

z/OS DFSMSdfp Storage Administration.....	13
Abstract for DFSMSdfp Storage Administration.....	14
Where to find more information.....	15
Changes for z/OS V2R1, as updated March, 2014.....	16
z/OS Version 2 Release 1 summary of changes.....	17
Introducing the Storage Management Subsystem.....	18
Understanding the Storage Management Subsystem.....	19
Using the Interactive Storage Management Facility.....	20
Customizing ISMF.....	23
ISMF libraries that can be customized.....	24
Restrictions to customizing ISMF.....	25
Defining your storage management policy.....	26
Preparing for and implementing the Storage Management Subsystem.....	29
Running SMS in a parallel sysplex environment.....	30
Basic terms and definitions.....	31
System grouping.....	32
Running SMS on mixed DFSMS releases.....	34
Preparing for the Storage Management Subsystem.....	35
Allocating control data sets.....	36
Source control data set (SCDS).....	37
Active control data set (ACDS).....	38
Communications data set (COMMDS).....	40
Calculating the size of storage and active control data sets.....	41
Calculating the size of a COMMDS.....	44
Selecting volumes for control data sets.....	45
Allocating an SCDS.....	46
Allocating an ACDS.....	47
Allocating a COMMDS.....	48
Recommendations.....	49
Modifying the SYS1.PARMLIB data set.....	50
Initializing SMS through the IGDSMSxx member.....	51
Starting the SMS address space.....	52
Accessing the storage administrator primary option menu.....	53
Planning for VSAM record-level sharing.....	54
Creating the base configuration.....	55
Planning the base configuration.....	56
Defining the base configuration.....	57
Validating the SCDS.....	60
Activating the SCDS.....	61
Specifying the default management class.....	62
Specifying the default unit.....	63
Specifying the default device geometry.....	64
Specifying the DS separation profile.....	65
Specifying systems and system groups in the SMS complex.....	66
Defining the base configuration for VSAM record-level sharing.....	69
Using cache structures.....	70
Defining cache sets.....	71

Defining lock sets.....	72
Completing the base configuration.....	73
Defining storage groups.....	74
Understanding storage groups.....	75
DASD storage groups.....	77
DASD volume status for data sets.....	79
Object and object backup storage groups.....	81
Defining real and pseudo optical libraries in object storage groups.....	84
Defining optical libraries in object backup storage groups.....	85
Writing objects.....	86
OAM collection names.....	88
Tape storage groups.....	89
SMS.....	90
Planning storage groups for data sets.....	91
Planning storage groups for OAM object collections.....	94
Defining storage group attributes.....	97
Storage group types.....	99
Defining a VIO storage group.....	102
Values for defining a VIO storage group.....	103
Defining a pool storage group.....	104
Values for defining a pool storage group.....	105
Defining system access to pool and VIO storage groups.....	110
Defining pool storage group status.....	113
Adding volumes and defining the SMS volume status.....	114
Values on the SMS volume status define panel.....	116
Assigning DASD storage groups to data sets.....	117
Defining a dummy storage group.....	118
Defining DASD volumes to a dummy storage group.....	119
Defining a copy pool backup storage group.....	122
Defining an object storage group.....	123
Attributes of the object storage group.....	124
Defining an object backup storage group.....	128
Attributes of the object backup storage group.....	129
Defining object or object backup storage group status.....	132
Attributes for object or object backup storage group status.....	133
Assigning an OAM object collection to a storage group.....	134
Defining a tape storage group.....	135
Attributes for tape storage groups.....	136
Defining tape storage group status.....	137
Defining additional storage groups.....	138
Listing volumes in a storage group.....	139
Defining reserve storage pools.....	140
Understanding reserve storage pools.....	141
Assigning volumes to a reserve storage pool.....	142
Defining management classes.....	143
Understanding management classes.....	144
Default management class.....	145

OAM management classes	146
Describing management classes	147
Planning management classes	148
Defining management class attributes	149
Defining management class expiration attributes	151
The use of expiration attributes and retention limit	153
Defining management class migration attributes	155
Defining management class backup attributes	159
Defining class transition attributes	162
Defining aggregate backup attributes	164
Assigning management classes	165
Defining additional management classes	167
Defining storage classes	168
Understanding storage classes	169
Storage classes for data sets	170
Storage classes for objects	171
Planning storage classes	172
Defining storage class attributes	173
Defining performance objectives	176
Defining millisecond response time	177
Millisecond response time and data set allocation	178
Millisecond response time and cache management	179
Defining bias	182
Defining initial access response seconds	183
VTS (Virtual Tape Server) cache management	184
Object access	185
Data set access	186
Defining sustained data rate	187
Physical sequential and VSAM data sets	188
Object access	190
Defining availability	191
Defining accessibility	192
Defining the guaranteed space attribute	195
Preallocating space for multivolume data sets	196
Honoring specific volume requests	199
Defining guaranteed synchronous write	201
Defining use of the coupling facility for VSAM record-level sharing	202
Defining when to disconnect the sphere	203
Assigning storage classes	204
Defining additional storage classes	205
SMS volume selection for data set allocation	206
Volume selection preference attributes	207
Conventional volume selection	208
Reasons a volume is not placed on the primary list	211
Striping volume selection	212
Rules for striping volume selection	215
Requirements for data set striping	218

Tuning considerations.....	219
Cluster and storage facility image considerations.....	220
Spreading allocations across multiple volumes.....	221
Space constraint relief.....	222
Using the multi-tiered storage group function.....	223
Using the parallel access volume option.....	224
Planning to use the PAV option in the storage class.....	225
Implementing the PAV option enhancement.....	226
Modifying storage classes to enable the PAV option.....	227
Defining new storage classes with the PAV option enabled.....	228
Displaying information about the PAV capability of a volume.....	229
Diagnosing problems with the PAV option.....	230
Possible reasons for volume selection failure.....	231
Defining secondary lock tables.....	234
Defining data classes.....	237
Understanding data classes.....	238
Planning data classes.....	240
Defining data class attributes.....	242
Defining record and space attributes for data class.....	243
Defining volume and data set attributes for data class.....	246
Specifying attributes to handle space constraints during allocation.....	251
Defining VSAM attributes and specifying media types for data class.....	254
Defining the encryption management mechanism.....	257
Defining Shareoptions and RLS attributes for data class.....	258
Specifying attributes for data set reuse and loading.....	261
Specifying attributes for backup-while-open (BWO) and recovery.....	262
Assigning data classes.....	265
Specifying data classes outside ACS routines.....	266
Processing data class attributes in JCL.....	267
Defining additional data classes.....	269
Defining aggregate groups.....	270
Understanding aggregate groups.....	271
Planning aggregate groups.....	272
Defining aggregate groups.....	273
Defining aggregate group attributes.....	275
Editing aggregate group attributes.....	276
Backing up and recovering an aggregate group.....	279
Defining additional aggregate groups.....	280
Defining copy pools.....	281
Planning a copy pool.....	283
Defining a copy pool.....	284
Steps for defining a copy pool.....	285
Defining a copy pool backup storage group.....	289
Defining ACS routines.....	290
Understanding ACS routines.....	291
Using ACS routines for data sets created by z/OS Network File System and Distributed FileManager/MVS.....	294

Restrictions on using ACS routines.....	295
Creating ACS routines.....	296
Translating ACS routines.....	297
Browsing the results of a translation.....	298
Browsing the results of an unsuccessful translation.....	299
Validating ACS routines or an entire SCDS.....	300
Validating an ACS routine.....	301
Validation results.....	302
Browsing validation results.....	303
Validating an entire SCDS.....	304
Validation results.....	305
Translating and validating in a sysplex environment.....	306
Testing ACS routines.....	307
ACS test usage.....	308
Creating ACS test cases.....	309
Running ACS test cases.....	310
ACS routines invoked for copying and importing data sets.....	311
Initial allocation.....	312
IMPORT (Access Method Services).....	313
COPY (DFSMSdss).....	314
COPY BYPASSACS (DFSMSdss).....	315
ACS routines invoked for restoring, recalling, recovering, and converting data sets.....	316
DFSMSdss RESTORE.....	317
DFSMSdss RESTORE BYPASSACS.....	318
DFSMSdss CONVERTV.....	319
DFSMShsm RECALL/RECOVER.....	320
DFSMShsm Class Transitions.....	321
DFSMShsm FORCENONSMS.....	322
ACS routine environments.....	323
JCL DD statement (batch), and dynamic allocation.....	324
Volume reference.....	325
Examples: using volume reference.....	328
Data set stacking.....	330
Using VOL=SER within a job step.....	331
Using VOL=REF.....	332
Possible values for the &UNIT read-only variable.....	333
Examples: storage class ACS routines for read-only variables.....	334
Non-data set stacking allocations.....	335
Volume reference and unit affinity.....	336
Example: storage class ACS routine specifying both volume reference and unit affinity.....	337
Tape management system support.....	338
Access Method Services.....	341
DFSMShsm.....	342
DFSMSdss.....	343
DFSMSrmm.....	344
ISMF.....	345
OAM.....	346

Processing of SMS classes and storage groups.....	347
Displaying ACS object information.....	348
Deleting an ACS object from an SCDS.....	349
Using security labels in ACS routines.....	350
Planning and installing.....	351
Planning to use security labels in ACS routines.....	352
Installing security labels in ACS routines.....	353
Administering.....	354
Creating a security label in the RACF data set or user's profile.....	355
Using JCL or dynamic allocation to extract the security label from the RACF user's profile.....	357
Specifying the &SECLABL read-only variable in the ACS routine.....	359
Writing ACS test cases using SECLABEL.....	362
Activating Storage Management Subsystem configurations.....	363
Manually activating the first Storage Management Subsystem configuration.....	364
Step one: IPL each system in the SMS complex.....	365
Step two: prepare one system.....	366
Step three: activate the configuration from one system.....	367
Activating with the ISMF ACTIVATE command.....	368
Activating with the SETSMS operator command.....	369
Step four: activate SMS on the other systems.....	370
Automatically activating a Storage Management Subsystem configuration.....	371
Displaying storage management subsystem information.....	372
Controlling DFSMSsts processing.....	373
Monitoring application programs that use DFSMSsts.....	374
Changing DFSMSsts status.....	375
Changing Storage Management Subsystem parameters.....	376
Parameters of the SETSMS operator command.....	377
SETSMS command to alter the setting of BreakPointValue.....	378
SETSMS command to alter the setting of USEEAV.....	379
Considerations when changing Storage Management Subsystem configurations.....	380
OAM considerations when changing SCDSs.....	381
Maintaining the Storage Management Subsystem.....	382
Displaying SMS and OAM information.....	383
Displaying information about the active configuration.....	384
Displaying SMS TRACE information.....	385
Displaying storage group status using the DISPLAY SMS command.....	386
Displaying the status of volumes in the storage group.....	389
Displaying storage group detail status.....	390
Displaying OAM status in a parallel sysplex.....	391
Displaying the caching statistics.....	392
Displaying storage group status using ISMF.....	393
Displaying volume status using ISMF.....	395
Values on the SMS volume status display panel.....	396
Displaying volumes using the DISPLAY SMS command.....	397
Displaying the device status.....	398
DEVSERV SMS.....	399
DEVSERV QPAVS.....	400

DEVSERV PATHS.....	405
DEVSERV QDASD.....	406
Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF.....	407
Listing with view and sort.....	408
Listing data classes.....	409
Listing storage groups, management classes, storage classes, aggregate groups, libraries and drives.....	412
Processing management classes.....	413
Altering data set associations.....	414
Altering SMS components.....	415
Altering storage groups.....	416
Redefining volumes among storage groups.....	417
Altering management classes.....	420
Values on the management class alter panel.....	421
Altering storage classes.....	422
Altering data classes.....	423
Altering copy pools.....	424
Altering aggregate groups.....	425
Editing Aggregate group attributes.....	426
Altering the SCDS on different DFSMS releases.....	428
Copying SMS components.....	429
Deleting DASD volumes from the system.....	431
Deleting DASD volumes from storage groups.....	432
Deleting storage groups.....	433
Deleting or moving system-managed data sets from the storage group.....	434
Preventing job failures.....	435
Deleting SMS classes.....	436
Converting volumes to SMS.....	439
Deleting DASD volume residual data.....	440
Recovering Storage Management Subsystem information.....	441
Recovering control data sets.....	442
Recovering an SCDS.....	443
Recovering an ACDS.....	444
Recovering a COMMDS.....	446
Recovering from a systems failure in the SMS complex.....	448
Recovering the SMS address space.....	449
Canceling the SMS address space.....	450
Protecting the Storage Management Subsystem.....	451
Identifying the resource owner and extracting the default classes.....	452
Protecting ISMF functions.....	453
Locating module names for ISMF applications.....	455
Locating module names for ISMF functions.....	456
Locating module names for line operators and commands.....	457
Protecting modules.....	458
Storage administration (STGADMIN) profiles in the FACILITY class.....	459
Command and keyword related profiles.....	460
Authority to activate a Storage Management Subsystem configuration.....	464

FIELD resource class.....	465
How authorization and protection of classes differ.....	466
Authorizing the storage and management classes.....	467
Protecting the control data sets.....	468
Authorizing for TSO/E and ISPF.....	469
Storage administrator authorization.....	470
Administering VSAM record-level sharing.....	471
Preparing for VSAM record-level sharing.....	473
Determining hardware requirements.....	474
Understanding the product environment for VSAM RLS.....	475
Determining applications that can use VSAM RLS.....	476
Ensuring same systems connectivity.....	478
Planning for availability.....	479
Defining sharing control data sets.....	480
Defining CF cache structures.....	484
Sharing CF structures.....	485
Determining CF cache structure size.....	486
Using the RLS_MAX_POOL_SIZE parameter to limit local buffer pool size.....	487
Defining the primary CF lock structure.....	488
Considerations for retained locks and record table full conditions.....	491
Avoiding false contention.....	494
Monitoring for false contention.....	496
How much contention is acceptable.....	497
Reducing false contention.....	498
Adjusting the lock structure size.....	499
A lock structure sizing example.....	500
Modifying the SYS1.PARMLIB IGDSMSxx member.....	501
Establishing authorization for VSAM RLS.....	504
Using dssTimeOut.....	505
Defining CF cache structures in the SMS base configuration.....	506
Defining storage classes for VSAM RLS.....	508
Defining VSAM RLS attributes in data classes.....	509
Activating VSAM RLS.....	510
Enabling VSAM RLS processing.....	511
Enabling a data set for VSAM RLS processing.....	512
Monitoring the coupling facility for VSAM RLS.....	513
Displaying CF information.....	514
Using the DISPLAY SMS command.....	515
Using ISMF.....	518
Data set application.....	519
Volume application.....	520
Storage class application.....	521
Control data set application.....	522
Changing the size of coupling facility cache structures.....	523
Changing the size of coupling facility lock structures.....	524
Changing IGDSMSxx parameters to support the coupling facility.....	525
Changing the state of coupling facility cache structures and volumes.....	529

Selecting data sets for coupling facility statistical monitoring.....	530
Recovering VSAM RLS processing.....	531
Recovering the CF lock structure.....	532
User-managed lock structure rebuild.....	533
System-managed lock structure duplexing rebuild.....	534
Recovering a CF cache structure.....	535
Recovering the SMSVSAM server address space.....	536
Recovering a shared control data set.....	537
Falling back from VSAM RLS processing.....	538
Fallback rules and considerations.....	539
Fallback procedure.....	540
Terminating the SMSVSAM address space.....	543
Quiescing or enabling a secondary lock structure.....	544
Deleting a VSAM RLS lock structure.....	545
Displaying information about a secondary lock structure.....	546
Writing ACS routines.....	547
Constants.....	548
Simple mask rules.....	550
Simple mask examples.....	551
Data set mask rules.....	552
Data set mask examples.....	553
Read-write variables.....	554
Using read-write variables with volume reference (VOL=REF).....	555
Using read-write variables with data set stacking.....	556
Read-only variables.....	557
Using read-only variables.....	564
Initializing read-only variables.....	565
Using default read-only variables.....	566
Ensuring correct values for &SIZE and &MAXSIZE.....	567
Using OAM read-only variables.....	569
Using &DSN for a partitioned data set.....	570
When the value of &USER is null.....	571
Using the &SYSNAME and &SYSPLEX read-only variables.....	572
Read-only variables on a data set rename.....	573
Constraints when using read-only variables.....	574
Read-only variables not allowed in storage group routine.....	575
Read-only variables in different environments.....	576
Read-only variables not available when LIKE is used.....	577
OAM read-only variables.....	578
Comparison operators.....	579
Comparison rules.....	580
Boolean expressions.....	581
Special functions.....	582
Data set qualifier indexing function.....	583
&ALLVOL and &ANYVOL functions.....	584
Accounting information indexing function.....	585
Virtual tape system parameter.....	586

Determining Distributed FileManager/MVS data set creation requests.....	587
Statements.....	589
PROC statement.....	591
FILTLIST statement.....	592
SET statement.....	594
DO statement.....	595
IF statement.....	596
SELECT statement.....	598
EXIT statement.....	600
WRITE statement.....	601
END statement.....	604
Sample ACS routine.....	605
Quick reference to ISMF commands and line operators.....	606
SETCACHE functions and device information.....	607
Maintaining the media characteristics of a volume.....	608
Modifying DASD storage control characteristics.....	609
Modifying caching characteristics.....	611
Read caching/Read and system caching.....	612
Write caching.....	613
Modifying duplexing characteristics.....	614
Modifying destaging characteristics.....	615
Submitting jobs.....	616
Providing remote authorization codes.....	617
Space utilization and capacity planning.....	618
Using the data collection application.....	619
Using data set separation.....	623
Overview.....	624
Syntax for creating a data set separation profile.....	625
Deprecated syntax.....	627
Migrating to the new syntax.....	628
Example.....	629
Writing comments.....	630
Indicating continuation and termination.....	631
Example.....	632
Volume selection for data set separation by volume.....	633
Data set requirements for a data set separation profile.....	634
Creating multiple data set separation profiles.....	635
Using data set separation with generation data groups and striping.....	636
Environmental conditions affecting data set separation.....	637
Factors affecting code path length.....	638
Using NaviQuest.....	639
Starting NaviQuest.....	641
Terminology.....	642
Methods for collecting test data.....	645
ISMF lists.....	646
DCOLLECT data.....	647
SMF data.....	648

VMA data.....	649
Data classification methodology.....	650
Testing ACS routines.....	652
Setting up the test environment.....	653
Testing procedures.....	656
NaviQuest testing scenario.....	663
Data testing.....	664
Method of testing.....	665
Performing storage administration tasks in batch.....	666
How to run storage administration tasks in batch.....	667
ISMF option 11.7 batch testing and configuration management.....	668
Using the sample JCL for batch.....	669
ISPSTART batch parameters for NaviQuest.....	670
SYS1.SACBCNTL sample JCL library.....	671
ACBJBAOB JCL.....	672
NaviQuest CLISTS and REXX EXECS.....	675
Generate a data set list: ACBQBAI2.....	676
Save or delete the data set list.....	677
Sample JCL for generating a data set list.....	678
Generate a DASD volume list: ACBQBAI4.....	681
Sample JCL for generating a volume list.....	682
Generate a tape volume list: ACBQBAI6.....	685
Sample JCL for generating a tape list.....	686
Define/Alter/Display management class: ACBQBAJ1.....	689
Sample JCL for define/Alter/Display management class.....	690
Storage group volume add/Delete: ACBQBAI9.....	695
Sample JCL for storage group volume add/Delete.....	696
ACS routine translate: ACBQBAO1.....	699
Sample JCL for ACS routine translate.....	700
SCDS validation: ACBQBAO2.....	704
Sample JCL for SCDS validation.....	705
Test ACS routines: ACBQBAIA.....	706
Sample JCL for test ACS routines.....	707
Generate test cases from ISMF-saved data set lists: ACBQBAG3.....	708
Sample JCL for generating test cases from ISMF-saved data set lists.....	709
Generate test cases from DCOLLECT data: ACBQBAG1.....	710
Sample JCL for generating test cases from DCOLLECT data.....	711
Generate test cases from SMF data.....	712
Sample JCL for generating test cases from SMF data.....	713
Generate test cases from VMA extract data: ACBQBAO3.....	714
Sample JCL for generating test cases from VMA.....	715
ACS test listings comparison: ACBQBAC1.....	716
Sample JCL for test listings comparison.....	717
Enhanced ACS test listing: ACBQBAX1.....	718
Sample JCL for generating an enhanced ACS test listing.....	719
Update test cases with expected results: ACBQBAU1.....	721
Sample JCL to update test cases.....	722

Generate report from ISMF-saved data set list: ACBQBAR1	723
Sample JCL for generating a report from ISMF-saved data set list.....	724
Generate report from ISMF-saved DASD volume list: ACBQVAR1	727
Sample JCL for generating a report from ISMF-saved DASD volume list.....	728
Generate report from ISMF-saved tape list: ACBQBAR4	731
Sample JCL for generating a report from ISMF-saved tape list.....	732
Generate data set report from DCOLLECT data: ACBQBAR7	735
Sample JCL for generating data set report from DCOLLECT data.....	736
Generate model commands from ISMF-saved list: ACBQBAM1	739
Sample JCL for generating model commands from ISMF-saved list.....	740
Creating SMS online reports.....	741
Creating data set reports from saved ISMF lists.....	742
Creating volume reports from saved ISMF lists.....	743
Creating customized tape reports from saved ISMF lists.....	744
Data set report from DCOLLECT data.....	745
DASD volume report from DCOLLECT data.....	746
SMS configuration report from DCOLLECT data.....	747
Additional storage administration functions.....	748
QSAVE and QRETRIEV commands: saving and retrieving ISMF selection criteria.....	749
How to use the QSAVE and QRETRIEV commands.....	750
Creating model commands.....	751
How to create model commands.....	752
COPYFILT macro: COPYLIB facility for FILTLISTS.....	755
How to set up the COPYFILT macro.....	756
How to use the COPYFILT macro after initial setup.....	757
Sample Batch Job for CICS Definitions.....	759

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Contents (exploded view)

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Abstract for DFSMSdfp Storage Administration
Where to find more information
Changes for z/OS V2R1, as updated March, 2014
Introducing the Storage Management Subsystem
Preparing for the Storage Management Subsystem
Creating the base configuration
Defining storage groups
Defining reserve storage pools
Defining management classes
Defining storage classes
Defining data classes
Defining aggregate groups
Defining copy pools
Defining ACS routines
Activating Storage Management Subsystem configurations
Maintaining the Storage Management Subsystem
Recovering Storage Management Subsystem information
Protecting the Storage Management Subsystem
Administering VSAM record-level sharing
Writing ACS routines
Quick reference to ISMF commands and line operators
SETCACHE functions and device information
Space utilization and capacity planning
Using data set separation
Using NaviQuest
Sample Batch Job for CICS Definitions
FOOTER



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Abstract for DFSMSdfp Storage Administration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Purpose of this information The purpose of this information is to introduce system programmers and storage administrators to DFSMSdfp and provide guidance on storage management concepts.

Who should read this information This information is intended for storage administrators and system programmers.



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Where to find more information

z/OS DFSMSdfp Storage Administration
SC23-6860-01

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

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

To find the complete z/OS library, including the z/OS Information Center, go to the [z/OS Internet library](#).



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Changes for z/OS V2R1, as updated March, 2014

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following changes are made for z/OS Version 2 Release 1 (V2R1) as updated March, 2014. In this revision, all technical changes for z/OS V2R1 are indicated by a vertical line to the left of the change.

New

The ZR (“zEDC Required”) and ZP (“zEDC Preferred”) options are added to the Compaction field in [Defining volume and data set attributes for data class](#).

Note: For more information on the zEDC compression enhancements, see z/OS DFSMS Using the New Functions.

[z/OS Version 2 Release 1 summary of changes](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

z/OS Version 2 Release 1 summary of changes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

See the following publications for all enhancements to z/OS Version 2 Release 1 (V2R1):

z/OS Migrationz/OS Planning for Installationz/OS Summary of Message and Interface Changesz/OS Introduction and Release Guide

Parent topic:[Summary of changes for z/OS Version 2 Release 1 \(V2R1\) as updated March, 2014](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Introducing the Storage Management Subsystem

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This topic introduces the Storage Management Subsystem (SMS). It explains various applications of the Interactive Storage Management Facility (ISMF) so that you can use them to define your storage management policy and manage your SMS configurations. It also provides an overview of the preparatory tasks for implementing SMS and explores considerations for running SMS in a Parallel Sysplex® environment and on mixed DFSMS releases.

[Understanding the Storage Management Subsystem](#)

[Using the Interactive Storage Management Facility](#)

[Defining your storage management policy](#)

[Preparing for and implementing the Storage Management Subsystem](#)

[Running SMS in a parallel sysplex environment](#)

[Running SMS on mixed DFSMS releases](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Understanding the Storage Management Subsystem

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The Storage Management Subsystem (SMS) is a DFSMS facility designed for automating and centralizing storage management. Using SMS, you can describe data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the systems. SMS improves storage space use, allows central control of external storage, and enables you to manage storage growth more efficiently. With SMS, you can easily manage conversion between device types and ultimately move toward system-managed storage.

Parent topic:[Introducing the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using the Interactive Storage Management Facility

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The Interactive Storage Management Facility (ISMF) provides a series of applications for storage administrators to define and manage SMS configurations. You can use these applications to: Define SMS base configuration information. Define, alter, delete, or copy individual SMS classes, storage groups, aggregate groups, optical libraries, optical drives, and tape libraries. Display parameters and values of individual SMS classes, storage groups, aggregate groups, mountable optical volumes, optical drives, mountable tape volumes, tape libraries and optical libraries. Generate, save and manage lists of SMS classes, storage groups, aggregate groups, mountable optical volumes, optical libraries, optical drives, mountable tape volumes and tape libraries. Edit ACS routines. Define, alter, and execute ACS test cases. Validate the correctness and completeness of an SMS configuration. Activate an SMS configuration. Display, define, alter, or delete storage group information pertaining to specific volumes using AUDIT, EJECT, ALTER, and RECOVER (RECOVER is for optical volumes only). Produce data set, volume, or capacity planning measurement data. Maintain mountable optical volumes and mountable tape volumes. Use DFSMSrmm to maintain tape volumes. Use the DFSMS NaviQuest tool to perform enhanced testing of your ACS routines, and to perform many storage management tasks in batch, such as: Updating and testing your base configuration. Translating and testing your ACS routines. Generating test cases from previously collected DCOLLECT data. Defining, altering, and displaying information for management classes, data classes and storage classes. Defining or altering information for storage groups. Defining, altering, and displaying information for the base configuration, as well as for aggregate groups. Generating data set and volume lists and reports. Diagnosing data set and volume problems. See [Using NaviQuest](#) for more information on the DFSMS NaviQuest tool.

[Figure 1](#) shows the ISMF Primary Option Menu for storage administrators. Figure 1. ISMF Primary Option Menu for Storage Administrators

```

ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R12
Selection or Command ==>                                         Panel Help
0  ISMF Profile          - Specify ISMF User Profile
1  Data Set               - Perform Functions Against Data Sets
2  Volume                 - Perform Functions Against Volumes
3  Management Class       - Specify Data Set Backup and Migration Criteria
4  Data Class              - Specify Data Set Allocation Parameters
5  Storage Class           - Specify Data Set Performance and Availability
6  Storage Group           - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set        - Specify System Names and Default Criteria
9  Aggregate Group         - Specify Data Set Recovery Parameters
10 Library Management      - Specify Library and Drive Configurations
11 Enhanced ACS Management - Perform Enhanced Test/Configuration Management
C  Data Collection          - Process Data Collection Function
G  Report Generation        - Create Storage Management Reports
L  List                     - Perform Functions Against Saved ISMF Lists
P  Copy Pool                - Specify Pool Storage Groups for Copies
R  Removable Media Manager   - Perform Functions Against Removable Media
X  Exit                     - Terminate ISMF
Use HELP Command for Help; Use END Command or X to Exit.

