621](#)

volume

- backup

- cleanup process [219](#)

- spill process [215](#)

- spill volume selection, initial [217](#)

- backup, defining to DFSMSHsm [192](#)

- command migration, SMS-managed [91](#)

- contention

- migration needs volume [137](#)

- multiple-processing-unit environment [139](#)

- recycle needs volume [138](#)

- single-processing-unit environment [137](#)

- tapes [136](#)

- controlling migration, non-SMS-managed [124](#)

- daily backup, selecting [207](#)

- disaster alternate [442](#)

- dump

- defining to DFSMSHsm [177](#)

- specifying concurrent copy with [169](#)

- specifying the kind of tape units to use [174](#)

- specifying what to do with [173](#)

volume (*continued*)

- dumping, non-SMS-managed [308](#)

- initial selection

- DASD daily backup [209](#)

- DASD or tape daily backup [213](#)

- spill backup [217](#)

- tape daily backup [211](#)

- making disaster backup copies [439](#)

- migration

- deleting non-SMS-managed data sets with the MIGRATE command [532](#)

- using the FREEVOL command for [533](#)

- migration level 1

- damaged, recovering backup versions from [554](#)

- moving all data sets [533](#)

- using the MIGRATE command [532](#)

- migration level 2, contention [136](#)

- non-SMS-managed

- automatic availability management of [306](#)

- backing up all primary [309](#)

- online, ensuring [343](#)

- pool

- configuration in JES3 [107](#)

- defining [105](#)

- pooling

- example [19](#)

- JES2 [105](#)

- JES3 [107](#)

- recovery, incremental backup [316](#)

- selecting

- backup [216](#)

- DASD and tape daily backup [212](#)

- DASD daily backup [209](#)

- recall, non-SMS-managed [119](#)

- tape daily backup [211](#)

- serialization, dump [197](#)

- source, backup [307](#)

- space management, preventing processing of data sets [149](#)

- subsequent selection

- DASD daily backup [211](#)

- DASD or tape daily backup [215](#)

- DASD spill [218](#)

- spill backup [218](#)

- tape daily backup [212](#)

- tape spill [218](#)

- tape

- alternate, replacing an original with [440](#)

- consolidating valid data [537](#)

- DFSMSHsm-owned, copying categories [438](#)

- DFSMSHsm-owned, making backup copies [435](#)

- DFSMSHsm-owned, making copies [436](#)

- DFSMSHsm-owned, protecting [558](#)

- dump, selecting [198](#)

- selecting during recycling [541](#)

- to be backed up, specifying [304](#)

- to be dumped, SMS-managed [166](#)

- to be dumped, specifying [305](#)

- with storage groups [11](#)

VOLUME parameter

- BACKDS command [309](#)

- BACKVOL command [222](#)

- MIGRATE command [532](#)

- RECYCLE command [539](#)

- volume, damage on an ML1 [641](#)
- VOLUMEPOOL parameter, DEFINE command [107](#)
- VOLUMES parameter, ARECOVER command [393](#)
- VSAM considerations for control data sets [516](#)
- VSAM data set
 - accessing during backup [204](#)
 - backing up [204](#)
 - GUARANTEED SPACE attribute
 - availability management [451](#)
 - recalling, changes in the storage group [153](#)
 - recalling, changes in the storage-class-to-data set mapping [153](#)
 - recalling, problems [153](#)
 - recalling, problems with conversions of data sets [154](#)
 - recovering [451](#)
 - recovering, changes in the storage group [451](#)
 - recovering, changes in the storage-class-to-data set mapping [452](#)
 - recovering, problems [451](#)
 - recovering, problems with conversions of data sets [452](#)
 - space management [152](#)
 - recovering [314](#)
 - recovering from a DFSMSHsm backup version [246](#)
 - restore [253](#)
 - restoring from a dump copy [247](#)
 - space management [75](#)
- VTOC
 - copy data set
 - backup, names [202](#)
 - create [202](#)
 - volume recovery [316](#)
 - dump copies
 - example [175](#)
 - specifying number to keep [175](#)
- VTOCCOPIES subparameter, DUMPCLASS parameter, DEFINE command [176](#)

W

- wait for a spill backup volume [217](#)
- what our example system shows, space management [23](#)
- who can back up data sets by command, specifying [185](#)
- withdrawing FlashCopy relationship [282](#)
- write dump VTOC copy data set [198](#)
- writing to a cartridge-type, single-file format tape during migration [78](#)

X

- XMIT parameter, ARECOVER command [393](#)
- XRC primary volume
 - FlashCopy operation
 - FRRECOV [887](#)

Z

- z/OS images [593](#)
- ZCOMPRESS [1207](#)
- zEDC [1207](#)
- zEDC compression
 - during migration [75](#)



Product Number: 5655-ZOS

SC23-6871-70