```

This primary option menu differs from the one that end users see. To the options found on the ISMF Primary Option Menu for end users, this primary option menu adds the following:

Storage Group, Automatic Class Selection, Control Data Set, Library Management, Data Collection, Enhanced ACS Management, Copy Pool. Additionally, the management class, data class, storage class, and aggregate group applications available through the ISMF Primary Option Menu for Storage Administrators allow you to define, alter, copy, and delete SMS classes. End users can list the available SMS classes, display the attributes of individual SMS classes, and list or display volumes and data sets.

To learn how to use the ISMF Data Collection application, see [Space utilization and capacity planning](#).



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Customizing ISMF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

There might be times when you want to customize the interactive storage management facility (ISMF). This topic lists the libraries that you can customize and some restrictions. For more information, see *z/OS DFSMS Using the Interactive Storage Management Facility*, which describes ISMF restrictions and customization with DFSMSdss.

[ISMF libraries that can be customized](#)

Parent topic:[Using the Interactive Storage Management Facility](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ISMF libraries that can be customized

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can customize the following ISMF libraries:

- Panel Library** ISMF allows you to make the following changes:
 - Change the initial priming values that ISMF ships
 - Change the default values for data entry panels
 - Provide additional restrictions to values that are entered for certain fields on panels
 - Remove fields from functional panels
 - Change highlighting and color
 - Change the format of the panel
 - Modify existing functional panel text and help text
 - Add new fields to panels
 - Add new panels.
- Message Library** You can modify existing messages and add new messages.
- Skeleton Library** You can modify the job skeletons for ISMF commands and line operators.
- Table Library** You can modify the Interactive System Productivity Facility (ISPF) command tables.
- Load Library** You can modify the ISMF command and line operator tables. The tables are contained in nonexecutable CSECTs in the load library.
- CLIST Library** You can modify the options on the CLIST CONTROL statement.

[Restrictions to customizing ISMF](#)

Parent topic: [Customizing ISMF](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Restrictions to customizing ISMF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

General restrictions for ISMF customization include the following:

Before changing anything, you should make a backup copy of ISMF. Keep this unmodified version of the product for diagnostic purposes. IBM® support and maintenance is provided only for the unmodified version of ISMF. Do not delete or rename any of the parts of ISMF. Deleting or renaming severely affects processing and can cause ISMF to fail. ISMF is copyrighted. Under the IBM licensing agreement you may modify ISMF for your own use. You may not, however, modify it for commercial resale.

Parent topic:[ISMF libraries that can be customized](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining your storage management policy

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SMS manages an installation's storage according to the currently active storage management policy. Through ISMF, you define an installation storage management policy in an *SMS configuration*. An SMS configuration contains the following:

- Base configuration information
- Classes and groups
- Automatic class selection (ACS) routines
- Optical library and drive definitions
- Tape library definitions

The *base configuration* identifies the systems that the SMS configuration manages. These systems constitute an *SMS complex*. The base configuration also contains installation defaults.

You can define more than one control data set, but only one at a time controls SMS. Each control data set defined for SMS is called a *source control data set (SCDS)*. The control data set that is in effect at a given time is the *active control data set (ACDS)*.

SMS classes and groups are lists of traits and characteristics that are associated with or assigned to data sets, objects and volumes. An SMS configuration can contain the following types of classes and groups:
Storage group Use this to define a list of volumes and manage them as if they were one large, single volume. SMS applies the properties you assign to a storage group to all the volumes within the storage group.
Management class Use this to define different levels of migration, backup and retention services. Through management class, you can associate a level of service with a data set or object that is independent of the physical location of the data set or object. Also, you can identify an object characteristic that might trigger a class transition.
Storage class Use this to define different levels of performance and availability services. Through storage class, you can separate the level of service for a data set or object from physical device characteristics. You can also separate the level of service for an object with different storage classes used to place objects at various levels of the storage hierarchy.
Data class Use this to define allocation defaults. Through data class, you can simplify and standardize the allocation of new data sets.
Aggregate group Use this to define groups of data sets for the purpose of backing up or recovering all data sets in a group in a single operation.
Copy pool Use this to define a pool of storage groups to be processed collectively for fast replication operations. An SMS configuration can contain multiple constructs of each type. Data sets managed by SMS are called *system-managed*. Each system-managed data set or object must reside in a storage group. The system-managed data sets must have a storage class, and might also have a management class and a data class. The objects must have a storage class and a management class.

You can assign the same name to various SMS classes and a storage group. For example, a data class and a storage class can have the same name.

ACS routines determine the SMS classes and storage groups for data sets and objects. You can also use ACS routines to control the transition of data sets to and from SMS management (objects are always SMS-managed).

Parent topic: Introducing the Storage Management Subsystem



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Preparing for and implementing the Storage Management Subsystem

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To prepare for and implement a storage management policy, you perform these steps:

Prepare for SMS, as described in [Preparing for the Storage Management Subsystem](#). This includes: Allocating control data sets, which contains information used by SMS. Modifying and creating SYS1.PARMLIB members to identify SMS to all the systems in the SMS complex. These new members take effect when you IPL. Establishing access to the ISMF Primary Option Menu for Storage Administrators, which is shown in [Figure 1](#). This is the final preparation step. Define the base configuration, which identifies the systems within the SMS complex. See [Creating the base configuration](#). Define the SMS classes and storage groups that you want SMS to assign to your data sets and objects, and the data sets that you want to assign to your aggregate groups. See [Defining storage groups](#), [Defining management classes](#), [Defining storage classes](#), [Defining storage classes](#), [Defining aggregate groups](#) and [Defining aggregate groups](#). Define optical libraries and drives and tape libraries, if you have them. See [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support](#) and [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries](#). Define ACS routines to assign the SMS classes and storage groups. Then test the routines. See [Defining ACS routines](#). Validate the ACS routines individually to check for errors. You should then validate the entire SMS configuration to check for errors that exist among its related parts. Activate the valid SMS configuration. See [Activating Storage Management Subsystem configurations](#).

Parent topic: [Introducing the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Running SMS in a parallel sysplex environment

z/OS DFSMSdfp Storage Administration
SC23-6860-01

An SMS complex consists of systems or *system groups* that share a common configuration. A Parallel Sysplex® is made up of systems that share a cross-system coupling facility (XCF); you can run multiple SMS complexes within a Parallel Sysplex.

Restrictions: An SMS complex should not span sysplexes. All of the volumes in the SMS complex should be in the same Parallel Sysplex. The cross-system sharing functions, such as VSAM record-level sharing (RLS), partitioned data set extended (PDSE) sharing, z/OS® Security Server RACF® security and global shared resources (GRS) serialization, work only within the scope of a single Parallel Sysplex. These functions are not supported when the SMS complex extends beyond the Parallel Sysplex in which they are carried out. Do not set up multiple SMS complexes sharing the same DASD. This requires extra work to maintain the duplicate SMS configurations and can also create problems such as running out of disk space because one configuration cannot know about changes made to the other configuration, such as data set allocations and deletions, and storage group and volume status changes. In a JES3 environment, system group name is not supported due to the fact that the system features and resources that are used to determine on which eligible system a job will be run are only applicable to each system, not to a system group.

[Basic terms and definitions](#)

[System grouping](#)

Parent topic: [Introducing the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Basic terms and definitions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following are some basic terms and definitions:

SMS complexA system or a collection of systems that share a common configuration including a common active control data set (ACDS) and a common communication data set (COMMDS) pair. The SMS configuration now supports up to 32 system names, system group names, or both.**Parallel Sysplex®**A collection of MVS systems in a multi-system environment supported by the cross-system coupling facility (XCF).**System name**The name of the system where an SMS operation is being performed.**System group**The system names within a Parallel Sysplex, excluding those systems in that sysplex, if any, that are individually defined in the SCDS. This support can be used even if the XCF is not active.

Parent topic:[Running SMS in a parallel sysplex environment](#)



©Copyright IBM Corporation 1990, 2014

System grouping

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In the MVS environment, a Parallel Sysplex® is a collection of systems linked by closely coupled hardware facilities to process customer workloads. SMS system group name support allows you to specify a system group as a member of an SMS complex.

In SMS, the term *system group* is used instead of *Parallel Sysplex*. A system group consists of system names within a Parallel Sysplex, excluding those systems that are individually specified in the SMS base configuration. A system group name can represent multiple systems. This allows SMS to support more than eight systems per SMS complex while retaining the existing configuration format of the configuration data set.

Object storage groups, object backup storage groups, and optical libraries do not support system groups. If you have object storage groups, object backup storage groups or optical libraries in your configuration, the systems where they are to be enabled must be specified as individual system names.

A system group name matches a Parallel Sysplex name and refers to all systems defined as part of the Parallel Sysplex that are: Running the same SMS configurationDefined in the configuration using the name of the Parallel Sysplex to which they belong (that is, system group)Not defined in the configuration by their system names.

The system group name represents all the systems in the Parallel Sysplex which are not explicitly specified in the SCDS base configuration.

[Figure 1](#) provides a visual representation of the system grouping concept.

Figure 1. System Grouping Concept

In [Figure 1](#), the SMS complex view shows that it is managing sys 1 and sys 4 as single systems, and Maui as a Parallel Sysplex. (sys 5 is not part of the SMS complex.)

The Parallel Sysplex (Maui) contains three system names: sys 1, sys 2, and sys 3. The single systems sys 1 and sys 4, as well as the Parallel Sysplex (Maui) are all defined in the SCDS. If you perform an SMS operation on Maui, it affects only sys 2 and sys 3 in the Parallel Sysplex. It does not affect sys 1 because it is defined separately in the SCDS.

In this example, the system group (sys 2 and sys 3) is represented by the Parallel Sysplex name, Maui.

Parent topic:[Running SMS in a parallel sysplex environment](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Running SMS on mixed DFSMS releases

z/OS DFSMSdfp Storage Administration
SC23-6860-01

See z/OS Migration for information on running SMS on mixed DFSMS releases.

Parent topic:[Introducing the Storage Management Subsystem](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Preparing for the Storage Management Subsystem

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you define and activate an SMS configuration, perform the following preparatory steps: Allocate control data sets to contain your SMS configuration and to permit the systems in your complex to communicate with each other. See [Allocating control data sets](#). Modify SYS1.PARMLIB, which contains three members that direct the initialization and activation of SMS. See [Modifying the SYS1.PARMLIB data set](#). Establish access to the ISMF Primary Option Menu for Storage Administrators (shown in [Figure 1](#)). This topic describes how to perform these preliminary steps so that you can begin defining a base configuration for an SMS configuration.

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

For information about planning and preparing for SMS with object support, optical libraries, or tape libraries, see [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support](#) and [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries](#).

[Allocating control data sets](#)

[Modifying the SYS1.PARMLIB data set](#)

[Accessing the storage administrator primary option menu](#)

[Planning for VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Allocating control data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you can activate an SMS configuration, allocate the SMS control data sets and define their contents. Control data sets are virtual storage access method (VSAM) linear data sets that contain: Base configuration informationSMS class, aggregate group, copy pool, optical library, tape library, optical drive, and storage group definitionsACS routines

You can allocate control data sets with access method services or TSO/E commands. You can define and alter the contents of control data sets using ISMF. SMS uses three types of control data sets: a *source control data set*(SCDS), an *active control data set*(ACDS), and a *communications data set*(COMMDS).

Restrictions: Do not name any of your SMS control data sets the single word ACTIVE. SMS uses the single word ACTIVE as a reserved word indicating the active configuration residing in the SMS address space. Naming an SMS control data set 'ACTIVE' results in errors. Do not allocate control data sets as extended format VSAM linear data sets.

[Source control data set \(SCDS\)](#)

[Active control data set \(ACDS\)](#)

[Communications data set \(COMMDS\)](#)

[Calculating the size of storage and active control data sets](#)

[Calculating the size of a COMMDS](#)

[Selecting volumes for control data sets](#)

[Allocating an SCDS](#)

[Allocating an ACDS](#)

[Allocating a COMMDS](#)

[Recommendations](#)

Parent topic:[Preparing for the Storage Management Subsystem](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Source control data set (SCDS)

z/OS DFSMSdfp Storage Administration
SC23-6860-01

An SCDS contains an SMS configuration, which defines a storage management policy. You can define any number of SMS configurations each of which has its own SCDS. Then, you select one SMS configuration to be the installation storage management policy and make an active working copy of it in an ACDS.

Parent topic:[Allocating control data sets](#)

|

©Copyright IBM Corporation 1990, 2014

Active control data set (ACDS)

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you activate an SCDS, its contents are copied to an ACDS. The current ACDS contains a copy of the most recently activated configuration. All systems in an SMS complex use this configuration to manage storage. You can define any number of SCDSs, but only one can be put in the ACDS. z/OS MVS Initialization and Tuning Guide explains how to specify the ACDS. You can define more than one IGDSMSxx member, each specifying a different ACDS, but you can use only one ACDS at a time.

Tip: You can save the current ACDS as an SCDS using the SETSMS SAVESCDS command. You can also create an ACDS from a SCDS with the SETSMS COPYSCDS command. For more information, see [Parameters of the SETSMS operator command](#).

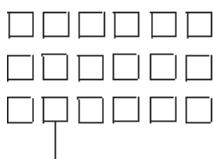
Restriction: You cannot define or alter an ACDS. This also means that you cannot use an ACDS as an SCDS if the SCDS is lost.

You can modify the SCDS from which your current storage management policy was activated without disrupting operations, because SMS manages storage with a copy of the SMS configuration (an ACDS) rather than with the original (an SCDS). While SMS manages storage using an ACDS, you can: Create a backup copy of the SCDSBuild a new SCDSUpdate the SCDS from which the ACDS was activatedModify any SCDSFigure 1 shows the relationship among SCDSs and ACDSs in an installation.

Figure 1. Relationship among SCDSs and ACDSs in an

Several SCDSs, each
containing one
storage management
policy

A few ACDSs



An SCDS is copied into an ACDS

The ACDS contains the
currently active SMS configuration
(storage management policy).
Only one can be active at a time.

Installation

DA 652014

Parent topic:[Allocating control data sets](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Communications data set (COMMDS)

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The COMMDS serves as the primary means of SMS communication among systems in the SMS complex. The active systems in an SMS complex access the COMMDS for current SMS complex information.

The COMMDS contains the name of the ACDS containing the currently active storage management policy, the current utilization statistics for each system-managed volume, and other system information. You can define any number of COMMDSs, but only one can be active in an SMS complex.

Parent topic:[Allocating control data sets](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Calculating the size of storage and active control data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you allocate control data sets, you need to estimate their size. When calculating the size of either an ACDS or an SCDS, you have to account for system and base configuration information, SMS class, aggregate group, storage group, optical library and drive, and tape library definitions.

The following formula can help you determine the size of a source or active control data set:

```
size (bytes) = 150000 +
(14000 * SG) +
(2312 * (MC + SC + DC + AG + CS)) +
(12000 * (DRV + LIB + VOL))
```

where: **150000 bytes** represent fixed data fields. For example, suppose you have the following configuration: Five storage groups (includes copy pools)Four management classesTwo storage classesFive data classesTwo aggregate groups6 optical libraries24 optical drives40 DASD volumesWith this configuration, the equation yields a value of 1088600 bytes for control data set allocation.

SG is the estimated number of storage groups **MC** is the estimated number of management classes **SC** is the estimated number of storage classes **DC** is the estimated number of data classes **AG** is the estimated number of aggregate groups **CS** is the estimated number of cache sets in the base configuration (for record-level sharing (RLS) only) **DRV** is the estimated number of optical drives in the SMS complex to be managed by SMS **LIB** is the estimated number of optical and tape libraries in the SMS complex to be managed by SMS **VOL** is the estimated number of DASD volumes in the SMS complex to be managed by SMS

If you are running SMS in a Parallel Sysplex® environment with different releases of DFSMS, you must allocate your control data sets using calculations that are based on the highest DFSMS level. If you do not, your control data sets might be too small, because the storage requirements for classes and groups might be different between releases or new classes, groups, or other items might be added. On a IBM® 3380 or 9345 DASD each track can contain 40 KB (40960 bytes). On a IBM 3390 DASD each track can contain 48 KB (49156 bytes).

If your SCDS or ACDS is not large enough, you might receive SMS reason code 6068 when attempting to save its contents. When reason code 6068 occurs, allocate a new, larger control data set and copy your existing SCDS into your new SCDS, or your existing ACDS into your new ACDS. You can then delete the old SCDS or ACDS and use the new one.

Allocating a new SCDS or ACDS resolves the problem only when reason code 6068 is caused by a data set size problem. Because this reason code is returned when a system service called by data-in-virtual (DIV) fails, the error might have other causes. Messages returned to you or the system console can help determine the cause of the failure.

Restriction: DIV has a current size limit of 4 GB. Make sure you do not exceed this limit.

Parent topic: Allocating control data sets



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Calculating the size of a COMMDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you calculate the size of a COMMDS, you have to account for both system and volume information. With SMS 32-name support, the amount of space required for a COMMDS increased. A previously allocated COMMDS might have insufficient space to support the changes. You might need to allocate a new COMMDS prior to activating SMS on a current level system, and you should always review the COMMDS size when migrating from prior DFSMS releases. The following formula helps you determine the size of a COMMDS (VOL = estimated number of DASD volumes in the SMS complex to be managed by SMS):

COMMDS size (bytes) = $8192 + (588 * VOL)$

For example, if you have 40 DASD volumes in the SMS complex, you need to allocate 31712 bytes for the COMMDS.

Parent topic:[Allocating control data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Selecting volumes for control data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SMS control data sets can be either SMS-managed or non-SMS-managed. Initially you should ensure that your control data sets have a volume count of one. The volume count can be either explicitly specified, implied by the number of volume serials provided, or derived from the data class assigned to the data set (the greater value of volume count or dynamic volume count). See [Defining data classes](#) for more information. For SMS control data sets, the volume count or dynamic volume count must be less than or equal to the number of storage volumes.

Otherwise, you might receive messages IEF244I and IEF877E.

If you have a multivolume SCDS that you are activating into a single volume ACDS, you might receive an error because the ACDS is not large enough and volumes cannot be dynamically added to it. To bypass this problem, you need to create a new multivolume ACDS and then activate the ACDS and SCDS simultaneously using the SETSMS command. See [Changing Storage Management Subsystem parameters](#) for further information on this command.

If your SMS complex includes more than 16 systems, be sure that the ACDS and COMMDS are accessible to every system in the complex. Define your control data sets on volumes that are capable of being attached to more than 16 systems, such as IBM® RAMAC Virtual Array volumes.

Parent topic:[Allocating control data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Allocating an SCDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following access method services job allocates a 6-track SCDS.

```
PGM=IDCAMS //STEP EXEC
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFINE CLUSTER(NAME(SMS.SCDS1.SCDS) LINEAR VOL(SMSV01) -
TRK(6 6) SHAREOPTIONS(3,3)) -
DATA(NAME(SMS.SCDS1.SCDS.DATA)REUSE)
/*
```

This job creates a VSAM linear data set named SMS.SCDS1.SCDS. You can combine the DEFINE commands for all your allocations into the same job step, but this example shows only one for purposes of illustration. After allocating an SCDS, you define its contents through ISMF dialogs.

You should allocate an SCDS on a device shared by all systems in the SMS complex. If you allocate an SCDS on a device that is not shared by all the systems, then you can activate the SCDS only from systems that have access to it.

You should specify the REUSE option when you define an SCDS to avoid running into space problems (SMS reason code 6068) as result of subsequent SCDS updates, or IMPORT/EXPORT functions.

See z/OS DFSMS Access Method Services Commands for information on using access method services commands.

Parent topic:[Allocating control data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Allocating an ACDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following access method services job allocates a 6-track ACDS.

```
PGM=IDCAMS //STEP EXEC
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE CLUSTER(NAME(SMS.ACDS1.ACDS) LINEAR VOL(SMSV02) -
    TRK(6 6) SHAREOPTIONS(2,3) -
    DATA(NAME(SMS.ACDS1.ACDS.DATA)REUSE)
/*
```

This job creates a VSAM linear data set named SMS.ACDS1.ACDS.

An ACDS must reside on a shared volume, accessible from all systems in the SMS complex. To ease recovery in case of failure, the ACDS should reside on a different volume than the COMMDS. Also, you should allocate a spare ACDS on a different shared volume. [Recovering Storage Management Subsystem information](#) provides additional information on the backup and recovery of control data sets.

You create the contents of an ACDS by activating a valid SCDS. The distinction between a valid SCDS and one that is not valid is described in [Defining the base configuration](#). The control data set (ACDS or COMMDS) must reside on a volume that is not reserved by other systems for a long period of time because the control data set (ACDS or COMMDS) must be available to access for SMS processing to continue.

You should specify the REUSE option when you define an ACDS to avoid running into space problems (SMS reason code 6068) as result of subsequent ACDS updates, or IMPORT/EXPORT functions.

Parent topic:[Allocating control data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Allocating a COMMDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following access method services job allocates a 1-track COMMDS;

//STEP

```
EXEC PGM=IDCAMS
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
  DEFINE CLUSTER(NAME(SMS.COMMDS1.COMMDS) LINEAR VOL(SMSVOL) -
    TRK(1 1) SHAREOPTIONS(3,3)) -
    DATA(NAME(SMS.COMMDS1.COMMDS.DATA)REUSE)
/*
```

This job creates a VSAM linear data set named SMS.COMMDS1.COMMDS.

The COMMDS must reside on a shared volume accessible from all systems in the SMS complex. To ease recovery in case of failure, the COMMDS should reside on a different volume than the ACDS. Also, you should allocate a spare COMMDS on a different shared volume. [Recovering Storage Management Subsystem information](#) provides additional information on the backup and recovery of control data sets. The control data set (ACDS or COMMDS) must reside on a volume that is not reserved by other systems for a long period of time because the control data set (ACDS or COMMDS) must be available to access for SMS processing to continue. Recommendations:Specify the REUSE option when defining the COMMDS data set to help avoid space problems (SMS reason code 6068) during subsequent COMMDS updates or IMPORT/EXPORT functions.The COMMDS and ACDS must be accessed from all systems in the complex simultaneously.

Parent topic:[Allocating control data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recommendations

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Specify the REUSE option when defining the COMMDS data set to help avoid space problems (SMS reason code 6068) during subsequent COMMDS updates or IMPORT/EXPORT functions. Use SHAREOPTIONS(3,3) when allocating an ACDS or COMMDS. This allows full authority to read from and write to the ACDS or COMMDS from any system. The ACDS and COMMDS must be accessed from all systems in the complex simultaneously. If SMS detects a lower value for SHAREOPTIONS when the CDS is activated, SMS attempts to alter the value to SHAREOPTIONS(3,3).

Parent topic:[Allocating control data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Modifying the SYS1.PARMLIB data set

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The IGDSMSxx, IEASYSyy, and IEFSSNxx members of SYS1.PARMLIB direct the initialization and activation of SMS. IGDSMSxx provides initialization parameters to SMS. The SMS=xx parameter of IEASYSyy indicates the name of the SYS1.PARMLIB member IGDSMSxx that is used for initialization. For example, if SMS=01 in IEASYSyy, then the IGDSMS01 member of SYS1.PARMLIB is used during initialization. The SMS entry in IEFSSNxx identifies SMS to z/OS®. For information about how to create an IGDSMSxx member in SYS1.PARMLIB and how to define SMS to z/OS through IEFSSNxx, see [z/OS MVS Initialization and Tuning Reference](#).

[Initializing SMS through the IGDSMSxx member](#)

[Starting the SMS address space](#)

Parent topic:[Preparing for the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Initializing SMS through the IGDSMSxx member

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Every SMS system must have an IGDSMS xx member in SYS1.PARMLIB that specifies a required ACDS and COMMDS pair. This ACDS and COMMDS pair is used if the COMMDS of the pair does not point to another COMMDS. If the COMMDS points to another COMMDS, the referenced COMMDS is used. This referenced COMMDS might contain the name of an ACDS that is different from the one specified in the SYS1.PARMLIB member. If so, the name of the ACDS is obtained from the COMMDS rather than from the IGDSMS xx member in SYS1.PARMLIB to ensure that the system is always running under the most recent ACDS and COMMDS. If the COMMDS of the ACDS and COMMDS pair refers to another COMMDS during IPL, it means a more recent COMMDS has been used. SMS uses the most recent COMMDS to ensure that you cannot IPL with a down-level configuration.

Parent topic:[Modifying the SYS1.PARMLIB data set](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Starting the SMS address space

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you have completed preparations and are ready to start SMS, use the T SMS=xx command, where xx identifies IGDSMSxx as the SMS initialization member. The T SMS=xx command is an abbreviation for the SET SMS=xx command, discussed in [Step two: prepare one system](#). To eliminate confusion with the SETSMS operator command, the abbreviated T SMS=xx form of the SET SMS=xx command is used throughout the remainder of this information.

When you have sufficiently tested your operations and are ready to have SMS automatically started at future IPLs, add the IGDSSIIN module name to the SMS entry.

Here are some examples of the SMS record: The following SMS record defines SMS to z/OS® without starting SMS at IPLs:

SUBSYS SUBNAME(SMS)

The following SMS record defines SMS to z/OS and starts SMS at future IPLs:

SUBSYS

SUBNAME(SMS) INITRTN(IGDSSIIN)

The system uses the default values of ID and PROMPT. IGDSMS00 specifies initialization information, and the operator has no control over the rest of SMS initialization. The following SMS record allows the operator to modify SMS initialization:

SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN)
INITPARM('PROMPT=YES')

The system uses the default value of ID, which identifies IGDSMS00 as containing initialization information. The PROMPT parameter requests that SMS display IGDSMS00, so that the operator can modify the parameters in IGDSMS00. The following SMS record initializes SMS using IGDSMS01:

SUBSYS SUBNAME(SMS) INITRTN(IGDSSIIN)
INITPARM('ID=01,PROMPT=DISPLAY')

The PROMPT parameter requests that the contents of IGDSMS01 be displayed, but the operator cannot modify them.

Parent topic:[Modifying the SYS1.PARMLIB data set](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Accessing the storage administrator primary option menu

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The first time you select ISMF, you get the ISMF Primary Option Menu for *end users*. To get the ISMF Primary Option Menu for *storage administrators* (which is shown in [Figure 1](#)), select option 0, ISMF PROFILE. Within the ISMF Profile Option Menu, select option 0, USER MODE, and press ENTER. You get the User Mode Entry panel, where you indicate that you want the storage administrator Primary Option Menu for all future ISMF sessions. To do this, select option 2 on the User Mode Entry panel. After changing the user mode, you must exit ISMF and then return to it to view the Primary Option Menu for Storage Administrators.

[Protecting the Storage Management Subsystem](#) explains how to prevent end users from gaining access to the storage administrator Primary Option Menu through the ISMF PROFILE option. The reason for restricting access to the Primary Option Menu for Storage Administrators is to prevent unauthorized users from performing storage administrator tasks.

Parent topic:[Preparing for the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Planning for VSAM record-level sharing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Planning for and installing VSAM record-level sharing requires coordination with system hardware and software groups. The following planning tasks are described fully in [Administering VSAM record-level sharing: Determining hardware requirements](#)
[Determining applications that can use VSAM RLS](#)
[Ensuring same systems connectivity](#)
[Planning for availability](#)
[Defining sharing control data sets](#)
[Defining CF cache structures](#)
[Defining the primary CF lock structure](#)
[Modifying the SYS1.PARMLIB IGDSMSxx member](#)
[Establishing authorization for VSAM RLS](#)

Parent topic:[Preparing for the Storage Management Subsystem](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Creating the base configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The first thing that you define in an SCDS is the base configuration. A base configuration contains installation defaults, such as a default management class, and identifies the systems to which the SMS configuration applies. This topic describes the base configuration and explains how to define its contents using the ISMF control data set application.

[Planning the base configuration](#)

[Defining the base configuration](#)

[Defining the base configuration for VSAM record-level sharing](#)

[Completing the base configuration](#)

 | 

©Copyright IBM Corporation 1990, 2014

Planning the base configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You need to determine the system names and system group names that you want to specify in the SCDS. These systems and system groups constitute the SMS complex.

Before defining the base configuration, consider what you want to do with system-managed data sets that do not have a management class. You can specify a default management class for these data sets in the base configuration. You can create the management class any time before validating the SCDS. Defining the management classes is described in [Defining management classes](#).

If the management class ACS routine does not determine a management class for a data set, DFSMShsm processes the data set using the default management class, if one exists. If you do not specify a default management class in your base configuration, DFSMShsm uses its own defaults.

If the management class ACS routine does not determine a management class for an object, the OSREQ request fails. This request can be OSREQ STORE, CHANGE, or OSMC class transition processing.

See z/OS DFSMS Implementing System-Managed Storage for additional planning information.

Parent topic:[Creating the base configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining the base configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: You must allocate an SCDS.

Perform the following steps to define a base configuration.

Select option 8, Control Data Set, from the ISMF Primary Option Menu for Storage Administrator. This displays the CDS Application Selection panel.

Supply values on the CDS Application Selection panel: **CDS name**This is the name of the SCDS that is to contain the base configuration. ISMF primes the CDS Name field with the last used SCDS name**Option**Select option 2, Define. This displays the SCDS Base Define panel. You can also: See the base configuration, by selecting option 1, Display. You can use this option to look at the base configuration of the current active SMS configuration by specifying 'ACTIVE' in the CDS Name field.Alter an SCDS base configuration that you defined previously, by selecting option 3, Alter. The Alter panel contains the same fields as the Define panel.

Supply values on the SCDS Base Define panel. Use the DOWN command to view the second page of the panel.

DescriptionYou can use the field to help you identify and describe the SCDS.

Your description can be up to 120 characters.**Default Management Class**See

[Specifying the default management class](#).**Default Unit**See [Specifying the default unit](#).

Default Device GeometrySee [Specifying the default device geometry](#).

DS Separation Profile (Data Set Name)See [Specifying the DS separation profile](#).

Systems and System Groups in the SMS ComplexSee [Specifying systems and system groups in the SMS complex](#).

If at any time you want to leave either the SCDS Base Define panel without saving the changed base configuration information, issue the CANCEL command.

When you have supplied all of the values, use the END command to save them and return to the CDS Application Selection panel.

Validate and activate the SCDS, as desired. See [Validating the SCDS](#) and [Activating the SCDS](#).

Requirements and Restrictions:The SMS configuration supports up to 32 system names, system group names, or both. When the system is running in compatibility mode (8-name mode) you can specify a maximum of eight system names, system group names or both in the SMS configuration. When the system is running in 32-name mode, you can specify a maximum of 32 system, system group names or both. The current system mode is defined in the IGDSMSxx member of SYS1.PARMLIB, specified when SMS is started (see [Preparing for the Storage Management Subsystem](#)). If SMS is not active, the number of systems you can put in the SCDS is determined by the SCDS itself. If it is an 8-name configuration, you can only put eight names in it. If it is a 32-name configuration, you can put up to 32 names in it. If it is an empty data set, it defaults to 8-name mode.

When you access an SMS control data set (SCDS, ACDS, or COMMDS) for update which supports only eight names on a system running in 32-name mode, the data set must be converted to a new, incompatible format in order to support 32 names. You must confirm this conversion, either through the operator console or ISMF. This conversion is permanent, so you should make copies of your control data sets before converting the system mode from compatibility mode to 32-name mode.

[Validating the SCDS](#)

[Activating the SCDS](#)

[Specifying the default management class](#)

[Specifying the default unit](#)

[Specifying the default device geometry](#)

[Specifying the DS separation profile](#)

[Specifying the DS separation profile](#)
[Specifying systems and system groups in the SMS complex](#)
Parent topic:[Creating the base configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Validating the SCDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To validate the SCDS or any of the ACS routines, select option 4, Validate, from the CDS Application Selection panel. This displays the status of the SCDS.

You should validate an SCDS before you activate it. The SCDS Status field tells you whether the SCDS is valid or not valid. SMS sets the status when you save an SCDS with the END command or when you select the validate option from the CDS Application Selection panel.

[Validating an entire SCDS](#) lists conditions that cause an SCDS to be not valid.

[Validating ACS routines or an entire SCDS](#) provides more information about the validate option.

Parent topic:[Defining the base configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Activating the SCDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To activate the SCDS, select option 5, Activate, from the CDS Application Selection panel.

You must activate the configuration for any changes made to the SCDS to take effect.

Parent topic:[Defining the base configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying the default management class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For system-managed data sets that have not been assigned a management class, DFSMShsm uses the default management class for expiration, migration, and backup information. The default management class does not apply to objects. You specify the name of the management class in the Default Management Class field. The default management class name is not saved in the data set catalog entry, so that you can tell the difference between data sets with assigned management classes and those without them. Specifying a default management class is optional. If a data set has no management classes and a default management class is not defined, DFSMShsm uses its own defaults for the data set.

Parent topic:[Defining the base configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying the default unit

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The default unit is an esoteric or generic device name, such as SYSDA or 3390, that applies to data sets that are not managed by SMS. For new data set allocations, the default unit allows users to omit the UNIT parameter from DD statements or dynamic allocation equivalent, as they can when allocating system-managed data sets. The default unit does not apply to objects.

If users do not specify the unit parameter on their DD statements or the dynamic allocation equivalent, SMS applies the default unit if the data set is non-system managed and has a disposition of either MOD (treated as NEW) or NEW. If you specify a default unit in the base configuration, make certain that it exists on the system performing the allocations. If you do not specify a default unit, end users must code the UNIT parameter to allocate data sets that are not system-managed.

Parent topic:[Defining the base configuration](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying the default device geometry

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When allocating space for a new data set on DASD, SMS converts all space requests in tracks (TRK) or cylinders (CYL) into requests for space in KB or MB. If a generic device type such as the 3380 is specified, SMS uses the device geometry for that generic device to convert tracks or cylinders into KB or MB. If an esoteric device type such as SYSDA or no UNIT is specified, SMS uses the default device geometry to convert tracks and cylinders into KB or MB. If the users in your installations specify space in tracks or cylinder units, and they specify an esoteric UNIT or no UNIT, you must specify a default device geometry prior to converting these allocations to system-managed data sets.

After SMS converts space requests to KB or MB, the space values are passed to the ACS routines. The values are later used to determine the number of tracks or cylinders to allocate for the data set. The default device geometry does not apply to objects or data sets allocated on tape.

There is only one default device geometry for the entire SMS complex. Default device geometry is an installation's definition of how much space is represented by a TRK or a CYL when an esoteric unit or no unit is specified. The device *geometry* is the track size and number of tracks per cylinder for the device.

The device geometry for 3380 is 47476 bytes per track, 15 tracks per cylinder. The device geometry for 3390 is 56664 bytes per track, 15 tracks per cylinder. It is up to each installation to decide what values to use.

Parent topic:[Defining the base configuration](#)



©Copyright IBM Corporation 1990, 2014

Specifying the DS separation profile

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Data set separation allows you to designate groups of data sets in which all SMS-managed data sets within a group are kept separate, on the physical control unit (PCU) or volume level, from all the other data sets in the same group. This reduces the effects of single points of failure. Using the volume level reduces I/O contention. To use data set separation, you create a data set separation profile and specify the data set name of the profile in the base configuration. DS separation profile is an optional field that provides SMS with the data set name of the provided data set separation profile. Before you can specify this data set name, you must have created a data set separation profile. See [Using data set separation](#) to learn how to create a data set separation profile.

You can specify any valid sequential or partitioned member data set name, with a maximum length of 56 characters, with or without quotation marks. For data set names without quotation marks, ISMF will add the TSO user ID prefix of the person who is defining or updating the base configuration.

Recommendation: Use a data set name that contains quotation marks.

The default value is **blank**, which indicates that a data set separation is not requested for the SMS complex.

Parent topic:[Defining the base configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying systems and system groups in the SMS complex

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can add, delete, or rename the systems or system groups defined to an SMS complex using page 2 of the SCDS Base Define panel.

A Parallel Sysplex® is a collection of systems linked by closely coupled hardware facilities to process workloads. In SMS, the term *system group* is used instead of Parallel Sysplex. SMS system group name support allows you to specify a system group as a member of an SMS complex. A system group consists of system names within a Parallel Sysplex, excluding those systems that are individually specified in the SMS base configuration. A system group name can represent multiple systems. This allows SMS to support more than eight systems per SMS complex while retaining the existing configuration format of the configuration data set.

A system group name matches a Parallel Sysplex name and refers to all systems defined as part of the Parallel Sysplex that are: Running the same SMS configurationDefined in the configuration using the name of the Parallel Sysplex to which they belong (that is, system group)NOT defined in the configuration by their system names For a discussion on running SMS in a Parallel Sysplex and special considerations to observe, see [Running SMS in a parallel sysplex environment](#).

On the SCDS Base Define panel, the System Name and Sys Group fields display the systems or system groups that you have previously defined. The system or group names appear in alphabetical order. You must add at least one system name or system group name to a base configuration before you can save it in the SCDS, and you can define up to 32 system names, system group names, or both, to the SMS complex.

The system name must match the value of the SYSNAME parameter in the IEASYSyy member of the SYS1.PARMLIB used on that system. The system group name must match the value of the SYSPLEX parameter in the COUPLExx (XCF) member of the SYS1.PARMLIB used on that system. To add a system name or a system group to the SCDS base configuration: Choose option 1, Add, and enter the system name in the System Name field or system group name in the Sys Group Name field. Press Enter to add the system.

➤**Restriction:** In a JES3 environment, system group name is not supported due to the fact that the system features and resources that are used to determine on which eligible system a job will be run are only applicable to each system, not to a system group. ◀

To delete a system or a system group from the base configuration: Choose option 2, Delete, and enter the system name in the System Name field or enter the system group name in the Sys Group Name field. Press Enter to delete the system or system group name from the SCDS. You are asked to confirm that you want the system or system group deleted before it is deleted.

To rename a system or a system group in a base configuration: Choose option 3, Rename, enter the current name of the system in the System Name or Sys Group Name field, and enter the new name of the system in the New System/Sys Group Name field. Press Enter to rename the system or system group.

Tip: The Rename option cannot be used to change the type of name. To change a system to a system group, or a system group to a system, you must delete the name, then add it as the desired type.

See [Recovering from a systems failure in the SMS complex](#) for information on recovering from a systems failure in the SMS complex.

Parent topic:[Defining the base configuration](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining the base configuration for VSAM record-level sharing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In order for DFSMSdfp to use the coupling facility (CF) for VSAM RLS, you must add CF cache structures to the SMS base configuration and define the cache sets with which they are associated.

[Using cache structures](#)

[Defining cache sets](#)

[Defining lock sets](#)

Parent topic:[Creating the base configuration](#)



©Copyright IBM Corporation 1990, 2014

Using cache structures

z/OS DFSMSdfp Storage Administration
SC23-6860-01

CF cache structures are defined to z/OS® using coupling facility resource management (CFRM) policies, which determine how and where the structures are allocated. You associate these cache structures with a cache set name in the base configuration. The cache set name is also specified in a storage class definition. When a storage class associated with a data set contains a cache set name, the data set becomes eligible for VSAM record-level sharing and can be placed in a CF cache structure associated with the cache set. The system selects the best cache structure within the cache set defined for the storage class.

CF cache structures must have the same system connectivity as any storage groups that might be assigned to those cache structures. For example, if Storage Class 1 maps to cache set CS1, then the CF structures in CS1 must have the same system connectivity as the storage groups with which Storage Class 1 is associated. Connectivity of CF cache structures to all systems in the Parallel Sysplex® simplifies managing and changing the configuration.

Recommendation: In a JES3 environment, be careful to define cache set names only in those SMS storage classes that are used by data sets opened for VSAM RLS processing. When you define a cache set name in a storage class, any job accessing a data set associated with that storage class is scheduled on a VSAM RLS-capable system (one where the SMSVSAM address space has been successfully initialized). If all storage classes have cache set names defined for them, then all jobs accessing SMS-managed data sets are scheduled to VSAM-RLS-capable systems. This could cause a workload imbalance between those systems and down-level systems.

See [Defining CF cache structures](#) for more information on defining CF cache structures. See [Defining the base configuration](#) for more information on defining the base configuration.

Parent topic:[Defining the base configuration for VSAM record-level sharing](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining cache sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: You must define Cache structures to z/OS® in a CFRM policy. Perform the following steps to define a cache set and specify the cache structures associated with it.

Select option 8, Control Data Set, from the ISMF Primary Option Menu for Storage Administrators. This displays the Control Data Set (CDS) Application Selection panel.

Supply values on the Control Data Set (CDS) Application Selection panel: **CDS Name**This is the name of the SCDS that is to contain the base configuration for VSAM RLS. The CDS Name field is primed with the last used SCDS.**Option**Select option 7, Cache Update. This displays the CF Cache Set Update panel.

Supply values on the CF Cache Set Update panel: **Cache Set**This is the name of a cache set. You can define up to 256 cache set names.**CF Cache Structure Names**These are the names of all CF cache structures associated with the cache set. You can specify up to eight CF cache structures for each cache set. Cache structures must be previously defined to z/OS in a CFRM policy.

Parent topic:[Defining the base configuration for VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining lock sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define a lock set and specify the lock set structure associated with it.

Select option 8, Control Data Set, from the ISMF Primary Option Menu for Storage Administrators. This displays the CDS Application Selection panel.

Supply values on the CDS Application Selection panel: **CDS Name**This is the name of the SCDS that is to contain the base configuration for VSAM RLS. ISMF primes the CDS Name field with the last used SCDS name.**Option**Select option 9, Lock Update. This displays the CF Lock Set Update panel.

Supply values on the CF Lock Set Update panel: **Lock Set**This is the name of a lock set. You can define up to 256 lock set names.**CF Lock Structure**This is the name of the CF lock structure associated with the lock set. You can specify a CF lock structure for each lock set.

Parent topic:[Defining the base configuration for VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Completing the base configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

After defining base configuration attributes, you can verify their completeness and correctness by pressing Enter. If your SCDS base configuration contains any errors, the cursor moves to the field in the error and an error message appears in the short message area. Correct any errors and press Enter to verify the new contents of the SCDS base configuration.

After correcting all errors, use the END command to save the SCDS base configuration.

If at any time you want to leave either the SCDS Base Define or Alter panel without saving the changed base configuration information, issue the Cancel command.

Parent topic:[Creating the base configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When managing non-system-managed DASD volumes, you view and maintain them as individual devices. If you have too many critical data sets on a volume, you have to spread the data sets across other volumes to remove I/O bottlenecks. If you are allocating data sets manually, you might not be using the volume as efficiently as you could. Some volumes containing critical data sets might be under-used. Other volumes with lower activity data sets might be overpopulated and require extra storage administrator attention to ensure that space is preserved for new data set allocations and extensions. In these and other instances, you have to work on individual volumes to solve your problems.

SMS simplifies the management of DASD, mountable optical volumes, and tape volumes by pooling them together in storage groups. This topic describes storage groups and shows you how to define them using the ISMF Storage Group Application.

[Understanding storage groups](#)

[Planning storage groups for data sets](#)

[Planning storage groups for OAM object collections](#)

[Defining storage group attributes](#)

[Defining additional storage groups](#)

[Listing volumes in a storage group](#)



©Copyright IBM Corporation 1990, 2014

Understanding storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Storage groups represent the physical storage managed by SMS. This storage can be collections of DASD volumes, volumes in tape libraries, volumes in optical libraries, or virtual input/output (VIO) storage. A storage group, used with storage classes, separates the logical requirements of accessing data from the physical requirements to store the data. You can use storage group attributes to specify how the system should manage the storage group. You use the storage group ACS routine to assign a new data set or object to a storage group. You can assign multiple candidate storage groups (except for objects), in which case the system chooses a specific storage group from your list. Storage group definitions are not apparent to users. Only you, as storage administrator, can define, alter, or display storage group definitions.

A storage group can be VIO, dummy, copy pool backup, pool, object, object backup, or tape. VIO storage groups are not associated with volumes. Dummy storage groups are associated with nonexistent volumes. When defining pool storage groups, the actual, physical paths connecting systems to DASD volumes must match the desired logical paths specified in the storage group definitions. Merely establishing a physical connection from a system to a DASD volume does not provide access. Within the storage group definition, you must specify which systems have access to which storage groups, and which storage groups have access to which DASD volumes. Likewise, merely defining a system to have access to a DASD volume does not establish a physical connection. The physical connection must exist.

For tape storage groups, one or more tape libraries are associated with them. Connectivity is defined at both the library level and the storage group level. If a storage group is connected to certain systems, then any libraries associated with that storage group must be connected to the same systems. Scratch volumes are added to storage groups when they are used. Private volumes can be added when they are entered in a library. Private volumes are removed from a storage group and returned to the common scratch pool when they are returned to scratch status. For more information on how to use storage groups with objects, see [Object and object backup storage groups](#).

[DASD storage groups](#)

[DASD volume status for data sets](#)

[Object and object backup storage groups](#)

[OAM collection names](#)

[Tape storage groups](#)

Parent topic:[Defining storage groups](#)

©Copyright IBM Corporation 1990, 2014

DASD storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you define a pool storage group, you specify the volume serial numbers of the DASD volumes you are including in the storage group rather than their physical addresses. Each DASD volume in a storage group must contain a VSAM volume data set (VVDS) and an indexed volume table of contents (VTOC). The VVDS is automatically created after allocation of the first system-managed data set on a volume, if the VVDS is not already defined.

Two storage groups cannot share a DASD volume. You must define an entire volume to a single pool storage group. Also, a data set can only reside in one pool storage group. A data set can span volumes within a single pool storage group, but it cannot span volumes belonging to several pool storage groups. For VSAM data sets, the entire sphere (base cluster and all alternate indexes) must be in the same storage group.

Recommendation: Define pool storage groups so that they only contain devices of the same geometry. The device *geometry* is the track size and number of tracks per cylinder for the device. 3390 devices in 3380 track compatibility mode are geometrically the same as 3380 devices, and the access methods see them as 3380 devices. Therefore, you can combine these devices in a single storage group.

Requirement: To enable extend processing in the extend storage group, ensure that the extend storage group contains devices with the same geometry as the initial storage group.

Although you can separate devices according to geometry, you do not need to separate them according to capacity. For example, you can combine all models of the 3390 into a single storage group. The only effect the different capacities has is on volume thresholds. See *z/OS DFSMS Implementing System-Managed Storage* for information on selecting appropriate threshold levels.

Devices of the same geometry can have different performance characteristics. These devices coexist in the same storage group, and enhanced volume selection for SMS manages data set placement accordingly. Even devices with vastly different performance characteristics can reside in the same storage group.

For striped data sets through sequential access, the effective data rate is limited by the slowest stripe. If a storage group contains volumes with widely different data delivery capabilities, such as 3990 Model 6 and ESS, the effective data rate for striped data sets through sequential accesses is gated by the stripes on 3990 Model 6.

Parent topic:[Understanding storage groups](#)

©Copyright IBM Corporation 1990, 2014

DASD volume status for data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To prepare new DASD volumes for SMS storage groups, use the Device Support Facilities (ICKDSF) INIT command to assign new DASD volume serial numbers and allocate an indexed VTOC. You can use the STORAGEGROUP (STGR) keyword of the INIT command to make a DASD volume available for allocation of new system-managed data sets.

You can use INIT from the ISMF Volume Application and the Storage Group (through LISTVOL) Application. If you go to the Volume Application directly from the Primary Option Menu, you initialize the DASD volume as a non-system-managed volume. If, instead, you work from a DASD volume list generated within the Storage Group Application, you initialize the DASD volume as a system-managed volume. When new volumes are added to a storage group in the active configuration, we recommend explicitly allocating small temporary data sets to each new volume to update the SMS volume space statistics.

As you bring DASD volumes under the control of SMS, you need to keep track of those that are ready to be system-managed. The physical DASD volume status can be one of the following: converted, initial, non-SMS, or unknown. To display this information, on the Storage Group Application panel, enter the LISTVOL command against pool storage groups.

Convert status indicates that the DASD volume is converted and is fully available for SMS control. To be converted, all data sets on the DASD volume must have an associated storage class, and all permanent data sets on the DASD volume must be cataloged in an integrated catalog facility (ICF) catalog. The DASD volume must have a VTOC index.

Initial status indicates that an attempt to convert the DASD volume has been made, but the DASD volume contains data sets that fail to satisfy the requirements of a converted volume. In the initial state, no new allocations can be made on the DASD volume. You cannot extend data sets to additional DASD volumes. The initial status allows DFSMSdss to process the DASD volume without having to issue a RESERVE for the entire duration of the conversion process. Also, to be eligible for conversion, data sets and the DASD volume must have an organization that is supported by SMS. For more information, refer to z/OS DFSMS Implementing System-Managed Storage.

Parent topic:[Understanding storage groups](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Object and object backup storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

An object storage group is a storage group that defines the physical storage used for objects. For objects, the storage group allows you to define an object storage hierarchy. The object storage hierarchy can consist of: Disk sublevel 1, which is associated with DB2 tables on a direct access storage device (DASD). The following object storage DB2 tables provide disk sublevel 1 storage for objects: 4 KB storage table32 KB storage tableLOB storage structureDisk sublevel 2, which is associated with an NFS or zFS file system.Tape sublevel 1 volumes associated with a tape library device (SMS-managed, library-resident tape volumes), and tape volumes outside of a library device (non-SMS-managed, shelf-resident tape volumes)Tape sublevel 2 volumes associated with a tape library device (SMS-managed, library-resident tape volumes), and tape volumes outside of a library device (non-SMS-managed, shelf-resident tape volumes)Optical volumes inside a library device (SMS-managed, library-resident optical volumes), and optical volumes outside of a library device (SMS-managed, shelf-resident optical volumes) You can define an object storage group that does not include an optical library. If an object storage group does not include an optical library, the objects reside on DASD volumes or tape volumes and do not get moved to optical volumes.

An object can move up and down the hierarchy within a single storage group but cannot move outside of that group. Objects move within the storage hierarchy depending initially on management class and storage class criteria for the object collection, and subsequently on ACS routines.

You can define two types of storage groups for the object access method (OAM): object and object backup. An object storage group defines an object storage hierarchy. The object backup storage groups define the optical libraries or tape units that are used for backing up objects. An object or object backup storage group can be enabled by and connected to more than one system in a Parallel Sysplex®. There is no limit on the number of object or object backup storage groups defined to each system in the SMS complex. Up to two object backup storage groups can be associated with each object storage group.

You can specify two timing attributes for object and object backup storage groups: cycle start time and cycle end time. These attributes define a window of time during which the OAM Storage Management Component (OSMC) can automatically start its storage management processing for the storage group. The window occurs once each day.

You can also specify an OSMC processing system name. This attribute identifies the OAM system in the sysplex that performs OSMC processing for the object or object backup storage group. Specifying the OSMC processing system name prevents multiple systems from trying to process the same storage group at the same time. This is particularly useful when you use the cycle start time attribute to automatically start storage group processing.

To protect objects in an object storage group from deletion: Specify deletion-protection mode. Objects cannot be deleted prior to their expiration date.Specify retention-protection mode. Objects cannot be deleted prior to their expiration date, and the expiration date can never be changed to an earlier date. For more information, see [Defining an object storage group](#).

See z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support for more information about managing objects.

[Defining real and pseudo optical libraries in object storage groups](#)

[Defining optical libraries in object backup storage groups](#)

[Writing objects](#)

Parent topic:[Understanding storage groups](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining real and pseudo optical libraries in object storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When defining an object storage group, you can specify up to eight real optical libraries or up to eight pseudo optical libraries. You cannot mix real and pseudo libraries in a storage group definition. If the object's storage class indicates that the object should reside on optical, OAM stores the object using any optical drive associated with any of the optical libraries listed in the object storage group definition, and any eligible optical volume that is accessible by the drive selected. The object or object backup storage groups can share optical libraries. A single optical library can contain optical volumes belonging to several object or object backup storage groups. There can be multiple object storage groups per system.

Parent topic:[Object and object backup storage groups](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining optical libraries in object backup storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When defining an object backup storage group, you can specify up to eight real optical libraries or up to eight pseudo optical libraries.

OAM stores the backup copy or copies of the object using any optical drive and optical volume that is associated with any one of the optical libraries listed in the object backup storage group definition.

If no tapeunitname is specified in the CBROAMxx member of PARMLIB for this object backup storage group, OAM stores the backup copies to optical. Otherwise, OAM stores the backup copy or copies of the object using the tapeunitname specified in the CBROAMxx PARMLIB member.

See [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support](#) for more information about defining optical libraries in object and object backup storage groups.

Parent topic:[Object and object backup storage groups](#)



©Copyright IBM Corporation 1990, 2014

Writing objects

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The storage class for an object determines where an object is written (DB2 table, file system, optical, or tape). Objects can be written to tape using the media type and tape unit name defined in the SETOAM statements in the CBROAMxx member of PARMLIB.

When the storage class indicates the object should reside on optical, objects can be written using any drive in any library associated with the object storage group, using any volume assigned to that storage group, as long as there is enough space to satisfy the request and the volume is write-compatible with the drive. If no group volumes are available, the request is satisfied using a scratch volume residing in a library associated with the object storage group.

When the storage class indicates the object should reside on tape, the object can be written using any volume assigned to that storage group, using the unit name in the tape volume record for allocation of a tape device. If no group volumes are available, the request is satisfied using a scratch volume allocated using the tape unit name specified for that storage group in the SETOAM command in the CBROAMxx member of PARMLIB.

Both object and object backup storage groups have three attributes that are concerned with writing objects: The drive startup thresholdThe volume full threshold or tape full thresholdThe mark volume full on first write failure option The attributes are specified in the CBROAMxx member of PARMLIB.

The *drive startup threshold* is the maximum number of write requests (for optical) or maximum KB of object data (for tape) that can be waiting for each drive that is currently processing write requests for the storage group. If the threshold is specified and then exceeded, an attempt is made to start another drive to process write requests to the storage group.

When the number of free KB on a volume falls below the *volume full threshold* specified for the storage group, the volume is marked full. Similarly, when the number of free KB on a volume falls below the *tape full threshold*, the tape volume is marked full. If you specified *mark volume full on first write failure*, the volume is marked as full the first time an attempt to write an object on the volume fails because not enough space remains on the volume.

See z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support for more information about writing objects.

Parent topic:[Object and object backup storage groups](#)



OAM collection names

z/OS DFSMSdfp Storage Administration
SC23-6860-01

OAM collection names provide another level of grouping called a *collection*. A collection is a group of objects that share a common storage group and have the same default initial storage class and management class attributes. All objects within a collection must have unique names. However, you can have two objects with identical names if the objects belong to separate collections.

OAM collections are cataloged in integrated catalog facility (ICF) catalogs using OAM collection names. An individual object does not have an entry in the catalog. Each object is a member of a collection and each collection is part of one storage group. A collection does **not** span storage groups. OAM identifies an object by its collection name and its object name. An object is described by an entry in a DB2® object directory. A collection is described by an MVS collection name catalog entry and a corresponding OAM collection identifier table entry.

OAM provides two primary interfaces for processing objects: the application program interface (OSREQ) and the OAM Storage Management Component (OSMC). Both address objects using the object's collection name and object name including those cases where the object must be identified to the ACS routines for the STORE (OSREQ), CHANGE (OSREQ), and CTRANS (OSMC class transition) environments. Refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *z/OS DFSMS OAM Application Programmer's Reference* for more information.

Parent topic:[Understanding storage groups](#)



©Copyright IBM Corporation 1990, 2014

Tape storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The Storage Group Application supports a tape storage group type for tape libraries. The LISTVOL line operator on the Storage Group List supports the mountable tape volume list. Note: The information provided by LISTVOL differs from that provided by LISTSYS, because the information for the two commands comes from different sources.

The storage administrator can specify tape as the storage group type and can then define and alter tape storage groups and their relationships to the systems in the SMS complex. These relationships are NOTCON, ENABLE, DISALL, DISNEW, QUIALL, or QUINew. See [Defining system access to pool and VIO storage groups](#) for an explanation of these relationships.

A tape storage group is a collection of tape cartridges and is associated with one-to-eight tape libraries. A scratch volume is added to a storage group on use. A private volume can be added to a storage group when it is entered in a library. You can direct allocations to a local or remote library or to a specific library by assigning the appropriate storage group in the storage group ACS routine.

If you use the DFSMSrmm storage group support for managing scratch tape pools, you must define the storage groups as tape storage groups. Refer to [z/OS DFSMSrmm Implementation and Customization Guide](#).

For more information about tape storage groups and tape libraries, refer to [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries](#).

SMS

Parent topic:[Understanding storage groups](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SMS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For SMS support of the tape library, an SCDS of the proper level must be activated. This SCDS must contain appropriate library definitions and tape storage group definitions. Information that the storage administrator must provide specifically for an Automated Tape Library Dataserver (ATLDS) or manual tape library (MTL) is listed below. See z/OS DFSMS Implementing System-Managed Storage for more information about MTL and ATLDS.

Data compaction, media type and recording technology information can be included in data classes that are used in allocation of devices in a tape library. New storage groups of type TAPE must be defined. The storage groups are involved in the allocation of new data sets to volumes within a tape library. A tape library must be defined in the SCDS. Default data class defined to the library. ACS routines must specify a storage class and the storage group selected must be associated with a tape library.

Parent topic:[Tape storage groups](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Planning storage groups for data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Ideally, you would have only one storage group containing all of your data sets, and the system would manage everything. Realistically, you have to account for a variety of data sets, such as databases, large data sets, temporary data sets, and tape data sets.

Before you actually define your storage groups, you should gather the following information:

Existing I/O hardware configuration	Projected hardware requirements
Estimated general requirements of user groups	Anticipated security requirements
Backup and recovery requirements	Data set sizes
access, based on both shared data requirements and the needs of user groups	Required system access, based on both shared data requirements and the needs of user groups
The number of systems to which the storage group is connected	The number of systems to which the storage group is connected

To begin your planning, define one primary storage group and identify all the data sets that do not fit into this general storage group category. Keep in mind your storage class and management class requirements.

For data sets that require continuous access (continuous for the storage class availability attribute) the storage group must contain sufficient volumes that do not interrupt data availability for a single device failure (for example, dual copy or array DASD).

The highest performance is generally obtainable through use of cache at the subsystem level, the device level, or both. Sufficient volumes with cache access should be available in a storage group for data sets that require high performance. Data sets that have primarily write access (write BIAS in the storage class) benefit most from a DASD fast write capability.

For data sets requiring concurrent copy, ensure that sufficient volumes in the storage group are attached through 3990 Storage Controls with Extended Platform, or through an IBM® RAMAC Virtual Array.

A *stripe* is the portion of a striped data set that resides on one volume. For striped data sets, ensure that there are a sufficient number of separate paths to DASD volumes in the storage group to allow each stripe to be accessible through a different path. The maximum number of stripes for physical sequential (PS) data sets is 59. For VSAM data sets, the maximum number of stripes is 16. Only sequential or VSAM data sets can be striped.

[Figure 1](#) shows the formula that is used to derive the optimum number of stripes that are allocated to the data set: [Figure 1. Formula for Deriving Optimum Number of Stripes](#)

Sustained data rate / Device transfer rate = Optimum number of stripes

where: Sustained data rate is the target throughput rate, specified in the storage class definitionDevice transfer rate is based on the published characteristics of the device

Ideally, the number of stripes should be on different paths, however, if this is not possible, SMS allocates to a smaller number of stripes.

If a data set has a management class that specifies automatic backup or migration, you must direct the data set to a storage group that is eligible to be processed for automatic backup or migration. You need to specify through ISMF the system or system group that is to perform the backup or migration only if that particular system or system group should do the processing. Otherwise, do not specify a particular system or system group to do the processing. It is done automatically. Management class attributes do not apply to tape data sets, so assign data sets to tape only if they do not require management class services.

Parent topic:[Defining storage groups](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Planning storage groups for OAM object collections

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For each major application that processes objects, one or more object collections must be defined to provide for the storing, cataloging, and retrieval of objects used by that application. Each object collection must be assigned to a storage group by the storage group ACS routine. Multiple collections can be assigned to the same storage group.

A storage group for OAM object collections provides a storage hierarchy containing a DB2® table for the object directory for objects in the storage group, DB2 tables for storage of objects on disk sublevel 1 (DB2 tables), disk sublevel 2 (file system), optical volumes for the storage of objects on optical media, and tape volumes for the storage of objects on tape media. Tape volumes can be defined as sublevel 1 or sublevel 2.

Each object that OAM stores is assigned a storage class and a management class. OAM uses the storage class to determine the initial placement of an object in the OAM object storage hierarchy. OAM also uses the storage class during the OSMC storage management cycle to determine the correct placement of the object when the storage management cycle processes that object. OAM uses the Initial Access Response (IARS) parameter in the storage class to determine if a primary copy of an object is stored on disk (DB2 tables or a file system) or on removable media (optical or tape). If the IARS parameter in the storage class that is assigned to the object is zero, the primary copy of the object is stored on disk. If the IARS parameter is nonzero, the primary object is stored on removable media.

If the IARS parameter specifies that removable media is to be used, the Sustained Data Rate (SDR) parameter of the storage class determines which removable media, optical or tape, is used to accept the primary copy of the object. If the SDR parameter of the storage class is *greater than or equal to* three (≥ 3), the primary copy of the object is stored on a tape volume. If the SDR parameter of the storage class is *less than* three (< 3), the primary copy of the object is stored on an optical disk volume.

If the IARS and SDR parameters, taken together, specify either disk or tape media, the OAM Sublevel (OSL) parameter of the storage class determines which disk or tape sublevel is to be used to accept the primary copy of the object. If it has been determined that the primary copy of the object is to be stored on disk, then if the OSL parameter of the storage class equals 1, the primary copy of the object is stored on disk sublevel 1 (DB2 tables), whereas if the OSL parameter of the storage class equals 2, the primary copy of the object is stored on disk sublevel 2 (file system).

Similarly, if it has been determined that the primary copy of the object is to be stored on tape, then if the OSL parameter of the storage class equals 1, the primary copy of the object is stored on a tape sublevel 1 volume. If the OSL parameter of the storage class equals 2, the primary copy of the object is stored on a tape sublevel 2 volume.

[Table 1](#) provides a summary of how the IARS, SDR, and OSL values determine the initial placement of an object in the OAM object storage hierarchy. Table 1. Summary of media selection for object storage

Before you actually define your storage groups, you should gather the following information:

Existing I/O hardware configuration	Projected hardware requirements
Estimated general requirements of user groups	Anticipated security requirements
Backup and recovery requirements	Object sizes
Access, based on both shared data requirements and the needs of user groups	Required system access, based on both shared data requirements and the needs of user groups
The number of systems to which the storage group is connected	The number of systems to which the storage group is connected

See z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support for more information on planning for objects.

Parent topic:Defining storage groups

 | 

©Copyright IBM Corporation 1990, 2014

Defining storage group attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: You must allocate an SCDS.

Perform the following steps to define storage groups.

Select option 6, Storage Group, from the ISMF Primary Option Menu for storage administrators. This displays the Storage Group Application Selection panel.

Supply values on the Storage Group Application Selection panel: **CDS Name** This is the name of an SCDS. ISMF primes the CDS Name field with the last used SCDS name. The default is 'active'. This represents the currently active configuration, but you cannot define or alter the storage groups for the active configuration. **Storage**

Group Name This is the storage group name. ISMF primes the Storage Group Name field with the name last used within ISMF. The default is an asterisk, *, which represents all storage groups in the specified control data set name. **Storage Group**

Type See [Storage group types](#) for more information. **Option** Select option 3, Define. This displays the panels for defining the storage group. These vary with the storage group type.

Complete the panels for defining the storage group. See [Storage group types](#) for more information. You can leave any of the Storage Group Define panels at any time without saving the storage group by issuing the Cancel command.

The remainder of this topic provides the panel sequences in the Storage Group Application and explains the attribute values for each type of storage group.

[Storage group types](#)

[Defining a VIO storage group](#)

[Defining a pool storage group](#)

[Assigning DASD storage groups to data sets](#)

[Defining a dummy storage group](#)

[Defining a copy pool backup storage group](#)

[Defining an object storage group](#)

[Defining an object backup storage group](#)

[Defining object or object backup storage group status](#)

[Assigning an OAM object collection to a storage group](#)

[Defining a tape storage group](#)

[Defining tape storage group status](#)

Parent topic: [Defining storage groups](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Storage group types

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The valid values for Storage Group Type are:

- VIO**Virtual I/O (VIO) storage groups are used to allocate data sets to VIO, which simulates the activity of a DASD volume. VIO storage groups do not contain any actual DASD volumes. You can put temporary data sets in VIO storage groups. See [Defining a VIO storage group](#) for more information.

PoolPool storage groups contain the volume serial numbers of system-managed DASD volumes. You can use pool storage groups for both temporary and permanent data sets. See [Defining a pool storage group](#) for more information.

DummyDummy storage groups contain the volume serial numbers of DASD volumes that no longer reside on the system but that you want to treat as SMS DASD volumes. Using dummy storage groups allows existing JCL that explicitly references the DASD volumes in VOL=SER statements to work. If end users specify a VOL=SER in their JCL, and that volume serial number is in a dummy storage group list, then SMS issues a catalog request to find the desired data set rather than using the volume serial number. Volumes in dummy storage groups cannot be used when performing volume allocations. For example, the following DD statement, where DUMMY1 is a volume in a dummy storage group, does not work:

```
/DD1 DD VOL=SER=DUMMY1,UNIT=SYSDA,DISP=SHR
```

A dummy storage group should not contain the volume serial number of a DASD volume that exists in the system. If the DASD volume exists in the system and the data set is system-managed, no JCL errors occur but the job fails during allocation. If the DASD volume exists in the system and the data set is not system-managed, then the resulting errors depend on the type of data set. For uncataloged data sets, either the data set cannot be found or the wrong data set with the same name is found. For cataloged data sets, the job fails during allocation.

See [Defining a dummy storage group](#) for more information.

Copy Pool BackupCopy pool backup storage groups contain the target volumes of fast replication backup requests. Ensure that the number of eligible target volumes in a copy pool backup storage group is sufficient to satisfy the needs of the number of backup versions that are specified in its associated copy pool. To be eligible for fast replication backup, a target volume must:

- Have the same track format as the source volume
- Be the same size as the source volume
- For FlashCopy®: Not be a primary or secondary volume in an XRC or PPRC volume pair
- Not be in a FlashCopy relationship at the time of the backup
- For SnapShot, the volume must reside in the same RVA/SVA as the source volume.

Volumes associated with copy pool backup storage groups are for DFSMSHsm use. Do not use these volumes for SMS volume allocation, or the allocation will fail.

See [Defining a copy pool backup storage group](#) for more information.

For information about using ISMF to define a copy pool, see [Defining copy pools](#).

For more information about FlashCopy and SnapShot, see z/OS DFSMS Advanced Copy Services.

ObjectObject storage groups identify an object storage hierarchy. You can define object storage groups with no optical libraries. If you want the backup copies of objects to be written to tape, specify SETOAM statements in the CBROAMxx member of PARMLIB.

See [Defining an object storage group](#) for more information.

Object BackupObject backup storage groups define the groups that are to be used to contain backup data and can specify one or more optical libraries that contain backup copies of objects. The object backup storage groups can be defined with no optical libraries and have the backup copies of objects written to tape by specifying SETOAM statements in the CBROAMxx member of PARMLIB.

See [Defining an object backup storage group](#) for more information.

TapeTape storage groups identify storage groups to maintain system-managed tape volumes. See [Defining a tape storage group](#) for more information.

Parent topic:Defining storage group attributes

◀ ▶ | ▶ ▶

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining a VIO storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define a VIO storage group.

Specify VIO for storage group type on the Storage Group Application Selection panel, as explained in [Defining storage group attributes](#). This causes the VIO Storage Group Define panel to be displayed.

Supply values on the VIO Storage Group Define panel. The SCDS Name and Storage Group Name fields are output fields that contain the SCDS and storage group names you specified in the Storage Group Application Selection panel. You specify: **Description**This is an optional field of 120 characters in which you can describe the VIO storage group.**VIO Maxsize**This is a data set maximum size for VIO. See [Values for defining a VIO storage group](#).**VIO Unit**This represents the generic DASD device type (for example, 3380 or 3390) that this storage group simulates.**System/Sys Group Name and Status**This is a storage group status for each system and system group. See [Values for defining a VIO storage group](#) for more information.

Use the END command to save the VIO storage group and return to the Storage Group Application Selection panel.

[Values for defining a VIO storage group](#)

Parent topic:[Defining storage group attributes](#)



©Copyright IBM Corporation 1990, 2014

Values for defining a VIO storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

VIO Maxsize This is a required value that represents the maximum size, in KB, of data sets to be allocated to VIO. If a data set exceeds the maximum size, then the allocation fails. However, if the storage group ACS routine assigns both a VIO storage group and a pool storage group to the list of candidate storage groups for the data set, and the data set exceeds the maximum size, then the data set is allocated to the pool storage group. Attention: z/OS® supports two space-related parameters called MXIG and ALX. Use these parameters with extreme caution for VIO as well as non-VIO allocations. You can specify them as subparameters of the JCL SPACE keyword. You can also specify them to the system as defaults in the allocation parmlib member ALLOCxx, in which case they get applied for VIO and dynamic allocation requests if space parameters are not specified. In these cases instead of allocating the requested space quantity the allocated space is based on MXIG or ALX values. See z/OS DFSMS Using Data Sets for details on the actual amount of space allocated. For VIO allocations, this can result in affecting paging space and continued operation of the system if all of the allocated space is actually used. These parameters can be overridden by the dynamic allocation installation exit. The value passed to the SMS ACS routines for the read-only variables &SIZE and &MAXSIZE are based on the user-specified space quantity and are not based on MXIG or ALX.

Parent topic:[Defining a VIO storage group](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining a pool storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define a pool storage group.

Specify POOL for storage group type on the Storage Group Application Selection panel, as explained in [Defining storage group attributes](#). This causes the Pool Storage Group Define panel to be displayed.

Supply values on the Pool Storage Group Define panel. The SCDS Name and Storage Group Name are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel. See [Values for defining a pool storage group](#) for information on the other values.

Use the END command to save the storage group and return to the Storage Group Application Selection panel. If you want to define SMS storage group status, press Enter. See [Defining pool storage group status](#) for more information.

[Values for defining a pool storage group](#)

[Defining system access to pool and VIO storage groups](#)

[Defining pool storage group status](#)

[Adding volumes and defining the SMS volume status](#)

Parent topic:[Defining storage group attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Values for defining a pool storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The storage group and management class are interrelated. The storage group Auto Migrate and Auto Backup parameters specify whether the volumes in this storage group are eligible to be processed automatically. The management class, assigned to the data sets residing on the volumes, determines whether and how to process the data sets on the volume. In contrast, if you set Auto Migrate or Auto Backup to N (No) in the storage group, the volumes in the storage group are not processed and data sets residing in the storage group are not migrated or backed up.

You can specify the following attributes on the Pool Storage Group Define panel:

DescriptionIs an optional field of 120 characters with which you can describe the pool storage group.

Auto MigrateSpecifies whether the DASD volumes in this storage group are eligible for automatic space management processing. **Auto Migrate** is a required field, which ISMF primes with the value Y (yes). **Y**(yes) specifies that data sets are eligible for primary space management and that DFSMSHsm performs automatic space management if a processing window is defined to DFSMSHsm. If the DFSMSHsm system setting SETSYS INTERVALMIGRATION is specified, the data sets are eligible for interval migration. If the DFSMSHsm system setting SETSYS ONDEMANDMIGRATION(Y) is specified, the data sets are eligible for on-demand migration and are not eligible for interval migration, even if SETSYS INTERVALMIGRATION is specified.**N**(no) specifies that data sets are not eligible for automatic migration.**I**(interval) specifies that data sets are eligible for primary space management and forces DFSMSHsm to perform automatic interval migration independent of the DFSMSHsm system setting for SETSYS INTERVALMIGRATION. Interval migration is performed on volumes that are at or above the high threshold only. Interval migration is performed hourly. If a low threshold value of 0 is specified, DFSMSHsm migrates all eligible data sets in the selected storage group. DFSMSHsm on-demand migration is not performed. This value is most useful for storage groups used with tape mount management.**P**(primary) specifies that data sets are eligible for primary space management. Interval migration and on-demand migration are not performed regardless of the DFSMSHsm system setting for SETSYS INTERVALMIGRATION and SETSYS ONDEMANDMIGRATION.**Note:** If for some reason (such as a system outage) system-managed temporary data sets are not deleted at the end of a job, DFSMSHsm attempts to delete them during primary space management. In order for DFSMSHsm to delete these data sets, you must specify **Auto Migrate** as Y, I or P.

Auto BackupSpecifies whether the DASD volumes in this storage group are eligible for automatic backup processing. Auto Backup is a required field, which ISMF primes with the value Y, yes. Y specifies that DFSMSHsm backs up the data sets on the volume according to management class requirements.**Auto Dump**In Auto Dump, you specify whether you want to automatically dump all the DASD volumes in this storage group. It is an optional field, which ISMF primes with the value N, no. If you specify a value of Y, yes, DFSMSHsm automatically dumps the volumes.

Migrate, Backup, and Dump Sys/Sys Group NameISMF no longer verifies that specified system or system group names are defined to the base configuration. You can specify either a system or a system group name in these fields but a specific system specified might not be defined in the configuration, because it might be defined as part of a system group. Therefore, if these fields are not blank, you must take special care to ensure that the values are correct. If the values are incorrect, it could result in the expected DFSMSHsm operation not occurring. DFSMSHsm has the capability of processing each storage group for automatic space management, data availability management, and automatic dump processing. A system is eligible to perform the processing when any of the following are conditions are met: The

The name specified is the name of the specific host system. The name specified is the name of the system group to which the system belongs and the system is not individually defined by its system name in the configuration.

DFSMShsm ignores storage groups for which you specify a different system name, and does not process DASD volumes that have already been processed. Do not specify a Sys/Sys Group Name unless processing of the storage group for the function must be performed only on that one host because that limits the capabilities of DFSMShsm to perform the request.

The same rules apply to Backup Sys/Sys Group Name for data availability management processing, and to Dump Sys/Sys Group Name for automatic dump processing. All three Sys/Sys Group Name fields are optional and primed with blanks.

OverflowSpecifies whether this pool storage group is designated as an overflow storage group to handle periods of high demand for initial primary space allocations. Overflow is a required field, which ISMF primes with the value N (No). Y (Yes) specifies that the storage group is an overflow storage group. To make overflow storage groups eligible for allocation, you must assign designated overflow storage groups in your storage group ACS routine.

If all volumes in a non-overflow storage group are so full that the current allocation request will push them over high threshold, while volumes in the overflow storage group are not so full, then the new data set will be allocated on a volume in the overflow storage group. The assumption is that all other attributes of the non-overflow storage group and overflow storage group and the volumes in those storage groups are the same.

During initial allocation, volumes in an overflow storage group are less preferred than volumes in an enabled storage group but more preferred than volumes in a quiesced storage group. Therefore, overflow volumes are not placed on the primary volume list but can be placed on the secondary volume list. When an overflow storage group contains more volumes than a buffer storage group, specified volume counts might result in overflow volumes being preferred over volumes in the buffer storage group.

An overflow storage group may also be specified as an extend storage group. Volumes residing in overflow storage groups are preferred over quiesced volumes and storage groups. If you quiesce an overflow storage group or volume then the quiesced volumes are preferred over quiesced overflow volumes.

Extend SG NameSpecifies the name of another pool storage group to which data sets from the primary storage group can be extended when there is an insufficient amount of storage on the primary storage group. A primary storage group is the storage group in which the initial allocation resides. Extend SG Name is an optional field of one-to-eight alphanumeric or national characters, or a combination, the first of which must be alphabetic. ISMF primes the field with the default value "blank."

You can define only one extend storage group to each storage group. However, you can define the same extend storage group to more than one primary storage group. Also, you can define two storage groups as extend storage groups of each other.

An extend storage group may also be specified as an overflow storage group. Extend storage groups will not be used for initial allocation unless they are specified in the ACS routines. All storage groups that are listed in the ACS routines are candidates for initial allocation. Extend storage group attributes are not referenced during initial allocation.

Example: You can define storage group (SG) 3 as an extend of SG 2, and define SG 2 as an extend of SG 1. If SMS selects SG 1 as the primary storage group, the data set can extend only to SG 2. If SMS selects SG 2 as the primary storage group, the data set can extend only to SG 3.

For more information about data set extend in the volume selection process, see [SMS volume selection for data set allocation](#).

Note:**Requirement:** When you specify an extend storage group, you must ensure that connectivity across the SMSplex is the same for both the primary storage group and the extend storage group. Because an extend storage group is a pool storage group, the SMS status also applies to the extend storage group.

Copy Pool Backup SG NameSpecifies the name of the copy pool backup storage group that contains eligible volumes for fast replication backup versions. You can specify the name of a copy pool backup storage group that is shared by more than one pool storage group.

Copy Pool Backup SG Name is an optional field of one-to-eight alphanumeric or national characters, or a combination, the first of which must be alphabetic. ISMF primes the field with the default value of a blank name.

Dump ClassSpecifies the name of a dump class that is defined in DFSMSHsm. There are five Dump Class fields, so up to five unique dump class names may be specified. When DFSMSHsm dumps DASD volumes that belong to the storage group, it directs their contents to the dump classes. To use dump classes, you must first define their names and parameters using DFSMSHsm. Then you can identify the dump class names to SMS using this panel.

Dump Class is an optional field. ISMF primes the field with the default value "blank."

DEFINE SMS Storage Group StatusSpecifies whether you want to change the status of the pool storage group with respect to a given system in the SMS complex. Initially, the status is set to ENABLE for all. The values are YChange a status. When you complete the current panel, you will see the SMS Storage Group Status Define panel. For more information, see [Defining pool storage group status](#). NLeave all the status fields as ENABLE. This is the default.

Migration and Allocation ThresholdsSpecifies an upper and lower space limit for the DASD volumes in a pool storage group. SMS tries to stay within these thresholds by looking at the primary space allocation of each data set before assigning it to a given DASD volume. For example, the SMS volume selection function attempts to prevent allocation of a data set to a given DASD volume if that allocation causes the volume's high threshold to be exceeded. When a volume reaches or exceeds the high threshold, SMS issues an ENF72 signal, regardless of the Auto Migrate setting.

In addition, this high threshold value is used by DFSMSHsm to determine whether data sets should be migrated off a DASD volume in the storage group. The low threshold value is used as the threshold goal in reducing the amount of space occupied on a DASD volume in the storage group during interval migration or daily space management. The low threshold value must be less than or equal to the high threshold value.

Both numbers are percentages of the total space on the DASD volume. If you specify Y for Auto Migrate, then you must specify both a high and low threshold. If you specify Y, the low threshold limit is 1. If you specify I for automatic interval migration, you can specify a low threshold value of 0 to migrate all the data sets in the selected storage group. The hourly migration trigger for storage groups with a value of AM=I is the occupancy at or above the midpoint of the high and low thresholds. Because storage groups used for tape mount management tend to fill up several times a day, allowing interval migration for these storage groups allows DFSMSHsm to better keep up with the demand.

ISMF requires that you enter a high threshold value when you specify N for Auto

Migrate, when defining a pool storage group. Because SMS needs the value for allocation purposes, the High Threshold field is a required field with a primed value of 85.

For a track-managed space of an extended address volume, the allocation and migration threshold specifies the threshold percentage of space allocation that triggers or stops migration of data sets from volumes in this storage group during interval migration. Valid values for high threshold are 1-100. Valid values for low threshold are 0-99. These fields are primed with default values and are ignored by SMS and DFSMSHsm for other than extended address volumes. For the default values, refer to the ISMF help. If you alter an existing storage group to add extended address volume values, the specified bits are set to ON, when you specify the BreakPointValue and the Track Allocation Thresholds. If the specified indicators are not ON, SMS uses the existing threshold values, which represent threshold values for the entire volume, to represent threshold values for the track-managed space.

For more information about specifying allocation and migration thresholds, see z/OS DFSMSHsm Storage Administration.

For more information about selecting appropriate threshold levels, see z/OS DFSMS Implementing System-Managed Storage.

Guaranteed Backup FrequencySpecifies the number of days within the last backup period in which the backup process should have a copy of each of the data sets within the applicable storage group. If Auto Backup is specified as Y, this attribute is required; otherwise it is optional. You specify the maximum number of days that can elapse between backups. You can specify from 1 to 9999 days or you can specify NOLIMIT. If you specify NOLIMIT, then data sets in the storage group are backed up according to management class specifications. There is no default.

BreakPointValueSpecifies the number of cylinders (0-65520) that SMS uses to select volumes for VSAM data sets. You can specify BreakPointValue for each Storage Group separately. This keyword overrides the value in the IGDSMSxx PARMLIB member. If you do not specify this keyword, SMS assigns the default value of 10 cylinders. When SMS is not active, the system assigns the default value.

Processing PrioritySpecifies the processing priority of the storage group. The value is in the range from 1 (lowest priority) to 100 (highest priority). It is used during DFSMSHsm space management. The default is 50.

Parent topic:[Defining a pool storage group](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining system access to pool and VIO storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you want a system to access a pool storage group, you must ensure that devices on which the groups reside are physically connected to the system.

No devices are associated with a VIO storage group. However, the SMS status associated with the VIO storage group determines if this storage group can be used by each system in the complex. The SMS statuses, described below, show the relationships between each system in the SMS complex and a given VIO or pool storage group. **DISALL**Disable All prevents the system from allocating or accessing data sets in the storage group. **DISNEW**Disable New prevents the system from allocating new data sets (and DISP=MOD data sets that do not currently exist) in the storage group. DISNEW prevents data sets from extending to a volume whose volume or storage group status is DISNEW. **ENABLE**The system can allocate and access data sets in the storage group. ENABLE is the default relationship between a system and a storage group. **NOTCON**Not Connected indicates that the storage group is defined but not accessible to the system. It resembles DISALL, except you cannot dynamically change the NOTCON system status for Storage Groups. You can, however, change it for volumes. Use the VARY command to change the status of a volume from NOTCON to another status. If you want a storage group to have access to NOTCON volumes, physically connect them to the system, define the connection by changing the system status in the ISMF Storage Group Application, and activate the configuration that defines the connection. If you use the VARY command to change the status of a volume the change is made in the ACDS and not the SCDS, so the status will change to whatever is defined in the SCDS if an SCDS is subsequently activated.

QUIALLQuiesce All prevents the system from scheduling jobs that allocate or access data sets in the storage group. This state only affects JES3 systems, that support job scheduling. When the job is executing, QUIALL has the same effect in a JES2 or JES3 environment. In an JES2 environment, the volumes become a secondary volume selection candidate through QUIALL. *Secondary volume selection candidacy* means that these volumes are still available for allocation, but less preferred than other volumes.

For more information about primary and secondary volume selection candidacy, see [Conventional volume selection](#).

QUINEWQuiesce New prevents the system from scheduling jobs that allocate new data sets (and DISP=MOD data sets that do not currently exist) in the storage group. If the system uses JES3 to schedule jobs, you can use QUINEW only as long as other volumes are available. Before scheduling a job, JES3 verifies that all resources are available, unlike JES2, which schedules a job even if all the needed resources are not available. This can lead to a contention for resources, because a job running under JES2 can hold some resources while it waits for others to become available. Under JES3, if any of the candidate volumes are available, JES3 schedules the job and SMS selects the volume. Under both JES2 and JES3, SMS selects QUINEW volumes only as a last resort.

For more information about primary and secondary volume selection candidacy, see [Conventional volume selection](#).

Note: Because an extend storage group is a pool storage group, the SMS status also applies to the extend storage group.

Parent topic:[Defining a pool storage group](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining pool storage group status

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: You must specify Y for DEFINE SMS Storage Group Status on the Pool Storage Group Define panel, as described in [Defining a pool storage group](#).

Perform the following steps to define pool storage group status.

Supply values on the SMS Storage Group Status panel, **System/Sys Group**

NameLists the system or system group names you defined in the base configuration.**SMS SG Status**Lists the relationship between the storage group and each system in the SMS complex. You can specify the statuses, which are explained in [Defining system access to pool and VIO storage groups](#).

Use the END command to save your values and return to the Pool Storage Group Define panel.

Use the END command again (on the Pool Storage Group Define panel) to save the pool storage group. This returns you to the Storage Group Application Selection panel. After establishing the relationships between your storage group and the systems in the SMS complex, you need to define DASD volumes to the storage group. For more information, see [Adding volumes and defining the SMS volume status](#).

Parent topic:[Defining a pool storage group](#)



©Copyright IBM Corporation 1990, 2014

Adding volumes and defining the SMS volume status

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: You must have displayed the Storage Group Application Selection panel (select option 6, Storage Group, from the ISMF Primary Option Menu).

Perform the following steps to add volumes and define the SMS volume status. On the Storage Group Application Selection panel, select option 5, Volume. This displays the Storage Group Volume Selection panel.

Supply values on the Storage Group Volume Selection panel: Select option 2, Define. Specify the volume serial numbers of the DASD volumes that you want to add to the pool storage group. Each time you press Enter, you see the SMS Volume Status Define panel. For an example, see [Figure 1](#).

Supply values on the SMS Volume Status Define panel. For more information on supplying values, see [Values on the SMS volume status define panel](#).

Issue the END command to return to the Storage Group Application Selection panel. You can define additional DASD volumes to the pool storage group. [Figure 1. Defining Volume System Status](#)

SMS VOLUME STATUS DEFINE					Panel	Utilities	Scroll	Help
					Page 1 of 2			
Command ==>								
SCDS Name : SMS.SCDS1.SCDS								
Storage Group Name . . : POOL1								
Volume Serial Numbers : 010000 - 010010								
To DEFINE SMS Volume Status, Specify:								
System/Sys Group Name	SMS Vol Status	System/Sys Group Name	SMS Vol Status	(Possible SMS Vol Status for each: NOTCON, ENABLE, DISALL, DISNEW, QUIALL, QUINEW)				
----- SYSTEM1	==> ENABLE	----- SYSTEM2	==> ENABLE	-----				
----- SYSTEM3	==> ENABLE	----- SYSTEM4	==> ENABLE	-----				
----- SYSTEM5	==> ENABLE	----- SYSTEM6	==> ENABLE	-----				
*SYSPLEX1	==> ENABLE		==>	* SYS GROUP = sysplex minus systems in the sysplex explicitly defined in the SCDS				
	==>		==>					
	==>		==>					
	==>		==>					
	==>		==>					
Use ENTER to Perform Verification; Use DOWN Command to View next Panel; Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.								

[Values on the SMS volume status define panel](#)

[Parent topic:Defining a pool storage group](#)



Values on the SMS volume status define panel

z/OS DFSMSdfp Storage Administration
SC23-6860-01

On the SMS Volume Status Define panel, you define the relationship between the DASD volume in the pool storage group and each system in the SMS complex.

Volume Serial Numbers Lists the volumes defined to the storage

group. **System/Sys Group name** Lists the systems you defined in the base

configuration. **SMS Volume Status** Lists the relationship between a DASD volume and each system in the SMS complex. DASD volumes have both a physical (actual) and a logical (system-defined) connection to systems in the SMS complex. You are responsible for maintaining consistency between these two types of connections.

For a system to have access to a data set, you need to define one of the types of access from the system to a storage group, and you need to define one of the five types of access from the storage group to a DASD volume (ENABLE, DISALL, DISNEW, QUIALL, and QUINEW for both types of access). You must ensure that a physical connection exists from the DASD volume to the system, and the MVS status of the DASD volume must be ONLINE.

See [Defining system access to pool and VIO storage groups](#) for an explanation of the SMS statuses. The volume statuses show the relationships between each system in the SMS complex and a given DASD volume in the pool storage group. When a system attempts to allocate a data set, it proceeds in order through the following checks with respect to the current system:

Storage group status-ENABLE/DISALL/DISNEW/QUIALL/QUINEW/NOTCONS
SMS volume status-ENABLE/DISALL/DISNEW/QUIALL/QUINEW/NOTCONM
VMS volume status-ONLINE/OFFLINE

Parent topic: [Adding volumes and defining the SMS volume status](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Assigning DASD storage groups to data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Storage groups can only be assigned through the storage group ACS routine. For a given data set, the storage group ACS routine runs only if the storage class ACS routine assigns a valid storage class. If the storage class is not valid, allocation fails. If no storage class is assigned, then the data set is not system-managed and is allocated according to the rules in a non-system-managed environment.

You can assign candidate storage groups to a data set allocation from which SMS selects eligible DASD volumes for the data sets. If the storage group ACS routine does not determine a storage group for a data set, allocation fails.

Restriction: SMS does not check or verify DASDVOL authorization for data set allocations on SMS-managed volumes. By adding a volume to SMS, you are removing it from any active DASDVOL authorizations because DASDVOL access is not verified on SMS-managed volumes. See *z/OS Security Server RACF Security Administrator's Guide* for further information on DASDVOL authority.

Parent topic:[Defining storage group attributes](#)

 | 

©Copyright IBM Corporation 1990, 2014

Defining a dummy storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define a dummy storage group.

Specify DUMMY for storage group type on the Storage Group Application Selection panel, as explained in [Defining storage group attributes](#). This causes the Dummy Storage Group Define panel to be displayed.

Supply a description on the Dummy Storage Group Define panel, if desired. The description is optional. It can be up to 120 characters. The SCDS Name and Storage Group Name are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel.

Use the END command to save the newly defined dummy storage group and return to the Storage Group Application Selection panel.

Select option 5, Volume, on the Storage Group Application Selection panel to display the Storage Group Volume Selection panel, where you can add volume serial numbers to the storage group. (Initially, the storage group contains no volume serial numbers.) ISMF primes the other fields on the Storage Group Application Selection panel with the most recently specified values. You do not need to specify a Storage Group Type.

Supply values on the Storage Group Volume Selection panel to define volume serial numbers to the dummy storage group. For more information, see [Defining DASD volumes to a dummy storage group](#).

[Defining DASD volumes to a dummy storage group](#)

Parent topic:[Defining storage group attributes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining DASD volumes to a dummy storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To add DASD volumes to the dummy storage group, supply values on the Storage Group Volume Selection panel:

Select option 2, Define. Specify the volume serial numbers in the **Specify a Single Volume (in Prefix) or Range of Volumes** field. You can specify a single volume serial number by typing the number under Prefix. The value can be from one to six characters, but it must be a fully specified volume serial number. You can specify a range of volume serial numbers by typing the prefix or suffix that is common to a set of volumes under Prefix or Suffix, the low individual volume number under From, and the high individual volume number under To.

You can specify an X under Type to include hexadecimal values in your range or an A under Type to include alphabetic values in your range. If you leave the Type field blank, only decimal values will be used in your range. Note: The number of characters in the FROM and TO fields must be the same. If TYPE is A, only one character is allowed in the FROM and TO fields. The value of the FROM field must be less than or equal to the value of the TO field.

For example, in [Figure 1](#), volume numbers SYS001 through SYS077 are specified on the first volume specification line (only decimal numbers will be used in the range). Volume number DFPIP1 is specified on the second line. Volume numbers SYS25A through SYS30A, including SYS2AA through SYS2FA, are specified on the third line. Volume numbers NVOL through PVOL are specified on the fourth line, which specifies alphabetic values. You can add as many as a hundred volume serial numbers at a time to a pool storage group this way.

[Figure 1. Defining Dummy Storage Group Volume Serial Numbers](#)

```
Panel Utilities
Help
-----
STORAGE GROUP VOLUME SELECTION
Command ===>
CDS Name . . . . . : SMS .SCDS1 .SCDS
Storage Group Name : DUMMY1
Storage Group Type : DUMMY

Select One of the following Options:
 2 1. Display - Display SMS Volume Statuses (Pool & Copy Pool Backup only)
    2. Define - Add Volumes to Volume Serial Number List
    3. Alter - Alter Volume Statuses (Pool & Copy Pool Backup only)
    4. Delete - Delete Volumes from Volume Serial Number List

Specify a Single Volume (in Prefix), or Range of Volumes:
  Prefix   From   To   Suffix   Type
      ____  ____  ____  ____  ____
  ===> SYS    001   077
  ===> DFPIP1
  ===> SYS    25     30     A       X
  ===> N        P     VOL
Use ENTER to Perform Selection;
Use HELP Command for Help; Use END Command to Exit.
```

After you specify volume serial numbers, press Enter to add the DASD volumes to the dummy storage group. Define any additional volume serial numbers, then use END to return to the Storage Group Application Selection panel. If any volumes cannot be added, a list of the volumes that could not be added is displayed.

When adding a new volume serial number (volser), you must make sure that all volsters defined in the volume list are unique. ISMF verification only verifies that all volsters are unique within the same dummy or pool storage group. No attempts are made to verify the device types of each volume. Therefore, if a library volser is duplicated in a dummy or pool storage group, this error is not detected until a request is issued to mount this volser.

Parent topic:[Defining a dummy storage group](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining a copy pool backup storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define a copy pool backup storage group. Specify COPY POOL BACKUP for storage group type on the Storage Group Application Selection panel, as explained in [Defining storage group attributes](#). This causes the Copy Pool Backup Storage Group Define panel to be displayed.

Supply a description on the Copy Pool Backup Storage Group Define panel, if desired. The description can be up to 120 characters. The SCDS Name and Storage Group Name are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel. You can indicate the name for the copy pool backup storage group on the associated source pool storage group. For more information, see [Defining a pool storage group](#).

Parent topic:[Defining storage group attributes](#)



©Copyright IBM Corporation 1990, 2014

Defining an object storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Before you define an object storage group that uses optical volumes, you should define your optical libraries. See z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support for more information.

Perform the following steps to define an object storage group.

Specify OBJECT for storage group type on the Storage Group Application Selection panel, as explained in [Defining storage group attributes](#). This causes the Object Storage Group Define panel to be displayed.

Supply values on the Object Storage Group Define panel. The SCDS Name and Storage Group Name are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel. For information on the other fields, see [Attributes of the object storage group](#). Use the DOWN command to advance to the next page as needed.

Use END to save the definition, or press Enter if defining values for status. See [Defining object or object backup storage group status](#) for more information.

[Attributes of the object storage group](#)

Parent topic: [Defining storage group attributes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Attributes of the object storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To define an object storage group you specify the following attributes:

DescriptionThis is an optional description of the object storage group. It can be up to 120 characters. **Qualifier**This identifies a DB2® database on DASD that contains table spaces, one of which contains the object directory for this object storage hierarchy. The qualifier attribute also points to the DB2 object storage tables on DASD that have been defined for the object storage hierarchy. You can use one-to-eight alphanumeric characters as qualifiers. This qualifier must be defined as a package in the CBRPBIND job to create the necessary DB2 package for the application plans.

Cycle Start TimeThis identifies the beginning of a window of time. Use this field in association with Cycle End Time to specify a specific window of time in which object processing can be started for this storage group. Specify a value from 0 to 23, or NONE. A value from 0 to 23 represents an hour of the day. For example, specify 00 for midnight; 01 for 1 a.m.; or 23 for 11 p.m.

Specify NONE if you do not want automatic processing for the storage group.

The value for Cycle Start Time must be different from the value for Cycle End Time. When you specify NONE, the value for Cycle End Time must be blank.

Recommendation: If you are running in an OAMplex and specify a cycle start time, specify an OSMC processing system name. Otherwise, object processing for that object storage group will start on all systems in the OAMplex at cycle start time.

Cycle End TimeIdentifies the end of a window of time when object processing can be started for this storage group. The function of the cycle end time depends on which cycle window mode is in effect when OAM initializes. The cycle window mode can be specified by using the CYCLEWINDOW keyword for the SETOSMC statement in the CBROAMxx parmlib member (refer to z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support). In Start Only mode, the cycle end time identifies the end of a window of time when object processing can be started for this storage group. In Start Stop mode, when the cycle end time has been reached, processing is the same as if a STOP command had been issued for that storage group. No new work will be scheduled, all work in progress is allowed to complete.

Specify a value from 0 to 23, or blank. A value from 0 to 23 represents an hour of the day, and is required when a value from 0 to 23 was specified for Cycle Start Time.

For example, specify an hour of the day as 00 for midnight; 01 for 1 a.m.; or 23 for 11 p.m.

The value for Cycle End Time must be different from the value for Cycle Start Time. Leave the field blank if you specified NONE for Cycle Start Time.

OSMC Processing SystemSpecifies an OSMC processing system name. This is the one that identifies which OAM system in the sysplex performs OSMC processing for this object storage group. This prevents multiple systems from trying to process the same storage group at the same time, especially when the Cycle Start Time is used for automatically starting storage group processing.

Library NamesYou can specify either one-to-eight real library names that represent optical libraries, or one-to-eight pseudo library names. The libraries you specify are used to process write requests to the storage group. Names must be valid real or pseudo optical library names that are defined in the SCDS. **Volume Full Threshold**This is the number of free KB that triggers volume full processing for an optical volume within the object storage group. When the number of free KB falls below the threshold, the object access method marks the optical volume as being full. If the threshold is reached while OAM is writing an object, it continues writing that object until finished. OAM then writes no more objects to that volume. Valid values are from 0 to 9999. **Drive Start Threshold**This is the maximum number of object write requests that are outstanding for an optical drive in this storage group. When the

number of object write requests to this storage group, divided by the number of optical drives currently processing write requests for this storage group, exceeds this threshold, the object access method attempts to start an additional optical drive. Valid values are from 0 to 9999. This field is optional. If no value is specified, the default is 17.

Volume Full at Write Error This field indicates when to mark as "full" optical or tape volumes within this object storage group. If Y, the object access method marks an optical or tape volume full the first time an attempt to write an object on the volume fails because not enough space remains on the volume. If N, OAM marks a volume full only when the number of available KB in the user data area falls below the Volume Full Threshold for optical or the TAPEFULL THRESHOLD for tape. An optical volume is also marked full if the optical volume table of contents area is full, regardless of how many KB are remaining in the user data area. **DEFINE SMS**

Storage Group Status Indicates whether you want to change the storage group status. The values are: **Y**Change the status. When you complete the current panel, you will see the SMS Storage Group Status Define panel. For more information, see [Defining object or object backup storage group status](#). If you are defining a storage group, Y is the default value. **N**Do not change the status. If you are altering a storage group, N is the default value. You must define object storage group system statuses on the SMS Storage Group Status Define panel before the object storage group definition can be saved. After you define the status of the systems, issue the END command to save the defined object backup storage group and return to the Storage Group Application Selection panel.

OAM Deletion Protection This field specifies the deletion-protection mode for all objects in this object storage group. When deletion-protection is enabled, objects in this object storage group cannot be deleted prior to their expiration date. Deletion-protection does not restrict any changes to an object's expiration date. You must have also specified DP=P in the IEFSSNxx PARMLIB member. This field is ignored when no value is specified for DP in PARMLIB member IEFSSNxx, and when DP=A or DP=N is specified.

OAM Retention Protection This field indicates whether new objects stored into an object storage group are flagged as retention-protected for the entire life of the objects. A retention-protected object cannot be deleted prior to its expiration date, and its expiration date can never move to an earlier date. OAM Retention Protection is used by OAM to set the retention-protection mode for an object at the time that the object is stored. If OAM Retention Protection is enabled at the time that the object is stored, then the object is in retention-protection mode for the life of the object. Setting OAM Retention Protection to disabled affects the retention-protection mode of subsequent objects stored into this object storage group, but does not impact the retention-protection mode of objects that were previously stored

Parent topic:[Defining an object storage group](#)



Defining an object backup storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you define object backup storage groups, you are naming the groups that are used to contain backup data and optionally specifying the names of optical libraries that can contain backup copies of objects. There can be one or two object backup storage groups per object storage group.

Perform the following steps to define an object backup storage group.

Specify OBJECT BACKUP for storage group type on the Storage Group Application Selection panel, as explained in [Defining storage group attributes](#). This causes the Object Backup Storage Group Define panel to be displayed.

Supply values on the Object Backup Storage Group Define panel. The SCDS Name and Storage Group Name are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel. For information on the other fields, see [Attributes of the object backup storage group](#). Press Enter to verify the values and display the SMS Storage Group Status Define panel.

Supply values on the SMS Storage Group Status Define panel. See [Defining object or object backup storage group status](#) for more information. You must define object backup storage group system statuses on the SMS Storage Group Status Define panel before the object backup storage group definition can be saved. Use the END command to save the newly defined object storage group and return to the Storage Group Application Selection panel.

[Attributes of the object backup storage group](#)

Parent topic:[Defining storage group attributes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Attributes of the object backup storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To define an object backup storage group you must specify the following attributes:

DescriptionThis is an optional value of up to 120 characters that describes the object backup storage group.**Cycle Start Time**This identifies the beginning of a window of time. Use this field in association with Cycle End Time to specify a specific window of time in which object processing can be started for this object backup storage group.

Specify a value from 0 - 23, or NONE. A value from 0 - 23 represents an hour of the day. For example, specify 00 for midnight; 01 for 1 a.m.; or 23 for 11 p.m.

Specify NONE if you do not want automatic processing for the storage group. The value for Cycle Start Time must be different from the value for Cycle End Time. When you specify NONE, the value for Cycle End Time must be blank.

Recommendation: If you are running in an OAMplex and specify a cycle start time, specify an OSMC processing system name. Otherwise, object processing for that object backup storage group will start on all systems in the OAMplex at cycle start time.

Cycle End TimeIdentifies the end of a window of time when object processing can be started for this object backup storage group. The function of the cycle end time depends on which cycle window mode is in effect when OAM initializes. The cycle window mode can be specified by using the CYCLEWINDOW keyword for the SETOSMC statement in the CBROAMxx parmlib member (refer to z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support). In Start Only mode, the cycle end time identifies the end of a window of time when object processing can be started for this storage group. In Start Stop mode, when the cycle end time has been reached, processing will be the same as if a STOP command had been issued for that storage group. No new work will be scheduled, all work in progress will be allowed to complete.

Specify a value from 0 - 23, or blank. A value from 0 - 23 represents an hour of the day, and is required when a value from 0 to 23 was specified for Cycle Start Time. For example, specify an hour of the day as 00 for midnight; 01 for 1 a.m.; or 23 for 11 p.m.

The value for Cycle End Time must be different from the value for Cycle Start Time. Leave the field blank if you specified NONE for Cycle Start Time.

OSMC Processing SystemSpecifies an OSMC processing system name, which identifies which OAM system in the sysplex performs OSMC processing for this object backup storage group. This prevents multiple systems from trying to process the same storage group at the same time, especially when the Cycle Start Time parameter is used for automatically starting storage group processing.**Library**

NamesSpecifies up to eight real library names that represent optical libraries, or one-to-eight pseudo library names. The libraries you specify are used to process write requests to the storage group. Names must be valid real or pseudo optical library names defined in the SCDS.

Volume Full ThresholdSpecifies the number of free KB triggering volume full processing for an optical volume within the object backup storage group. When the number of free KB falls below the threshold, the object access method marks the optical volume full. If the threshold is reached while OAM is writing an object, OAM continues writing that object until finished. OAM then writes no more objects to that volume. Valid values are from 0 to 9999.

This is a required field if a library name is specified.**Drive Start Threshold**Specifies the maximum number of object write requests outstanding for an optical drive in this storage group. When the number of object write requests to this storage group divided by the number of optical drives currently processing requests for this storage group exceeds this threshold, the object access method attempts to start an additional optical drive. Valid values are from 0 to 9999. This is a required field if you specify a library name.

Volume Full at Write ErrorIndicates when to mark as full optical volumes within this backup object storage group. If Y, the OAM marks an

optical volume full the first time an attempt to write an object on the optical volume fails because not enough space remains on the optical volume. If you specify N, the object access method marks an optical full only when the number of available KB in the user data area falls below the Volume Full Threshold. The volume is also marked full if the optical volume table of contents area is full, regardless of how many KB are remaining in the user data area. This is a required field if you specify a library name. **DEFINE SMS Storage Group Status** Indicates whether you want to change the storage group status. This is a required field. The values are: **Y(Yes)**. If you are defining a storage group, the default value is Y and cannot be changed to N. If you specify Y, you will see a panel for changing the storage group status next. See [Defining object or object backup storage group status](#) for more information. **N(No)**. If you are altering a storage group, the default value is N and can be changed to Y or N.

Object tape parameters for object and object backup storage groups are specified with SETOAM statements in the CBROAMxx member of PARMLIB. For more information, see z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

Parent topic: [Defining an object backup storage group](#)



©Copyright IBM Corporation 1990, 2014

Defining object or object backup storage group status

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: You must define an object storage group or object backup storage group, as described in [Defining an object storage group](#) and [Defining an object backup storage group](#). After you have supplied all required values on the Object Storage Group Define panel or the Object Backup Storage Group Define panel, the SMS Storage Group Status Define panel is displayed. Only systems and not system groups are shown on this panel, because you cannot connect an object or object backup storage group to a system group.

Perform the following steps to define object or object backup storage group status. Supply values on the SMS Storage Group Status Define panel. The SCDS Name, Storage Group Name and Storage Group Type fields are output fields that contain the information you specified on the Storage Group Application Selection panel. The System Name field is also an output field. It lists the systems that you defined in the base configuration.

The SMS Storage Group Status field lists the relationship between the object or object backup storage group and each system in the SMS complex. For more information, see [Attributes for object or object backup storage group status](#).

Use the END command to save your values and return to the Object Storage Group or Object Backup Storage Group Define panel.

Use the END command again, on the Object Storage Group or Object Backup Storage Group Define panel, to complete and save the definition of the storage group.

[Attributes for object or object backup storage group status](#)

Parent topic: [Defining storage group attributes](#)



Attributes for object or object backup storage group status

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The devices associated with an object or object backup storage group have both physical (actual) and logical (system defined) connections to systems in the SMS complex. You are responsible for maintaining consistency between these two types of connections.

One of the following four relationships exists between each system in the SMS complex and the object or object backup storage group: **ENABLE**For object storage groups, SMS permits application access to the object storage hierarchy of this group. For object backup storage groups, SMS permits access to the volume set. All OSREQ functions are allowed.**DISALL**For object or object backup storage groups, SMS permits restricted access to the object storage hierarchy. The OSREQ functions STORE, RETRIEVE, and DELETE are denied to applications. All object processing continues to be done for the storage group.**DISNEW**For object or object backup storage groups, SMS permits restricted access to the hierarchy. The OSREQ function STORE is denied to applications. All object processing continues to be done for the storage group.**NOTCON**The named system cannot process this object/object backup group. NOTCON is the default.

You can enable object and object backup storage groups to more than one system. If you are not in an OAMplex, OAM ignores object storage groups that are defined as being connected to more than one system and issues a message.

Parent topic:[Defining object or object backup storage group status](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Assigning an OAM object collection to a storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

OAM object collections are always managed by SMS. The ACS routines are executed when the OAM object collection catalog entry is initially defined. The ACS routines provide storage class and management class names that are used as the default storage class and management class assignments recorded in the catalog entry, and the storage group assignment for the collection. OAM uses the storage group assignment to identify the DB2® tables used for storage of the object directory information, for storage of objects in DB2 tables, and to select the file system, optical volume, or tape volume used to store objects that belong to the collection.

Parent topic:[Defining storage group attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining a tape storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define a tape storage group.

Specify TAPE for storage group type on the Storage Group Application Selection panel, as explained in [Defining storage group attributes](#). This causes the Tape Storage Group Define panel to be displayed.

Supply values on the Tape Storage Group Define panel. The SCDS Name and Storage Group Name are output fields containing the SCDS and storage group names that you specified in the Storage Group Application Selection panel. For information on other fields, see [Attributes for tape storage groups](#).

If the value for DEFINE SMS Storage Group Status is Y, press Enter to view the SMS Storage Group Status Define panel. See [Defining tape storage group status](#) for more information.

[Attributes for tape storage groups](#)

Parent topic:[Defining storage group attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Attributes for tape storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DescriptionThis is an optional value of up to 120 characters that describes the tape storage group. **Library Names**Specifies the tape libraries that own the volumes within this storage group. One to eight library names can be associated with a tape storage group. At least one library name must be specified when defining a tape storage group. The library name in the tape storage group definition must also be defined in the same SCDS.**DEFINE SMS Storage Group Status**This is used to specify whether or not you want to modify the status (ENABLE, DISALL, DISNEW, NOTCON) of this storage group for each one of the systems in the SMS complex. The values are: **Y**If you specify Y (Yes), the SMS Storage Group Status Define panel is displayed when you press Enter. See [Defining tape storage group status](#) for more information.**N**If you specify N (No), the storage group defaults to enable for each one of the systems in the complex. N is the default.

Parent topic:[Defining a tape storage group](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining tape storage group status

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: If you specified Y for the Define SMS Storage Group Status attribute on the Tape Storage Group Define panel, as described in [Defining a tape storage group](#), you see the SMS Storage Group Status Define panel, where you can define the status of the storage group to each system in the SMS complex. Perform the following steps to define tape storage group status.

Supply values on the SMS Storage Group Status Define panel. The SCDS Name, Storage Group Name and Storage Group Type fields are output fields that contain the information you specified on the Storage Group Application Selection panel. The System/Sys Group Name field is also an output field. It lists the systems that you defined in the base configuration, using the CDS application. **SMS SG**

StatusThis field lists the relationship between the storage group and each system in the SMS complex. See [Defining system access to pool and VIO storage groups](#) for an explanation of these relationships.

Use the END command to save your values and return to the Tape Storage Group Define panel.

Use the END command again, on the Tape Storage Group Define panel, to complete and save the definition of the storage group. This returns you to the Storage Group Application Selection panel, which is primed with your CDS and storage group name.

Parent topic:[Defining storage group attributes](#)



©Copyright IBM Corporation 1990, 2014

Defining additional storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can copy existing storage groups and modify them to create new storage groups by using the COPY line operator, which is explained in [Copying SMS components](#).

When you copy either a pool or VIO storage group within an SCDS, all of the system status values and system names for automatic processing are copied. When you copy either a pool or VIO storage group from one SCDS to a different SCDS, all of the system status values default to ENABLE rather than to the system status values of the source storage group. Also, the Migrate System Name, Backup System Name, and Dump System Name fields are set to blank.

Whenever you copy an object storage group within an SCDS, you must modify the qualifier to ensure that the new object storage group has a unique qualifier for the connected system. All qualifiers within a system must be unique. Specify Y for the Perform Alter attribute on the Copy Entry panel to display the Object Storage Group Alter panel. Modify the qualifier and issue an END command to save the new qualifier.

Whenever you copy an object storage group from one SCDS into a different SCDS, the qualifier must be unique within the target SCDS. If necessary, alter the qualifier in the same manner as required for copying an object storage group within an SCDS.

The system status values of object and object backup storage groups are retained when copying within an SCDS. Object and object backup storage groups can be enabled by and connected to more than one system. Specify Y for the Perform Alter attribute on the Copy Entry panel, and Y for the Alter SMS Storage Group Status attribute on either the Object Storage Group Alter panel or Object Backup Storage Group Alter panel to display the system statuses for modification.

Parent topic:[Defining storage groups](#)



Listing volumes in a storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to list volumes for a pool, dummy, copy pool backup, object, object backup storage group or tape storage group. Volumes are not associated with VIO storage groups.

Select option 6, Storage Group, on the ISMF Primary Option Menu for storage administrators. This displays the Storage Group Application Selection panel.

Select option 1, List, on the Storage Group Application Selection panel.

Optionally, type / next to one or more list options, which allow you to specify view or sort criteria, and to see space information in gigabytes (GB).

Type LISTVOL in the Line Operator column next to a storage group. This displays volume information, such as volume serial numbers, for the storage group. The LISTVOL line operator without any parameters lists all volumes in a storage group. The list displayed depends on the storage group type. See [Table 1](#) for more information. [Table 1. Volumes Listed in a Specified Storage Group](#)

The DASD Volume STATUS line operator allows users to display 32 sets of SMS and MVS volume statuses corresponding to 32 system names, system group names or both for a volume from the Volume List panel. The Volume List panel only displays eight of the possible 32 sets of SMS and MVS volume statuses at a time. In order to view volume statuses beyond the first eight sets, this line operator has to be entered against the volume on the Volume List Panel. The STATUS line operator can be abbreviated to ST.

Refer to [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support](#) for information on the Mountable Optical Volume List.

Refer to [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries](#) for information on the Mountable Tape Volume List.

Parent topic:[Defining storage groups](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining reserve storage pools

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This topic describes reserve storage pools.

[Understanding reserve storage pools](#)

[Assigning volumes to a reserve storage pool](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Understanding reserve storage pools

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Reserve storage pools are groups of volumes that have been formatted for future use. Reserve storage pools can make it easier for you to manage your defined but unused volumes. You give each reserve storage pool a name. You can have any number of reserve storage pools.

Using reserve storage pools can make it easier to manage your defined but unused volumes.

To belong to a reserve storage pool, a volume must have been initialized as reserved and have an owner ID beginning with IBMRSP followed by the name of a reserve storage pool. It then has a volume serial number but cannot be brought online. It remains offline until it is re-initialized as a non-reserved volume.

Parent topic:[Defining reserve storage pools](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Assigning volumes to a reserve storage pool

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To assign a volume to a reserve storage pool, you initialize the volume using the ICKDSF INIT command with these parameters: **RESERVED**Initializes the volume as reserved, which specifies that it cannot be brought online.**OWNERID**Specifies the name of the reserve storage pool, in the form IBMRSP*rspname* where *rspname* is the name of the reserved storage pool (up to 8 characters). OWNERID is optional. A volume that has been initialized as reserved cannot be brought online. Before it can be brought online, it must be re-initialized without the RESERVED keyword. For more information, refer to the topic about the [INIT command](#) in Device Support Facilities (ICKDSF) User's Guide and Reference.

A volume that has been initialized as reserved is not to be confused with a device for which a system issues a hardware RESERVE, either through the RESERVE or ISGENQ macros, to ensure successful serialization of shared DASD. For more information, refer to [z/OS MVS Programming: Authorized Assembler Services Reference LLA-SDU](#) or [z/OS MVS Programming: Authorized Assembler Services Reference EDT-IXG](#).

Parent topic:[Defining reserve storage pools](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DFSMShsm manages non-SMS data sets at the volume level, applying the management criteria to all data sets on a given volume. SMS automates the management of storage at the data set level by introducing management classes. When assigned to data sets, management classes replace and expand attributes that otherwise would be specified on JCL DD statements, IDCAMS DEFINE commands, and DFSMShsm commands. When assigned to objects, OAM uses a subset of the management class attributes and the OSMC class transition attributes to manage objects. This topic describes management classes and shows you how to define them using the ISMF management class application.

With the Copy Technique attribute, the storage administrator has the option of specifying whether a system-managed backup process called concurrent copy should be used for data sets to enhance system availability during data set backup and aggregate backup processing. Data Set Alter can be used to alter a management class so data sets can use the concurrent copy technique during backup processing. The storage administrator has the option of specifying whether the concurrent copy process is required, preferred, or discouraged. The Copy Technique attribute is related to the Backup and Aggregate Backup components of management class. A Copy Technique field exists for each of the Backup and Aggregate Backup components. These fields appear on the management class list, display, define/alter, view, and sort panels.

[Understanding management classes](#)

[Planning management classes](#)

[Defining management class attributes](#)

[Assigning management classes](#)

[Defining additional management classes](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Understanding management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A management class is a list of data set migration, backup and retention attribute values. Management class also includes object expiration criteria, object backup requirements, and class transition criteria for management of objects. DFSMSHsm uses the attributes of the management class associated with a data set to manage storage. When you assign a management class to a system-managed DASD data set, SMS places the management class name in both the basic catalog structure (BCS) and the VSAM volume data sets (VVDS) catalog entries of the data set. Management class is optional for system-managed data sets and does not apply to non-system-managed data sets. Management class is *not* used for tape data sets. If you alter a management class definition, SMS applies the changes to any new data sets or objects after you activate the changed configuration. SMS also applies the changes to any previously allocated data sets or objects, beginning with their next scheduled management cycle (such as daily space management or backup).

[Default management class](#)

[OAM management classes](#)

[Describing management classes](#)

Parent topic:[Defining management classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Default management class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For data sets, you can specify a default management class in an SCDS. DFSMSHsm applies the default management class to all system-managed data sets that do not already have a management class. Unlike the data sets that you have already assigned a management class, the catalog entries of these data sets do not contain a management class name. For objects, the default management class is defined in the catalog entry for the object collection to which the objects belong. The default management class is assigned by the management class ACS routine when the first object is stored in the collection.

Parent topic:[Understanding management classes](#)



©Copyright IBM Corporation 1990, 2014

OAM management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

OAM uses some attributes in the management class associated with the object to manage the object. Class transition attributes allow OAM to change the way an object is managed based on its age, its usage, or a predefined periodic function. A class transition is a change in the object's management class or storage class when an event occurs which brings about a change in an object's management criteria or service level. Class transitions occur during an OSMC storage management cycle. Objects requiring class transition use the ACS routines to determine if they should be managed using a different management class or placed at a different level of the storage hierarchy according to a new storage class.

OAM uses the backup attributes of the management class definition to initiate the writing of one or two backup copies of an object. An Auto Backup of 'Y' indicates to OAM that backup copy(s) are required for this object. A Backup Frequency of zero indicates that OAM is to schedule the first backup copy be written at the time the object is initially stored, whereas a non-zero value results in the first backup copy being written during the first storage management cycle after the object has been stored. If two backup copies are desired, specify a value of two or greater in the management class Number of Backup Versions attribute, and specify the SETOSMC statements in your CBROAMxx member of PARMLIB using SECONDDBACKUPGROUP to invoke multiple object backup support.

Tip: The default number of backup copies is two. If you want only one backup copy, you must set the management class definition for the number of backup versions to one.

Retention attributes determine the OAM action for expiration of objects. An object can expire automatically based on its age, its usage, or a specific date derived from its management class and an object-specific retention period, if provided. OSMC deletes expired objects from the directory automatically during the storage management cycle with the approval of the auto-delete installation exit.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for a more specific discussion of how to use management classes with objects.

Parent topic:[Understanding management classes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Describing management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A management class definition contains both descriptive and storage-related information. To identify and refer to your management classes, you assign each one a unique name that contains from one to eight alphanumeric characters. Each management class definition maintains an owner ID that identifies the storage administrator who originally created or last modified the management class. The *owner ID* is a z/OS® user ID on the ISMF Management Class List in column 19-Last Modified Userid. Also, each management class contains an optional 120 character description field for describing its contents.

Parent topic:[Understanding management classes](#)



©Copyright IBM Corporation 1990, 2014

Planning management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You should create a management class for each type of service that is to be provided by the installation. A type of service is defined for a collection of data sets that have similar migration, backup, and retention requirements or objects that have similar backup, expiration, and class transition requirements. Before you actually define your management classes, you should gather the following information:

Requirements for releasing over-allocated space
Migration requirements
Retention criteria
Treatment of expired data sets
Frequency of backup
Number of backup versions
Retention of backup versions
Number of backup versions
Retain only version
Retain only version unit
Retain extra versions
Retain extra versions unit
Copy serialization
Generation data group (GDG) information
Class transition criteria
Transition copy technique
Serialization exit

Based on this information, you can establish management classes that centralize storage management in the SMS complex.

If a data set has a management class that specifies automatic backup, migration, or transitions, then you must direct the data set to storage groups that are eligible to be processed for automatic backup or migration capabilities.

A generation data group (GDG) is a group of related cataloged data sets that have sequentially ordered names. SMS uses GDG-related information in the management class definition to manage the storage associated with these data sets. Some management criteria is specified in the definition of the GDG. Some management criteria might be specified by assigning a management class to each individual generation in the GDG. Generation data sets within the same GDG might have different management classes assigned by JCL.

DFSMSrmm uses management class names, instead of attributes, for tape data set retention and movement. It matches a management class name with a DFSMSrmm vital record specification (VRS) and implements the policies defined in the VRS.

Parent topic:[Defining management classes](#)



Defining management class attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define management classes.

Select option 3, Management Class, from the ISMF Primary Option Menu for storage administrators. This displays the Management Class Application Selection panel.

Supply values on the Management Class Application Selection panel: **CDS Name**This is the name of an SCDS. ISMF primes the CDS Name field with the last used SCDS name. The default is 'active', which represents the currently active configuration. You cannot define or alter the management classes for the active configuration.**Management Class Name**This is the name of the management class. Your management class names should be generic rather than specific, so that attribute changes do not make names meaningless or misleading. STANDARD and INTERIM are examples of generic names. RETAIN30 and BUDAILY are examples of specific names. ISMF primes the field with the name last used.**Option**Select option 3, Define. This displays the Management Class Define panel. You can use the DOWN command to display the second page, where you can select an attribute group to display first. Selecting an attribute group is only useful when performing a display, define, or alter function. You select an attribute group for processing ahead of other Management Class attributes by entering its corresponding number in the selection field. You can display panels containing the other management class attributes by paging up and down in the usual manner.

Supply values on the Management Class Define panel. Use the DOWN command to move to the next page. For more information, see: [Defining management class expiration attributes](#)[Defining management class migration attributes](#)[Defining management class backup attributes](#)[Defining class transition attributes](#)[Defining aggregate backup attributes](#)

You can leave any page of the Management Class Define panel without saving the management class by issuing the Cancel command.

[Defining management class expiration attributes](#)
[Defining management class migration attributes](#)
[Defining management class backup attributes](#)
[Defining class transition attributes](#)
[Defining aggregate backup attributes](#)

Parent topic:[Defining management classes](#)

©Copyright IBM Corporation 1990, 2014

Defining management class expiration attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Page 1 of the Management Class Define panel contains the management class expiration attributes. DFSMShsm processes the expiration attributes before the migration attributes that you specify on the second page of the Management Class Define panel. SCDS Name and Management Class Name are output fields that contain the SCDS and management class names you specified in the Management Class Application Selection panel. Supply these values: **Description** is an optional field of 120 characters you can use to describe the management class. **Expiration**

Attributes Determine the action for data set and object expiration and deletion. DFSMShsm deletes expired data sets during automatic space management processing. OAM deletes objects during a storage management cycle with the approval of the auto-delete installation exit. Expiration attributes are required values that indicate when a data set or object becomes eligible for expiration. You can base expiration criteria on a specific date, on the number of days since the data set or object was allocated, or on the number of days since the data set or object was last referenced. **Expire after Days Non-usage** Specifies how much time must elapse since last access before a data set or object becomes eligible for expiration. The default is NOLIMIT. **Expire after Date/Days** Specifies an absolute date or period after its allocation for a data set or object to become eligible for expiration.

The default is NOLIMIT. **Retention Limit** Is a required value that limits the use of retention period (RETPD) and expiration date (EXPDT) values that are explicitly specified in JCL, are derived from management class definitions or are explicitly specified in the OSREQ STORE macro. If the value of a user-specified RETPD or EXPDT is within the limits specified in the Retention Limit field, it is saved for the data set. For objects, only RETPD is saved. The default retention limit is NOLIMIT. If you specify zero, a user-specified or data class derived EXPDT or RETPD is ignored. If users specify values that exceed the maximum period, the retention limit value overrides not only their values but also the expiration attributes values. The retention limit value is saved. ISMF primes the Retention Limit field with what you specified the last time.

[The use of expiration attributes and retention limit](#)

Parent topic: Defining management class attributes



The use of expiration attributes and retention limit

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DFSMShsm determines if a data set has expired based on the expiration date found in the catalog entry of the data set. OAM determines if an object has expired based on the expiration date in its object directory entry. If an expiration date is not found, DFSMShsm and OAM use the management class expiration attributes.

These attributes are used as follows: If both expiration attributes are NOLIMIT, the data set or object never expires. If one of the expiration attributes is NOLIMIT, then the other attribute must be satisfied. If neither expiration attribute is NOLIMIT, both of the expiration attributes must be satisfied.

Tip: If you want to change the expiration date in a catalog entry, you can either use the access method services ALTER command or you can specify either the RETPD or EXPDT parameter in the JCL for an old managed non-VSAM data set, as long as the new expiration date is valid based on the retention limit specified. See z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support for information on changing the expiration date of an object.

Data sets or objects having the INTERIM management class defined in [Defining management class expiration attributes](#) become eligible for expiration when both of the following criteria are met: at least seven days since allocation and not referenced in the last two days.

Table 1 shows several combinations of retention attributes used for space management processing. Use the highlighted values for each instance. Table 1. Comparing Retention Period Attributes

In the first case, the retention limit is zero, so DFSMShsm and OAM honor the values of 50 and 100 which are the values specified for Expire after Days Non-usage and Expire after Date/Days and ignore any user-specified or data class values. In Case 2, the management class expiration values are ignored because RETPD and EXPDT values have been specified or derived and the retention limit is nonzero. However, because the retention limit is less than the user-specified or data class values, 50 is saved and used to calculate the expiration date.

In Case 3, the user-specified or data class values fall within the retention limit. So, DFSMShsm uses the values of 60 and 60. OAM uses a RETPD value of 60, because OAM does not consider EXPDT.

In Case 4, the RETPD and EXPDT are used because the retention period value is NOLIMIT.

In Case 5, because no user-specified or data class derived values are available, DFSMShsm and OAM use the values specified in the management class expiration attributes which are shown on the Management Class Define panel.

After you specify the expiration attributes, issue the DOWN command to see the next page of the Management Class Define panel, on which you can specify the migration attributes.

Parent topic:[Defining management class expiration attributes](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining management class migration attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To DFSMSHsm, a data set occupies one of two distinct states in storage:

PrimaryAlso known as *level 0*, the primary state indicates that end users can directly access a data set residing on a volume.**Migrated**End users cannot directly access data sets that have migrated from the primary state to a migrated state. To be accessed, the data sets must be recalled to primary storage. A migrated data set can reside on either *migration level 1* (usually permanently mounted DASD) or *migration level 2* (usually tape). A data set can move back and forth between these two states, and it can move from level 0 to migration level 2 (and back) without passing through migration level 1. Objects do not migrate. Movement back to level 0 is called *recall*.

Page 2 of the Management Class Define panel contains the management class migration and GDG management attributes. SCDS Name and Management Class Name are output fields that contain the SCDS and management class names you specified in the Management Class Application Selection panel.

To define a management class, you must specify values for the Partial Release and Command or Auto Migrate attributes. If you specify BOTH for the Command or Auto Migrate attribute, then Primary Days Non-usage and Level 1 Days Non-usage become required fields. If you specify NONE for the Command or Auto Migrate attribute, the expiration attributes specified on page 1 of this panel still apply. The other fields on this panel are optional.

Partial Release AttributeApplies to non-VSAM data sets and to all VSAM data set types allocated in the extended format. You can use partial release to choose the conditions under which unused allocated space is released. If you select partial release, space release is carried out automatically at the time you have selected. Each partial release option releases the same amount of space. Unused space is released in cylinders or tracks, depending on the space allocation unit used. The following options are available for partial release: **Y**Yes. Release unused space at Space Management cycle time.**YI**Yes Immediate. Release unused space at Space Management cycle time, when close is issued for a data set that was open for output, and on subsequent volumes.**C**Conditional. If a nonzero secondary space allocation has been specified for the data set, release unused space at Space Management cycle time. This option only applies to physical sequential and partitioned data sets. If specified for an extended format VSAM data set, this option is processed as if Yes were specified.**CI**Conditional Immediate. If secondary space has been allocated, release unused space at Space Management cycle time, when close is issued for a data set that was open for output, and on subsequent volumes.**N**No. No release of unused space.**Note:**ISMF primes the field with the last used value. The default is N. The Partial Release attribute is not checked during DFSMSHsm interval migration. The Partial Release attribute is mutually exclusive with the Guaranteed Space storage class attribute. Partial release is ignored under the following conditions: Another job is sharing the data set (DISP=SHR). Another task in the same job is processing an OPEN, CLOSE, EOV, or FEOV request for the data set. Another DCB is open for the data set.

For more information on partial release, refer to z/OS DFSMS Using Data Sets.

Migration AttributesRepresents the minimum number of days that must elapse since last access before a data set is eligible for normal migration. Days refers to calendar days, not to a 24-hour time period. In other words, the day you close the data set counts as the first full day of nonusage. The minimum number of Primary Days Non-usage days required for migration to ML2 includes the time already spent unreferenced on ML0. The default is 2. (Refer to **GDG Management Attributes**, below, for the special GDS migration eligibility rules.) A nonzero number for Level 1 Days Non-usage indicates the number of days that must elapse since the last reference on level 0 before a data set on ML1 can migrate from level 1 to level 2. The default is 60.

The Command or Auto Migrate attribute allows you to specify if a data set is eligible to be migrated by BOTH command and automatic processing, by command alone, or not at all (NONE).

If you do not want to migrate data sets that belong to a particular management class, specify NONE in the Command or Auto Migrate field. The data sets remain on primary storage until they expire. If you want data sets to be eligible for migration directly from level 0 storage to level 2 tape, specify 0 in the Level 1 Days Non-usage field. Otherwise, the data sets must first migrate to level 1 (the days spent on level 0 count toward the eligibility of moving to level 2).

GDG Management Attributes Indicate criteria for early migration of GDGs off primary storage and what to do with rolled-off generation data sets (GDSs). The # GDG Elements on Primary attribute indicates how many of the most recent generations of a GDG use the normal Primary Days Non-usage attribute for migration criteria. Generations that are over this limit are eligible for early migration during the next processing of primary space management. The inactive age is irrelevant for these 'over the limit' GDSs and they are given a higher selection priority so they are likely to be migrated if space is needed to meet volume thresholds.

Tip: Specifying the # GDG Elements on Primary attribute does not ensure that the specified number of GDSs are kept on primary storage, but rather that their migration is not accelerated.

To specify that old generations are to have priority for migration and to specify how many generations are to 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. Those data sets made eligible for early migration do not have to satisfy the Primary Days Non-usage criteria.

All generations that meet the Primary Days Non-usage criteria are eligible to migrate. # GDG Elements on Primary enables you to accelerate migration of older generations while Primary Days Non-usage enables you to specify migration criteria of younger generations separately.

The Rolled-off GDS Action value indicates whether to expire rolled-off GDSs or to make them eligible for migration. If you specify MIGRATE, management class expiration attributes are applied to the data set to determine if the data set should be deleted; if not, the data set is eligible for migration. If you specify EXPIRE, the rolled-off GDS is deleted unless an explicit expiration date is in the DSCB. In this case, the rolled off GDS becomes a non-GDG data set and migrates according to non-GDG migration attributes.

Tip: The DFSMSHsm parameter, EXPIREDATASETS, controls the deletion of a data set with an expiration date in the DSCB.

Both the # GDG Elements on Primary and the Rolled-off GDS Action attributes are optional and have default values of blank. If left blank, no special treatment results from the data set being a GDS or a rolled-off GDS. That is, other management class attributes is used to process the data set.

During automatic primary space management, GDSs that have been determined as eligible for migration are migrated in the following priority order:

GDSs that are rolled-off.GDSs that exceed the limit specified in # GDG Elements on Primary.Other GDSs are handled by the non-GDG algorithm which bases priority on a function of size and the length of time it has been eligible to migrate in Primary Days Non-usage.After specifying the migration attributes, issue the DOWN command to specify the backup attributes on the next page of the Management Class Define panel.

Parent topic:Defining management class attributes



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining management class backup attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Page 3 of the Management Class Define panel contains the management class backup attributes. (For objects, see [Defining class transition attributes](#).)

Backup FrequencyFor DFSMShsm, the Backup Frequency attribute specifies how many days must elapse before DFSMShsm can back up data sets that have changed since the last backup. If you want to back up changed data sets every time that DFSMShsm processes the volumes containing them, specify 0. The default is 1.

For DFSMSdfp OAM object support, if Auto Backup = 'Y', the Backup Frequency attribute specifies when the first backup copy of an object is to be written. If 0 is specified, then OAM will schedule a backup copy at the time the object is initially stored. If non-zero is specified then OAM will schedule backup copy(s) to be written during the OAM Management Component (OSMC) storage group processing cycle. The default is 1.

Number of Backup VersSpecifies the maximum number of backup versions to retain for a data set. The default is 2 if the data set still exists and 1 if it has been deleted. Creating a new backup version when the number of backup versions already equals the value specified for the appropriate *Number of Backup Vers* attribute (Data Set Exists) causes the oldest version of the appropriate type to be deleted.

The number of backup versions is used to determine whether OAM should write one or two backup copies of the objects, when you activate the SECONDBACKUPGROUP function for objects using SETOSMC in the CBROAMxx member of PARMLIB. If the number of backup versions is greater than 1 and AUTO BACKUP is Y, OAM will create two backup copies. When the original object is expired or deleted, all backup copies are also deleted.

Retain Days Only Backup VerIndicates how many days to keep the most recent backup version of a deleted data set, starting from the day DFSMShsm detects it has been deleted. This attribute applies only when a data set no longer exists on primary (level 0) or migrated (levels 1 and 2) storage. The default is 60. This field does not apply to objects. Backup copies of objects are not retained when the original object is deleted.

Retain Days Extra Backup VersIndicates how many days to keep backup versions other than the most recent one, starting from the day backups were created. It applies only when more than one backup version exists, and when a data set has low activity. This attribute applies whether the data set has been deleted or not. The default is 30. The number of extra versions is the number of backup versions minus one. If you specify 1 for Number of Backup Vers, there are no extra versions. For example, if you specify 3 for *Number of Backup Vers (Data Set Deleted)*, the number of "extra" versions for deleted data sets is 2. These 2 versions are managed according to the *Retain Days Extra Backup Vers* attribute. Any other versions that may have existed when the data set was deleted will be deleted the next time EXPIREBV is processed.

Admin or User Command BackupIndicates if both the end user and the storage administrator can issue command backups of the data sets in this management class, if only the storage administrator can, or if neither of them can. The Admin or User command Backup field is required, and has a default value of BOTH.

Otherwise, the remaining fields are optional. **BOTH**Indicates that both storage administrators and users can perform command backups against these data sets.**ADMIN**indicates that only storage administrators can perform command backups against these data sets.**NONE**indicates that neither storage administrators or end users can perform command backups.If you specify BOTH or ADMIN in the Admin or User Command Backup field, the remaining fields on this panel are required. If the value is BOTH or ADMIN, then AUTO BACKUP specifies whether these data sets are eligible for automatic backup.

Auto Backups is required, and has a default value of Y. If you specify Y in the Auto Backup field, the remaining fields on this panel are required. Otherwise, the remaining fields are optional. OAM backs up the object during the first management cycle after the object is assigned to a management class requiring backup. This can occur when the object is stored or when its management class is changed by you or by a class transition. Backup copies of an object are deleted when the object is deleted. No archived copy is saved.

Backup Copy Technique Specifies whether concurrent copy should be used during data set backup processing. When deciding whether or not to use concurrent copy for a particular dump or copy operation, you need to decide why you are making a copy of the data. If the intention of the copy is to capture a point-in-time image of the data at a specific time, concurrent copy might not be appropriate. When you use concurrent copy there is always a chance that a dump or copy operation does not complete once you have started to make updates to the data. If this happens, the specific point-in-time is lost and you might not be able to recover an image of the data at that time. **AR|R** indicates DFSMSdss keyword

CONCURRENT(ANYREQ) **VR** indicates DFSMSdss keyword

CONCURRENT(VIRTUALREQ) **CR** indicates DFSMSdss keyword

CONCURRENT(CACHEREQ) **APIP** indicates DFSMSdss keyword

CONCURRENT(ANYPREF) or **CONCURRENTVP** indicates DFSMSdss keyword

CONCURRENT(VIRTUALPREF) **CP** indicates DFSMSdss keyword

CONCURRENT(CACHEPREF) **S** indicates not specifying CONCURRENT keyword to DFSMSdss. This is the default.

Recommendation: Avoid specifying conflicting conditions in the storage class and the management class. For example, if you specify Backup Copy Technique as R (required) in the management class and Accessibility as NOPREF in the storage class, you would be requesting concurrent copy for backup while having placed the data set on a device that does not support concurrent copy, which would result in the backup failing. Specifying the backup copy technique option in the management class in no way affects the placement of data sets on a concurrent copy volume.

After you specify the backup attributes, issue the DOWN command to see the next page of the Management Class Define panel, on which you can specify the object attributes.

Parent topic: Defining management class attributes



Defining class transition attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Pages 4 and 5 of the Management Class Define panel contains the class transition attributes.

Time Since Creation Years, Months, or DaysIndicate the time since the creation date that must pass before transition occurs.
Time Since Last Use Years, Months, or DaysIndicate the time since the last reference date that must pass before transition occurs.
PeriodicIndicates a time based on the calendar at which transition occurs. Restriction: The Time Since Creation, Time Since Last Use, and Periodic fields cannot be specified together. A maximum date of 9999/12/31 is used if the requested Time Since Creation or Time Since Last Used exceeds the maximum date.

Monthly On DaySpecifies the day of each month that the transition occurs. If there are fewer days in the month than the number specified, the transition occurs on the last day of the month.
Quarterly On Day or In MonthSpecify the time of each quarter that the transition occurs. If both Day and Month are specified, this attribute specifies the day of the month in each quarter that the transition occurs. If there are fewer days in the specified month than the number specified in Day, then the transition occurs on the last day of the specified month.

Yearly On Day or In MonthSpecify the day or month of each year that transition occurs. If both Day and Month are specified, this attribute specifies the day of the month in each year that transition occurs. **FIRST**Specifies that the transition occurs on the first day of each month, quarter, or year, whichever attribute is specified.
LASTSpecifies that the transition occurs on the last day of each month, quarter, or year.

Transition Copy TechniqueSpecify which copy technique should be used for the class transition of data associated with this management class.
FRPFast replication
FRRFast replication required
STDStandard (the default)
PMPFlashcopy
Preserve Mirror preferred
PMRFlashcopy Preserve Mirror requiredFor more information about fast replication, FlashCopy®, Preserve Mirror and other copy services functions, refer to z/OS DFSMS Advanced Copy Services.

Serialization Error ExitSpecify the database or exit to invoke when there is a serialization error. **DB2**
DB2CICSCICS Note: See [Sample Batch Job for CICS Definitions](#) for a sample batch job that will define the group, transaction, program, session and connection to allow the CICS server program to process the closing and opening of files.

ZFSz/OS File System
EXITUser exit

NONENone (the default)

Parent topic:[Defining management class attributes](#)

©Copyright IBM Corporation 1990, 2014

Defining aggregate backup attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Page 6 of the Management Class Define panel contains the aggregate backup attributes. The SCDS Name field specifies the name of the SCDS into which the management class is defined.

VersionsSpecifies the number of versions to be maintained.**Retain Only**

VersionIndicates how long the most recent backup version of an aggregate group is kept.**Unit**Specifies the unit of measure for the time period specified for the Retain Only Version field. This field cannot be blank if the Retain Only Version field is specified, except when Retain Only Version is NOLIMIT.**Retain Extra**

VersionsIndicates how long to keep backup versions of an aggregate group that precede the most recent version.**Unit**Specifies the unit of measure for the time period specified for the Retain Extra Versions field. This field cannot be blank if the Retain Extra Versions field is specified, except when Retain Extra Versions is NOLIMIT.**Copy Serialization**Specifies whether aggregate backup should continue if an enqueue failure is encountered.**Backup Copy Technique**Specifies whether concurrent copy is to be used to back up aggregates of data sets associated with this management class. The default value is STANDARD. Valid values for this field are:

CPCACHE PREFERRED**R** CACHE REQUIRED**P**CONCURRENT
PREFERRED**R**CONCURRENT REQUIRED**S** STANDARD**V**PVIRTUAL
PREFERRED**V**RVIRTUAL REQUIRED

Parent topic:[Defining management class attributes](#)



©Copyright IBM Corporation 1990, 2014

Assigning management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can assign management classes through the management class ACS routine, by explicit specification or as part of a class transition. For permanent system-managed data sets, the management class ACS routine is executed only if the storage class is valid. The management class ACS routine is not executed for temporary data sets.

An object's management class is determined when you store the object. The default management class for an object is defined in the collection to which the object belongs. (The default management class for an object collection is assigned by the management class ACS routine when the first object is stored in the collection.) This can be overridden if you specify a management class when you store the object. Then, the management class ACS routine is run to assign a management class to the object. The management class of an object can be changed with the OSREQ CHANGE macro or through class transition.

You can explicitly specify a management class on the following, but a management class determined by the management class ACS routine takes precedence: JCL DD statements TSO/E ALLOCATE command DFSMSdss COPY and RESTORE commands Access method services ALLOCATE, DEFINE, and IMPORT commands Dynamic allocation requests through ISPF/PDF data set allocation panels OSREQ STORE and OSREQ CHANGE macro requests End users cannot override any of the attribute values that you assign to a data set or object through the management class ACS routine except for the expiration attributes, on the condition that the retention limit is not exceeded (for data sets or objects) or set to NOLIMIT (for data sets). See [Defining management class expiration attributes](#) for information about the order of precedence for expiration attributes.

The syntax for specifying a management class on a JCL statement is:

MGMTCLAS=management-class-name

The syntax for specifying a management class on a TSO/E command is:

MGMTCLAS (management-class-name)

The syntax for specifying a management class on an access method services command is:

MGMTCLAS(-) or MANAGEMENTCLASS(management-class-name)

The syntax for specifying a management class on an OSREQ STORE or CHANGE macro is:

MGMTCLAS=management-class-area or MGMTCLAS=(management-class-area-pointer)

For information on determining management classes through ACS routines, see [Defining ACS routines](#).

Parent topic:[Defining management classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining additional management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can copy existing management classes and modify them to create new management classes by using the COPY line operator, which is explained in [Copying SMS components](#).

Parent topic:[Defining management classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining storage classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Without SMS, you must place critical data sets on selected storage devices in order to improve performance. For example, if you have data sets that consistently require short response times, you place them on DASD volumes that have low I/O rates or that are connected to cache storage controllers. If you have data sets that require continuous availability, you can place them on dual copy, RAID volumes, peer-to-peer remote copy (PPRC), or extended remote copy (XRC).

SMS uses storage classes to separate data set performance objectives and availability from physical storage. It also provides attributes for the following:
Sequential data striping for batch processingAllocation to a volume that supports concurrent copy, SnapShot, or FlashCopy®.VSAM record-level sharing between different systems with storage classes defined to use the coupling facility (CF).
This topic describes storage classes and shows you how to define them through ISMF.

Restriction: *Duplication of Library Volser*: DUPT@SMS is an IBM® reserved name and must not be defined in SCDS by the installation.

With STORCLAS=DUPT@SMS and DISP=OLD and VOL=SER=nnnnnn specified in the JCL, you can access an imported tape outside an automated library for input if you have another tape with the same volser in a system-managed library.

[Understanding storage classes](#)

[Planning storage classes](#)

[Defining storage class attributes](#)

[Assigning storage classes](#)

[Defining additional storage classes](#)

[SMS volume selection for data set allocation](#)

[Defining secondary lock tables](#)



©Copyright IBM Corporation 1990, 2014

Understanding storage classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A storage class is a list of storage objectives and requirements. Each storage class represents a list of services that are available to data sets and objects having similar access requirements. A storage class does not represent any physical storage, but rather provides the criteria that SMS uses in determining an appropriate location to place a data set or object.

In general, SMS attempts to select a location that meets or exceeds the specified objective, but it does not guarantee response time. If no location satisfies the performance objective, SMS attempts to find a location that most closely matches the specified objective for DASD data sets. If there is more than one device of similar characteristics to choose from, SMS selects the volume with the most available space for a majority of times.

Only you, as a storage administrator, can define or alter the storage classes for an SMS complex. End users can specify but cannot override storage classes assigned for a data set by the ACS routines.

[Storage classes for data sets](#)

[Storage classes for objects](#)

Parent topic:[Defining storage classes](#)



©Copyright IBM Corporation 1990, 2014

Storage classes for data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

With the exception of tape data sets, a data set is system-managed if it has a storage class. When you assign a storage class to a system-managed DASD data set, SMS places the storage class name in both the BCS and the VVDS catalog entries of the data set.

To satisfy the availability requirements of critical data sets, SMS selects a volume that can provide data access even in the event of a single device failure. For example, this can be satisfied by a device that has dual copy active or that is an array DASD.

If you need to access data sets while they are being copied or backed up, you can place them on volumes that support concurrent copy, virtual concurrent copy, or FlashCopy®.

With storage classes, you can also specify whether a data set is referenced primarily in the read or write mode.

For information on establishing dual copy volumes and enabling or disabling cache functions, see [SETCACHE functions and device information](#). In this section, you can also find information on IBM® storage control units that support these features. Tape data sets are not SMS-managed, even though you can assign them a storage class. Only tape volumes are SMS-managed. Therefore, none of the storage class attributes apply to tape data sets. Tape data sets do not have to be cataloged. If you do catalog a tape data set, the SMS information, such as data classes, storage classes, and management classes, is not saved in the catalog. You can direct tape data sets with a storage class to an SMS-managed tape volume by using the storage group ACS routine.

Parent topic:[Understanding storage classes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Storage classes for objects

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For OAM, storage classes separate performance objectives from physical storage and determines where the object resides in the storage hierarchy.

OAM uses storage classes to logically represent the level of service required by an object. This could result in placement of the object on one of these types of physical storage in its object storage hierarchy:

- Direct access storage devices (DB2® object tables)
- File system (NFS or zFS)
- Optical volumes
- Tape volumes

The default storage class for an object is defined in the catalog entry for the object collection to which the object belongs. If the default management class is not available from the catalog entry, OAM uses the collection entry in the DB2 table and updates the catalog entry. The default storage class is assigned by the storage class ACS routine when the first object is stored in the collection.

Note: If the catalog entry for the collection does not exist, but a DB2 table entry does, this is an error. If neither exist, both are created during a first time store to a new collection, using the storage class, management class, and storage group from the ACS routine.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for a more specific discussion of how to use storage classes with objects.

Parent topic:[Understanding storage classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Planning storage classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You should create a storage class for each level of service provided by the installation. Before you actually define storage classes, you need to identify the hardware available in your installation. The fastest response time and data sharing for VSAM data are obtained from the coupling facility (CF). The fastest response time for DASD data sets is obtained from cached devices, such as devices behind a 3990 Storage Control with cache active, an IBM® RAMAC Virtual Array, or an IBM ESS. Object storage classes should reflect the hardware available in the object storage hierarchy.

Parent topic:[Defining storage classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining storage class attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A storage class definition contains both descriptive and access-related information. To identify and refer to storage classes, you assign each one a unique name that contains from one to eight alphanumeric characters. Each storage class definition maintains an owner ID that identifies the storage administrator who originally created or last modified the storage class. Also, each storage class contains an optional 120 character description field for describing its contents.

This section discusses defining storage class attributes. For related topics, see: [Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF](#) for information on generating lists of storage classes [Altering storage classes](#) for information on altering a storage class z/OS DFSMS Using the Interactive Storage Management Facility for details on displaying storage class information

Perform the following steps to define storage class attributes.

Select option 5, Storage Class, from the ISMF Primary Option Menu for storage administrators. This displays the Storage Class Application Selection panel.

Supply values on the Storage Class Application Selection panel: **CDS Name** This is the name of an SCDS. ISMF primes the CDS Name field with the last used SCDS name. The default is 'active'. This represents the currently active configuration. You cannot define storage classes for the active configuration.

Storage Class Name This is the name of the storage class. ISMF primes the Storage Class Name field with the name last used within ISMF. **Option** Select option 3, Define. This displays the Storage Class Define panel.

Supply values on the Storage Class Define panel. The SCDS Name and Storage Class Name fields are output fields that contain the SCDS and storage class names you specified in the Storage Class Application Selection panel. **Description** Is an optional 120-character description of the storage class.

For more information on the storage class attributes, see: [Defining performance objectives](#) [Defining availability](#) [Defining accessibility](#) [Defining the guaranteed space attribute](#) [Defining guaranteed synchronous write](#) [Defining use of the coupling facility for VSAM record-level sharing](#) [Defining when to disconnect the sphere](#)

You can leave the Storage Class Define panel at any time without saving the storage class by issuing the CANCEL command.

The following sections describe the storage class attributes you can define.

[Defining performance objectives](#)

[Defining availability](#)

[Defining accessibility](#)

[Defining the guaranteed space attribute](#)

[Defining guaranteed synchronous write](#)

[Defining use of the coupling facility for VSAM record-level sharing](#)

[Defining when to disconnect the sphere](#)

Parent topic: [Defining storage classes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining performance objectives

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In the performance objectives fields of the Storage Class Define panel, you can request millisecond response (MSR) times and indicate the bias of both direct and sequential access data sets. All of the performance attributes are optional. If you leave all MSR and bias fields blank (direct and sequential), SMS ignores device performance during volume selection. With the introduction of the IBM® Enterprise Storage Server® (ESS) and large cached controllers, the storage class MSR value is less significant except as noted below.

Tip: The MSR and bias values you specify in the storage class can be used to determine how buffers are to be allocated when system-managed buffering is used for VSAM applications. To do this, specify a value of System for the Record Access Bias attribute in the data class. This capability applies only to system-managed VSAM data sets allocated in extended format and accessed by batch applications. MSR values are also used to cache PDSE members and HFS files. For guidance on specifying MSR and bias values in the storage class, see [Processing Techniques](#) in z/OS DFSMS Using Data Sets.

[Defining millisecond response time](#)

[Defining bias](#)

[Defining initial access response seconds](#)

[Defining sustained data rate](#)

Parent topic:[Defining storage class attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining millisecond response time

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The MSR serves two purposes in SMS. First, it is used as the performance objective for selecting candidate volumes for new data set placement. During a new data set allocation, SMS searches for a volume that meets or closely matches this objective. If no volume satisfies the objective, then SMS attempts to find a volume that comes closest to matching it. If more than one MSR is explicitly or implicitly specified, the storage class and associated device MSRs are averaged and compared.

Second, if the data is placed on a volume attached through an IBM® 3990 Storage Control with cache, and cache is enabled for that volume, the MSR is used to determine if caching is mandatory, optional, or should be inhibited for the data set. This attribute does not apply to objects.

You can request SMS to ignore various device performances during volume selection by leaving all MSR and BIAS fields blank. This lets you spread data evenly across noncached and cache active devices.

[Millisecond response time and data set allocation](#)

[Millisecond response time and cache management](#)

Parent topic:[Defining performance objectives](#)



©Copyright IBM Corporation 1990, 2014

Millisecond response time and data set allocation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DASD can have different performance capabilities for direct access (random access, for example) and for sequential access applications. Its performance capabilities depend on whether you are reading data or writing data. Each device type and model has a predetermined MSR capability for each condition. Additionally, if the device is attached to a cache capable control unit, the response capabilities are improved when caching is active. Therefore, each device is represented in [Table 1](#) by eight MSR values: Uncached Performance Direct Read MSR Direct Write MSR Sequential Read MSR Sequential Write MSR Cache performance, DASD Fast Write performance, or both (if active) Direct Read MSR Direct Write MSR Sequential Read MSR Sequential Write MSR If a device is cache capable, it must also have caching active at the time of allocation in order to be represented by the caching MSR values.

In table [Table 1](#), Read is abbreviated as R and Write is abbreviated as W.

Table 1. MSR Capabilities

From the numbers included in [Table 1](#), you can observe that: The fastest write performance that can be currently achieved is 6 milliseconds. The fastest read performance that can be currently achieved is 10 milliseconds. When cache fast write is not active, write performance is the same as for an uncached device. You can, for example, help direct allocation to a 3390-9 by specifying an appropriate MSR. A direct MSR of 35 milliseconds with a bias of read would favor a cache-active 3390-9.

To direct allocation to a solid state drive, specify a direct MSR of 1 and a direct bias of R. Do not specify a value for sequential MSR or sequential bias. To direct storage allocation away from solid state drives, specify a direct MSR of 10 and a direct bias of R. Do not specify values for sequential MSR or sequential bias.

Parent topic:[Defining millisecond response time](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Millisecond response time and cache management

z/OS DFSMSdfp Storage Administration
SC23-6860-01

At run time, when control unit caching is available, data is divided into three categories: must cache, may cache, and never cache.

Dynamic data set cache management is an SMS feature that permits expanded use of a storage controller that supports dynamic cache management enhancement (DCME) with cache, caching and DASD fast write features when these features are under-used or provides restricted use when they are overused. Only system-managed data sets that reside on volumes attached through a storage controller that supports DCME with cache are affected. Data sets not managed by SMS and data sets that reside on volumes attached through storage controls other than a storage controller that supports DCME with cache are not affected.

Dynamic cache management improves performance when a storage controller that supports DCME with cache is overloaded. It implements a cache management algorithm that optimizes the selection of data sets that are cache candidates. This enhancement also prevents over-commitment of nonvolatile storage. New I/O statistics by data set and SMS storage class can be collected and monitored by using the System Management Facility (SMF). Refer to z/OS MVS System

Management Facilities (SMF) for details on the structure of SMF type 42 records.

As a function of dynamic data set cache management, each system-managed data set is assigned a cache usage attribute and a DASD fast write usage attribute for sequential and direct accessing modes. These usage attributes are based on the MSR and bias specifications found in the storage classes associated with the data set. These attributes are:

- Cache usage attribute: The data set might have one cache usage attribute for sequential accessing mode and a different attribute for direct accessing mode.
- **Must Cache** The respective MSR specification can be met only through the use of cache. BIAS=Read, Write or blank. As the name implies, this is data that must always be cached in order to meet the performance requirement. If you specify an MSR that is lower than any device can provide without caching, then allocation places the data on a device closest to the requested MSR that has cache active (if one is available) and the data is cached at execution time (when cache is active) because the MSR cannot be met by the uncached performance of the device.

It is possible that as new devices are introduced, equivalent MSR might be obtainable without the use of cache. In that case, the data might be allocated to a noncached device, but the MSR is still achievable.

May Cache The respective MSR specification can be met without the use of cache. BIAS=Read, Write or blank. Any data that is on a cache capable control unit is considered a candidate for *may cache*. At execution time, if the data is neither must cache nor never cache, dynamic cache management determines if and when to cache the data.

Never Cache Never Cache data is data that is known to be cache unfriendly. An MSR of 999 indicates that data is never to be cached at execution time, even if it resides on a cache active device. SMS volume selection prefers a volume whose performance is equivalent to 25ms and has Cache inactive. See [Table 1](#).

DASD fast write usage attribute: The data set might have one DASD fast write usage attribute for sequential accessing mode and a different attribute for direct accessing mode.

- **Must DASD Fast Write** The respective MSR specification cannot be met without the use of DASD fast write. BIAS=Write.
- **May DASD Fast Write** The respective MSR specification can be met without the use of DASD fast write.

BIAS=Write, Read or blank.

- **Never DASD Fast Write** The respective MSR specification is 999.

Recommendation: Direct your IDCAMS query regarding the status of cache/DFW to the controller, because querying the device can lead to incorrect conclusions.

Parent topic:Defining millisecond response time



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining bias

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Bias determines which volumes MSR performance numbers (read, write, or both) to consider during volume selection. If you specify a read (R) bias, a cache storage control should be available to allow caching. If you specify a write (W) bias, the DASD fast write feature of an IBM® 3990 Storage Controller with cache should be available to allow the use of DASD fast write. If you do not specify a value for bias (blank), the MSR time determines whether caching or DASD fast write are used. Once the data sets have been allocated, the MSR time determines whether caching or DASD fast write are used. You can inhibit caching or DASD fast write by specifying a MSR time of 999. If a bias is specified without an accompanying MSR, an MSR of 6ms for write bias or an MSR of 10ms for read bias is used for volume selection. If the MSR is blank, then the data set is May Cache and May DASD Fast Write.

Tip: Allocation does not fail when a storage control device does not have caching or DASD fast write available.

[Table 1](#) can help you determine which setting (May, Must, Never) is set by DCME:
Table 1. D/T3990 SMS Cache Candidate Tokens for Sequential and Direct Requests

See [Table 1](#) about performance characteristics of a particular IBM device.
Set the storage class to indicate the desired performance for a data set. SMS volume selection and DCME work together to provide the closest performance possible at the time of allocation and open.

Parent topic:[Defining performance objectives](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining initial access response seconds

z/OS DFSMSdfp Storage Administration
SC23-6860-01

[VTS \(Virtual Tape Server\) cache management](#)

[Object access](#)

[Data set access](#)

Parent topic:[Defining performance objectives](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

VTS (Virtual Tape Server) cache management

z/OS DFSMSdfp Storage Administration
SC23-6860-01

An initial access response time (IART) of 100 or greater means the volume has least preference in the cache. 0-99 means the volume has most preference.

Parent topic:[Defining initial access response seconds](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Object access

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the Initial Access Response Seconds (IARS) field to specify the desired response time (in seconds) required for locating, mounting, and preparing a piece of media for data transfer. OAM uses this value to interpret the storage level, that is, to place an object at an appropriate level in the object storage hierarchy. For objects, the IARS, OAM Sublevel (OSL), and sustained data rate (SDR) are applicable.

OAM uses IARS as follows: If the IARS value is 0, OAM writes to either a DB2 table or a file system, depending on the OSL value, as described in [Planning storage groups for OAM object collections](#). If the IART value is 1-9999, OAM selects removable media, either optical or tape. If the SDR is greater than or equal to 3, the primary copy of the object is stored on a tape volume. If the SDR for the object is less than 3, the primary copy of the object is stored on an optical disk volume.

The IARS for an optical volume depends on its location. The time required for an operator to mount a shelf-resident optical volume is significantly longer than that for the automatic mounting of a library-resident optical volume. You can use up to four characters to specify a valid value of 0-9999, or leave the field blank. The default is blank and selects fixed DASD. However, an OAM request fails if it tries to use a storage class with a blank IARS value.

Parent topic:[Defining initial access response seconds](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data set access

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SMS allows the system resources manager (SRM) to select a DASD volume from the primary volume list if the IART value is 0 or unspecified. SRM volume selection is ideally suited for batch jobs.

You can allow SMS, instead of SRM, to select a DASD volume by assigning a storage class with a nonzero IART value. In this case, none of the eligible volumes is placed on the primary list; all of the eligible volumes are placed only on the secondary or tertiary list. SMS uses the randomizing technique, for example, so as not to always select the volume that has the most free space.

See [SMS volume selection for data set allocation](#) for further information on SMS volume selection.

Parent topic:[Defining initial access response seconds](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining sustained data rate

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The Sustained Data Rate (SDR) field applies to physical sequential data sets, VSAM data sets, and objects.

[Physical sequential and VSAM data sets](#)

[Object access](#)

Parent topic:[Defining performance objectives](#)

|

©Copyright IBM Corporation 1990, 2014

Physical sequential and VSAM data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The Sustained Data Rate (SDR) has an effect only for extended format data sets. Striping allows you to spread data across different DASD volumes and controllers. The number of stripes is the number of volumes on which the data set is initially allocated. Striped data sets must be system-managed and must be in an extended format.

When no volumes that use striping are available, the data set is allocated as nonstriped with EXT=P specified in the data class; the allocation fails if EXT=R is specified in the data class.

Physical sequential (PS) data sets cannot be extended if any of the stripes cannot be extended. For VSAM data sets, each stripe can be extended to an available candidate volume if extensions fail on the current volume.

If you do not specify Guaranteed Space=Y in the storage class, the following is true: The system uses the SDR field in conjunction with the assumed device transfer rate to derive the number of stripes that a striped data set can have. The SDR field specifies the target throughput rate. Valid values for this attribute are blank or from 0 to 999. The maximum number of stripes for PS data sets is 59. For VSAM data sets, the maximum number of stripes is 16. For nonguaranteed space requests, if the SDR field is blank or 0, the target number of stripes is 1. If the value in the SDR field is greater than 1, it is divided by 4 for 3390s, or by 3 for 3380s to determine the stripe count. For example, for an SDR value of 24, a storage group of 3380s would have a stripe count of 8, and a storage group of 3390s would have a stripe count of 6. The volume must be able to satisfy the requested primary space. Depending on the number of eligible volumes in the candidate storage groups, the actual number of stripes allocated to a data set can be less than the derived one. As the stripe count is decreased, the primary space for each stripe increases, except for the guaranteed space requests.

If you specify Guaranteed Space = Y in the storage class, and for further information about guaranteed and nonguaranteed space allocation, see [Rules for striping volume selection](#).

See z/OS DFSMS Using Data Sets for more information on using SMS to assign storage classes for striped data sets.

Parent topic:[Defining sustained data rate](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Object access

z/OS DFSMSdfp Storage Administration
SC23-6860-01

OAM uses the SDR field to determine whether the primary copy of the object should be placed on optical or tape media. An SDR value of 3 or greater places the data on tape media, and a value of 2 or less places the data on optical media. OAM uses the OSL field to determine (as appropriate): what disk sublevel the storage class is associated with, or what tape sublevel the storage class is associated with. For both disk and tape sublevels, the valid values are 1 or 2; and the default value is 1.

See *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support* for further information.

Parent topic:[Defining sustained data rate](#)



©Copyright IBM Corporation 1990, 2014

Defining availability

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The Availability field is used to specify whether data set access should continue in the event of a single device failure. Storage classes with a blank Availability field default to NOPREF. **CONTINUOUS**Specify an availability of CONTINUOUS if you do not want a device failure to affect processing. Only duplexed and RAID volumes are eligible for this setting. If CONTINUOUS availability is specified, data is placed on a device that can guarantee that it can still access the data in the event of a single device failure. This option can be met by A dual copy volumeAn array DASD **PREFERRED**Array DASD volumes are preferred over nonduplexed volumes. Dual Copy volumes are not candidates for selection.**STANDARD**If data sets do not require such a high level of availability, specify STANDARD availability, which represents normal storage needs. Specify an availability of STANDARD to cause processing of a data set to stop after a device failure. Simplex volumes are preferred over array DASD. SMS selects only volumes that are *not* dual copy. This attribute does not apply to objects. Array DASD are acceptable candidates for both STANDARD and CONTINUOUS availability requests.

NOPREFSimplex and array DASD are equally considered for volume selection. NOPREF is the default. Dual copy volumes are not candidates for selection.

Parent topic:[Defining storage class attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining accessibility

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The storage class accessibility attribute defines the function of the hardware supporting the point-in-time copy, using either concurrent copy, virtual concurrent copy, or FlashCopy®. The point-in-time copy allows database management systems (DBMSs) to take what appears to be an instantaneous copy of data or a "fast" point-in-time copy. The copy can be for backup purposes (generally to tape) or for copying a database from one set of DASD volumes to another. The accessibility attribute allows you to direct allocation of new data sets to DASDs connected to an IBM® 3990 Storage Control unit with cache that supports the concurrent copy function, IBM RAMAC Virtual Array (RVA) devices with the SnapShot copy support, DFSMSdss with the virtual concurrent copy, and the IBM ESS with the FlashCopy service.

The IBM Enterprise Storage Server® (ESS) FlashCopy service provides the appearance of instantaneous replication of a range of track images. You can invoke this service with DFSMSdss and use it to create copies for disaster recoveries, business intelligence applications, data in a test environment, and instantaneous checkpoints.

See z/OS DFSMS Advanced Copy Services for further information on the ESS FlashCopy service.

When you specify accessibility attributes, you identify whether you want SMS to use *versioning* or *backup* devices for either concurrent copy or virtual concurrent copy.

Versioning DeviceCan create a fast point-in-time version of a data set, which is then available for application testing, reporting, or backup operations. While the version is being made, the data set is unavailable for normal application processing for a minimal period of time. Versioning is done using the SnapShot feature of the RAMAC Virtual Array or the ESS FlashCopy service.**Backup Device**Can create a fast point-in-time backup copy of a data set. While the backup copy is being made, the data set is unavailable for normal application processing for a minimal period of time. Two methods are supported: **Method 1**Establish a concurrent copy session with the 3990 DASD controller and make the backup copy.**Method 2**Use virtual concurrent copy. DFSMSdss uses the SnapShot feature of the RAMAC Virtual Array or the ESS FlashCopy service to create a point-in-time copy and then does I/O to create the backup to whatever target device you specified. You can use a combination of values for the accessibility attribute and its subparameters, Versioning and Backup, to request the point-in-time copy. In the Accessibility field, you specify whether allocation to a point-in-time copy-capable volume is required (CONTINUOUS), preferred (CONTINUOUS PREFERRED), or discouraged (STANDARD). You then specify values for the Versioning and Backup subparameters to select which devices you want used for the copy.

The following defines your allocation request to a point-in-time copy-capable volume: **CONTINUOUS (C)**Only point-in-time copy volumes are selected. **CONTINUOUS PREFERRED (P)**Point-in-time copy volumes are preferred over non-point-in-time copy volumes. **STANDARD (S)**Non-point-in-time copy volumes are preferred over point-in-time copy volumes. **NOPREF (N)**Point-in-time copy capability is ignored during volume selection. This is the default.

[Table 1](#) identifies the values you specify based on your specific accessibility requirements:

Table 1. Combinations for Requesting Point-in-Time Copy Devices

You can also use the data set alter application to alter a storage class so that data sets can be allocated on point-in-time copy-capable volumes.

Parent topic:[Defining storage class attributes](#)

©Copyright IBM Corporation 1990, 2014

Defining the guaranteed space attribute

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can allocate space for single volume and multivolume data sets by specifying a storage class with the Guaranteed Space attribute. SMS fails the request when space is insufficient. The Guaranteed Space attribute does not apply to objects, but it does apply to both VSAM and non-VSAM-managed data sets.

For a multivolume system-managed data set, the system preallocates primary space on all the volumes. The first volume becomes the primary volume. The remaining volumes become candidate volumes with preallocated space. The system operates differently depending on what the user codes for DISP.

When the existing allocated space on the current volume becomes full with DISP=NEW or MOD, the system attempts to create secondary extents on the current volume. If not enough space is left on the current volume, the system uses the preallocated primary extent on the next volume. The system converts this next volume from a candidate volume to a primary volume.

With DISP=MOD, you can specify different primary space amounts to be allocated to each volume. To accomplish this you can specify multiple DD statements using the same data set name, each with different values for primary space. Refer to [Preallocating space for multivolume data sets](#) for examples.

When the allocated space (possibly including secondary amounts) on the current volume becomes full with DISP=OLD or SHR, the system switches to the next volume. The system allocates new space only on the last volume. The system converts each candidate volume to a primary volume.

For VSAM key-sequenced data sets with key ranges specified, KEYRANGES will be ignored if it is specified for the DEFINE CLUSTER command of IDCAMS.

For information about key ranges, see the description of the DEFINE CLUSTER command in z/OS DFSMS Access Method Services Commands.

[Preallocating space for multivolume data sets](#)

[Honoring specific volume requests](#)

Parent topic:[Defining storage class attributes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Preallocating space for multivolume data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This topic shows scenarios for preallocating space for multivolume data sets.

Allocating a multivolume data set on SMS-selected volumes

The following JCL allocates a multivolume data set on SMS-selected volumes. The storage class DBLOG must have Guaranteed Space=Y. This example allocates 100 MB on each of five volumes. When all of the allocated space is used for the data set on one volume, the secondary space is allocated as required in extents on that volume. For example:

```
//DD1 DD DSN=ENG.MULTFILE,DISP=(,KEEP),STORCLAS=DBLOG,  
//           SPACE=(1,(100,25)),AVGREC=M,  
//           UNIT=(3380,5)
```

Note: After 100 MB is used on the first volume, 25 MB extents of secondary space is allocated on it until the extent limit is reached or the volume is full. If more space is needed, 100 MB of primary space is used on the second volume. Then, more secondary space is allocated on that volume. The same process is repeated on each volume, as shown in [Figure 1](#).

Figure 1. Allocation of Primary and Secondary Space for Multivolume Data Sets

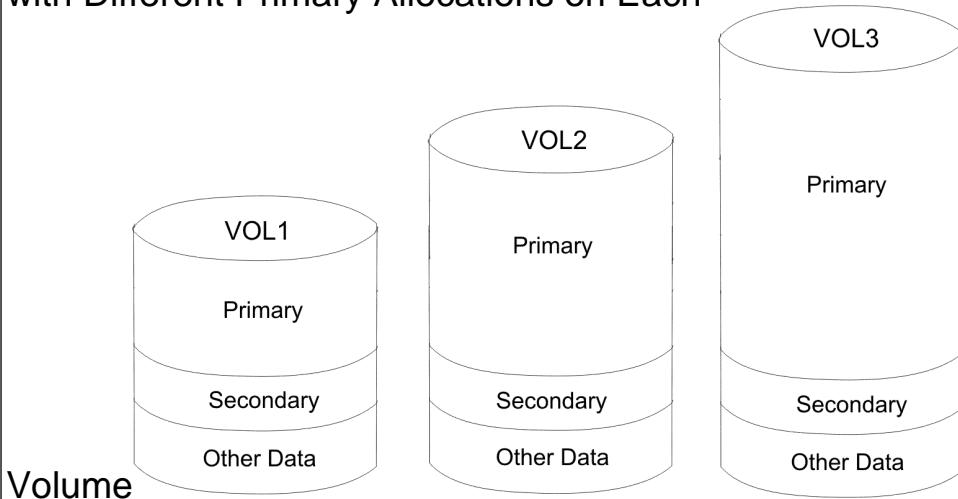
Allocating a multivolume data set with different primary allocation on each volume

The following JCL allocates a multivolume data set on SMS-selected volumes with different primary allocation on each volume. The storage class DBLOG must have Guaranteed Space=Y. This example preallocates 10 cylinders on the first volume (VOL1), 20 cylinders on the second volume (VOL2) and 30 cylinders on the third volume (VOL3). When all of the primary space is used on a volume, the secondary space is allocated as required in extents on that volume. For example:

```
//DD1 DD DSN=DATASET1,DISP=(MOD,CATLG),SPACE=(CYL,(10,5)),VOL=SER=VOL1  
//DD2 DD DSN=DATASET1,DISP=(MOD,CATLG),SPACE=(CYL,(20,5)),VOL=SER=VOL2  
//DD3 DD DSN=DATASET1,DISP=(MOD,CATLG),SPACE=(CYL,(30,5)),VOL=SER=VOL3
```

Note: After the 10 preallocated cylinders are used on VOL1, 5 cylinder extents of secondary space are allocated on this volume until the extent limit is reached or the volume is full. If more space is needed, the 20 cylinders of preallocated primary space is used on VOL2. Secondary space is allocated on this volume, as required, until the extent limit is reached or the volume is full. If more space is needed, the 30 cylinders of preallocated primary space is used on VOL3. Secondary space is allocated on this volume, as required, until the extent limit is reached or the volume is full.

Figure 2. Allocation of Primary and Secondary Space for Multivolume Data Sets with Different Primary Allocations on Each



Parent topic:[Defining the guaranteed space attribute](#)

©Copyright IBM Corporation 1990, 2014

Honoring specific volume requests

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you specify Guaranteed Space=N, SMS chooses volumes for allocation, ignoring any VOL=SER JCL statements. Primary space on the first volume is preallocated. NO is the default.

Specifying volume serials with the Guaranteed Space attribute of the storage class is strongly discouraged. If used, the following considerations must apply: Ensure that the user is authorized to the storage class with the Guaranteed Space attribute. Write a storage group ACS routine that assigns a storage group containing the volumes explicitly specified by the user. Ensure that all volumes specified by the user belong to the same storage group by directing an allocation with a Guaranteed Space attribute to all the storage groups in the installation. Ensure that the requested space is available on the volume because there is no capability in SMS to allow specific volume requests except with the Guaranteed Space attribute. Ensure that the availability and accessibility specifications in the storage class can be met by the specified volumes.

With the IBM® Enterprise Storage Server® (ESS), the guaranteed space attribute of a storage class with specific volume serials is no longer required for data sets other than those that need to be separated (for example, DB2® online logs and BSDS) or that must reside on specific volumes because of their naming conventions (for example, VSAM RLS control data sets). The ESS storage controllers use the RAID architecture that enables multiple logical volumes to be mapped on a single physical RAID group. If required, you can still separate data sets on a physical controller boundary for the purpose of availability.

The ESS capabilities of multiple allegiance and parallel access volumes (PAV), along with its bandwidth and caching algorithms, make it unnecessary to separate data sets for the purpose of performance. Traditionally, IBM storage subsystems allow only one channel program to be active on a disk volume at a time. This means that once the subsystem accepts an I/O request for a particular unit address, this unit address appears “busy” to subsequent I/O requests. This ensures that additional requesting channel programs cannot alter data that is already being accessed. By contrast, the ESS is capable of multiple allegiance, or concurrent execution of multiple requests from multiple hosts. That is, the ESS can queue and concurrently execute multiple requests for the same unit address from multiple hosts, provided that no extent conflict occurs.

The ESS can also execute multiple concurrent accesses to a single volume from a single host or PAV. To access a volume concurrently, you must associate multiple device numbers with a single volume. The ESS provides this capability by allowing you to define a PAV-base address and one or more PAV-alias addresses. It allows up to 255 aliases per logical volume. Therefore, you no longer have to separate data sets from each other for performance reasons.

Parent topic:[Defining the guaranteed space attribute](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining guaranteed synchronous write

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the Guaranteed Synchronous Write attribute to indicate whether your system should return from a BSAM CHECK (or WAIT) issued for a WRITE against a PDSE member before (unsynchronized) or after (synchronized) the data has actually been written to DASD. This attribute does not apply to objects. Specify a Y for synchronized write or an N for no synchronization.

Parent topic:[Defining storage class attributes](#)



©Copyright IBM Corporation 1990, 2014

Defining use of the coupling facility for VSAM record-level sharing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can associate a storage class with a CF cache set that is defined in the base configuration, thereby making any data set associated with the storage class eligible for VSAM record-level sharing. You can also assign a weight value to the data, so as to indicate its relative importance. You should be familiar with the information in the following topics: [Administering VSAM record-level sharing](#)[SMS volume selection for data set allocation](#)[Using the multi-tiered storage group function](#) Specify values for the following attributes: **Multi-Tiered SG**Specify Y if you want SMS to prefer the storage group sequence order as specified in the ACS storage group selection routines. ISMF primes the field with the default N, No.**Parallel Access Volume Capability**Specify R for required, P for preferred, S for standard, and N for no preference. The default is N.**CF Cache Set Name**Is the name of a CF cache set that is defined in the base configuration. When you specify a cache set name, any data set associated with this storage class becomes eligible for record-level sharing using the CF. CACHE SET NAME maps the storage class to a CF cache set defined in the SMS base configuration, for which CF cache structures have been defined.

In a JES3 environment, be careful to define cache set names only in those SMS storage classes that are used by data sets opened for VSAM RLS processing. When you define a cache set name in a storage class, any job accessing a data set associated with that storage class is scheduled on a VSAM RLS-capable system. If all storage classes have cache set names defined for them, then all jobs accessing SMS-managed data sets are scheduled to VSAM RLS-capable systems. This could cause a workload imbalance between those systems and down-level systems.

CF Direct Weight or the CF Sequential Weight Specify a weight attribute indicating the data's relative importance. Use the CF Direct Weight field for direct data; use the CF Sequential Weight field for sequential data. The default is a weight value of 6. Restriction: DFSMS supports only the default value. All data is assigned a weight value of 6 regardless of the value you specify.

Parent topic:[Defining storage class attributes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining when to disconnect the sphere

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the Disconnect Sphere at CLOSE attribute to indicate whether the sphere should be disconnected upon closing the data set or stay connected for a short period of time.

Type Y to disconnect the sphere when the data set is closed. This may be beneficial in an environment with constrained buffer pool space when the same data sets are not repeatedly closed and re-opened. Type N to keep the sphere connected for a short time after the data set is closed. N is the default.

Parent topic:[Defining storage class attributes](#)

|

©Copyright IBM Corporation 1990, 2014

Assigning storage classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

With the exception of tape data sets, a data set is SMS-managed if it is assigned a storage class. You can assign storage classes either through the storage class ACS routine or by explicit specification. If you do not specify an explicit storage class when you store an object, the object is assigned the storage class that is defined in the collection to which the object belongs. The default storage class for an object collection is assigned by the ACS routine when the first object is stored in that collection. If the storage class ACS routine determines a storage class, it takes precedence over one that is explicitly specified by any of the following:

- JCL DD statements
- TSO/E ALLOCATE command
- DFSMSdss COPY and RESTORE commands
- Access method services ALLOCATE, DEFINE, and IMPORT commands
- Dynamic allocation requests, such as with ISPF/PDF data set allocation panels
- OSREQ STORE and CHANGE requests

ACS routines are required for Distributed FileManager/MVS-created data sets to ensure that they are SMS-managed. These data sets must be SMS-managed. If a remote application attempts to create a local data set in non-SMS-managed storage, Distributed FileManager/MVS refuses to honor the request because it only creates SMS-managed data sets. Distributed FileManager/MVS does, however, support the access of non-SMS-managed data sets.

The syntax for specifying a storage class on a JCL statement is `STORCLAS=storage-class-name`

The syntax for specifying a storage class on a TSO/E command is: `STORCLAS(storage-class-name)`

The syntax for specifying a storage class on an access method services command is: `STORCLAS(-)`

The syntax for specifying a storage class on an OSREQ STORE or CHANGE

macro is: `STORCLAS=storage-class-area` or `STORCLAS=(storage-class-area-pointer)`

For information on determining storage classes with ACS routines, see [Defining ACS routines](#).

Parent topic:[Defining storage classes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining additional storage classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can copy existing storage classes and modify them to create new storage classes by using the COPY line operator, which is explained in [Copying SMS components](#).

Parent topic:[Defining storage classes](#)



©Copyright IBM Corporation 1990, 2014

SMS volume selection for data set allocation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When a data set is created, SMS follows a sequence of steps to place it on a volume. This section describes how SMS selects a volume based on the requirements of a data set. It explains conventional volume selection, striping volume selection and multi-tiered volume selection. Finally, it discusses why volume selection might fail. **Restriction:** SMS does not check or verify DASDVOL authorization for data set allocations on SMS-managed volumes. By adding a volume to SMS you are removing it from any active DASDVOL authorizations because DASDVOL access is not verified on SMS-managed volumes. See *z/OS Security Server RACF Security Administrator's Guide* for further information on DASDVOL authority.

SMS attempts to spread initial allocations evenly across similar volumes and storage groups provided by the ACS storage group selection routine unless you specify Multi-Tiered SG (Y) in the storage class. In that case, SMS performs an allocation using storage groups in the order in which they are specified in the ACS storage group selection routine. For more information, see [Using the multi-tiered storage group function](#).

[Volume selection preference attributes](#)

[Conventional volume selection](#)

[Striping volume selection](#)

[Tuning considerations](#)

[Cluster and storage facility image considerations](#)

[Spreading allocations across multiple volumes](#)

[Space constraint relief](#)

[Using the multi-tiered storage group function](#)

[Using the parallel access volume option](#)

[Possible reasons for volume selection failure](#)

Parent topic:[Defining storage classes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Volume selection preference attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Table 1 shows some of the preference attributes used in volume selection for allocation of a data set.

Table 1. Volume Selection Preference Attributes

The volume selection preference attributes in **Table 1** are assigned a weight in the order shown (highest to lowest). SMS selects the volume with the highest cumulative preference weight.

Parent topic:[SMS volume selection for data set allocation](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Conventional volume selection

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Conventional volume selection is used for all nonstriped data sets, as well as for data sets allocated with an SDR of zero or blank.

Conventional volume selection uses the criteria specified in the storage class and the data class as preference attributes for volume selection. SMS uses the preference attributes to: Classify the volume as primary, secondary or tertiary. Only volumes that meet all of the requested attributes can be classified as primary volumes. Assign a selection weight to the volume.

SMS begins by classifying each volume in a storage group as primary, secondary, tertiary or rejected, as follows: **Primary** All the volumes in all the specified storage groups are candidates for the first, or primary, list. Exception: When Multi-Tiered SG (Y) is specified in the storage class, only the volumes in the first storage group assigned by the ACS routine are eligible for the primary list. Also, volumes in Overflow and Extend storage groups are not eligible for the primary list.

The primary list consists of online volumes that meet all the requested preference attributes, are below their threshold, and whose volume status and storage group status are enabled. All volumes on this list are considered equally qualified to satisfy the data set allocation request. Volume selection starts from this list.

SMS chooses from the candidates on this list, preferring volumes that are not already allocated to a job or subsystem and that have the least I/O delay, as determined by the system resource manager (SRM). If no devices in the storage groups meet all of the primary list criteria, no volumes will appear on the primary list. It is possible that, even though there are devices on this primary list, the data set cannot be successfully allocated to any of these devices. That is, there is not enough space available when the actual allocation request is made.

For example, if you specify a 25 MSR in your storage class, volumes that are close to an MSR of 25 would be POOL1 volumes. POOL1 volumes are those volumes that meet your MSR requirement by a certain percentage. Each MSR percentage represents a range of MSRs. All volumes that fall within an MSR percentage are considered equal in performance. Volumes that provide a faster MSR by a larger percentage than POOL1 volumes are considered POOL2 volumes. POOL2 volumes include POOL1 volumes plus the volumes in the next higher MSR percentage. POOL1 volumes are preferred over POOL2 volumes.

Primary volumes include online POOL1 and POOL2 volumes that can meet the selection preference attributes listed in [Table 1](#), in addition to guaranteed space requirement.

Volume selection from the primary list can result in skewing in extreme circumstances. These circumstances include the occasions when a VTOC index is disabled or when a new volume is added to SMS and selected most of the time.

SMS uses the randomizing technique to avoid favoring one volume over others, such as an empty volume, a volume with a disabled index, or a volume under the allocation threshold. You can force SMS to always use the randomizing technique by specifying a nonzero IART value. This causes all fixed DASD volumes to be placed on the secondary list, which uses randomization between similar volumes.

Secondary Volumes that do not meet all the criteria for the primary volume list are placed on the secondary list. If there are no primary volumes, SMS selects from the secondary volumes. If a data set cannot be allocated on a primary volume, SMS continues to evaluate volumes on the secondary list until allocation is successful. SMS evaluates volumes in the secondary list based upon volume characteristics, performance, available space and system accessibility. See [Table 1](#) for more information about volume selection preferences.

Tertiary Volumes are marked for the tertiary list if the number of volumes in the storage group is less than the number of volumes requested. If there are no secondary volumes available, SMS selects from the tertiary candidates. When a storage group does not contain enough volumes to satisfy the volume count, all

volumes in the storage group are flagged as tertiary. Tertiary volumes are only selected when there are no primary or secondary volumes and the request is for a non-VSAM, non-GUARANTEED SPACE request. The concept of tertiary volumes does not apply to VSAM data sets. In other words, for all VSAM non-GUARANTEED SPACE requests, the volume count does not play a role in determining which storage group is selected.

RejectedVolumes that do not meet the *required* specifications (ACCESSIBILITY = CONTINUOUS, AVAILABILITY = STANDARD or CONTINUOUS, ENABLED or QUIESCED, ONLINE...) are marked rejected and are not candidates for selection. To improve performance, SMS limits the number of eligible volumes to those most likely to satisfy the request:

For non-best-fit allocation where the primary space must be gotten on one volume, SMS will exclude a volume from selection if the requested primary space exceeds its total capacity. For best-fit allocations where the primary space is allowed to be spread out on multiple volumes, DADSM will be called until the primary space requested has been gotten or the maximum number of volumes has been used and requested space has not yet been completely gotten. In this latter case, SMS will reject a storage group after 59 volumes are rejected by DADSM for insufficient space. The remaining volumes in the storage group are not considered for further selection since they are less preferred and deemed to have less chance to fulfill the allocation request. For non-best-fit allocations using 'fast' volume selection, SMS will perform volume selection from the prioritized list until 100 volumes have been rejected by DADSM for insufficient space. When that occurs, SMS will exclude, based on the volume statistics in the SMS configuration, all volumes that do not have sufficient free space. This 'fast' volume selection approach can greatly reduce the number of candidate volumes, and thus the number of retries. You can activate 'fast' volume selection by using the FAST_VOLSEL(ON) parameter in IGDSMSxx or SETSMS FAST_VOLSEL(ON) command. After the system selects the primary space allocation volume, that volume's associated storage group is used to select any remaining volumes requested for the data set. If you specify an extend storage group, the data may be extended to the specified extend storage group. For information on how to specify an extend storage group using the Extend SG Name parameter, see [Defining a pool storage group](#).

[Reasons a volume is not placed on the primary list](#)

Parent topic:[SMS volume selection for data set allocation](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Reasons a volume is not placed on the primary list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A volume is not placed on the primary list if it does not meet all of the criteria listed in [Table 1](#), or for one of the following reasons: The volume is rejected. See [Possible reasons for volume selection failure](#) for more information. The VTOC index is broken or disabled, resulting in SMS not being updated with space statistics from the Common VTOC Access Facility (CVAF). This can also happen if OEM products bypass CVAF, the component that notifies SMS of space changes. SMS not being updated with space statistics can also result in a volume being overused. The volume was placed on the tertiary list because the number of volumes in the storage group was less than the number of requested volumes. For example, this happens when a tape request of five volumes causes a tape mount management buffer storage group containing only two volumes to be marked tertiary, while an overflow storage group or a quiesced storage group containing the requested five volumes is marked secondary. The overflow volumes and quiesced volumes on the secondary list are preferred over the tape mount management volumes on the tertiary list. SMS will attempt to allocate space using the overflow volumes first, then the quiesced volumes, and lastly the tape mount management buffer volumes. The controller was IMLED while online to MVS. This can result in the MVS device control blocks not reflecting the current state of the volume. The device or devices should be varied offline and back online to update the MVS control block status. RAID devices are eliminated from the primary list when STANDARD is specified. Use NOPREF, the AVAILABILITY default, to allow both SIMPLEX and RAID devices to be placed on the primary list.

For more information to help you determine why a particular volume is selected or not selected, see [Table 1](#).

Parent topic:[Conventional volume selection](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Striping volume selection

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Striping volume selection is entered only in the following situations: During initial allocation: If the data class specifies an extended format as either 'required' or 'preferred' and the SDR value in the storage class is nonzero. If the SDR value is zero, a nonstriped data set in extended format is allocated, and conventional volume selection is used. Tip: A nonstriped data set in extended format may refer to a *single-striped data set*.

During restore/recall processing: If the data set was a multistripe data set when it was migrated or backed-up. If the data set was single-striped when migrated or backed-up, it follows the conventional volume selection path.

Striping volume selection is very similar to conventional volume selection. Volumes that are eligible for selection are classified as primary and secondary, and assigned a volume preference weight, based on preference attributes. See [Table 1](#) for more information on preference attributes.

Volumes are classified as follows: **Primary**For each controller, SMS randomly assigns a single volume that meets all of the requested preference attributes as the primary volume.**Secondary**For each controller, SMS classifies all eligible volumes other than the primary volume as secondary volumes.

SMS calculates the average preference weight of each storage group using the preference weights of the volumes that will be selected if the storage group is selected for allocation. Then, SMS selects the storage group that contains at least as many primary volumes as the stripe count and has the highest average weight. If there are no storage groups that meet these criteria, the storage group with the largest number of primary volumes is selected. If multiple storage groups have the largest number of primary volumes, the one with the highest average weight is selected. If there are still multiple storage groups that meet the selection criteria, SMS selects one at random.

After selecting a storage group, SMS selects volumes by their preference weight. Primary volumes are preferred over secondary volumes as they have a higher preference weight. Secondary volumes are selected when there are an insufficient number of primary volumes. If there are multiple volumes with the same preference weight, SMS selects one of the volumes at random.

Volumes that meet the requested MSR are preferred over volumes that do not meet the requested performance. A volume is considered to meet the requested performance if the volume's performance is within a predetermined range of the requested MSR.

The throughput of striped data sets is gated by the slowest device if the striped set includes devices of varying data delivery capabilities.

When allocating striped data sets, SMS prefers allocating stripes across extent pools. If this is not feasible, SMS allocates stripes across logical controllers (LCUs). When extending a stripe of a striped data set to a new volume, SMS prefers volumes in the extent pools that are not being used by other active stripes.

[Rules for striping volume selection](#)

[Requirements for data set striping](#)

Parent topic:[SMS volume selection for data set allocation](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Rules for striping volume selection

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For both guaranteed and nonguaranteed space allocations, the following key rules apply for striping volume selection: Storage groups containing mixed device types are not considered.Target volumes An SDR of 0 results in a target stripe count of 1 for both guaranteed and nonguaranteed space requests, and conventional volume selection occurs. A nonzero SDR that is 1 when divided by the device value results in a target stripe count of 1, but striping volume selection is used instead of conventional volume selection.For nonguaranteed space, the number of target volumes is computed by dividing the SDR that is specified in the storage class by a value of 3 for 3380 devices, by a value of 4 for 3390 devices, and rounding up the result, if required. For example, an SDR of 18 results in a target stripe count of 6 on a 3380 device and a target stripe count of 5 (after rounding up) on a 3390 device.For guaranteed space, if the SDR value is 1 or greater, it is ignored; in which case, the target number of stripes is the greater of either the volume count that is specified or the number of specified volume serial numbers. All specified volumes must be in the same SMS storage group for a guaranteed space request. SMS assumes that the amount of space that the user wants is the target number of stripes times the specified primary space. All temporary data sets with a volume count greater than one are allocated as nonstriped.The volume must be able to satisfy the primary space requested by the number of stripes.The maximum number of stripes (volumes) for VSAM data sets is 16. The maximum number of stripes for physical sequential (PS) data sets is 59.The maximum number of extents is five for each space allocation.The maximum number of extents per stripe is 123. For VSAM data sets, the maximum number of extents: Per volume is 123 Per component is 7257The minimum allocation is one track per stripe.The maximum allocation can exceed the 64K track limit.All stripes must be able to satisfy the secondary space allocation (divided by the number of stripes) during extend processing or the allocation fails. Secondary space amount is divided by the number of stripes for both guaranteed and nonguaranteed space requests. OPEN, CLOSE, and EOV perform this calculation.Non-VSAM multistriped data sets cannot be extended to additional volumes. Striped VSAM data sets can be extended to additional volumes.Volume fragmentation information is not available to SMS at volume selection time. An allocation failure by DADSM because of fragmentation results in striping volume reselection.Primary space For nonguaranteed space, the volume must be able to satisfy the primary space that is requested divided by the number of stripes. For example, if primary space is 15 MB and the number of stripes is three, the volume must be able to satisfy an allocation of 5 MB.For guaranteed space, for requests that contain specified volume serial numbers, each stripe must be able to satisfy the requested primary space (15 MB in above example).Secondary space amount is divided by the number of stripes and rounded up for each volume.

The following is true for guaranteed space allocations: If you explicitly specify volume serial numbers with guaranteed space and the target number of stripes is equal to the number of volume serial numbers that you specify, SMS must allocate the primary space requested on each of these volumes. If this is not possible, the allocation fails.If you do not specify any volume serial numbers, then the target number of stripes is equal to the volume count. SMS tries to select the same number of volumes, but settles for less if this number is unavailable. If fewer stripes are allocated for the non-VSAM data sets, SMS increases the allocation per volume to compensate for the fewer stripes. For VSAM data sets, the primary space requested will be allocated on each stripe regardless of whether the target number of stripes is acquired or not.For VSAM data sets, if the number of guaranteed space volumes exceeds sixteen (which is the maximum number of stripes for a VSAM data component), the number of guaranteed space volumes will be reduced to sixteen with the remaining volumes becoming candidates for secondary space.If

the target number of stripes is higher than the number of volume serial numbers you specify, SMS must select all the specified volumes plus additional ones. These nonspecific volumes are not mandatory and if none are available, SMS allocates the primary quantity on each of the specified volumes. Recommendation: Ensure that you have a sufficient number of volumes behind each controller in the storage group to reduce volume overuse. An example of a problem would be if a storage group contains eight controllers and the average stripe count is four, each controller is selected approximately 50 percent of the time. If one controller contained only two volumes, each of the two volumes is selected for approximately 25 percent of all new striped allocations.

Parent topic:[Striping volume selection](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Requirements for data set striping

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following lists the requirements for data set striping. Volumes must be behind one of the following controllers: Controllers that are ESCON®-attached and support concurrent copy3990-6 controllers3990-3 controllers which are Extended Platform and ESCON-attached3990-3 controllers which have the RAMAC support-level microcode9394 controller9343 controller with cacheIBM® RAMAC Virtual ArrayIBM Enterprise Storage Server®Volumes must be ENABLED or QUIESCED and varied ONLINE.

Parent topic:[Striping volume selection](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Tuning considerations

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Be careful when tuning your storage class ACS routines to match the mixture of volume capabilities assigned by your storage group ACS routines. For example, if you have both cached and noncached 3390-3 in the assigned storage groups and your ACS routine assigned the same storage class (MSR=25) to all data sets, the noncached volumes are preferred over the cached volumes. This results in the noncached volumes being overused and the cache-active volumes are not selected until all the noncached volumes are above high threshold. If an MSR=2 is specified, then the cached volumes are overused.

This also holds true for other attributes. If you always assign ACCESSIBILITY=PREFERRED, then concurrent copy volumes are overused.

Parent topic:[SMS volume selection for data set allocation](#)

|

©Copyright IBM Corporation 1990, 2014

Cluster and storage facility image considerations

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The DFSMSdss data set fast replication function requires that all volumes of a multi-volume data set reside in the same storage facility image (SFI). Data set fast replication is most efficient when both source and target data sets reside in the same cluster. To help you take advantage of data set fast replication, SMS volume selection:When allocating or extending an SMS-managed multi-volume data set that has point-in-time copy volumes requested, prefers candidate volumes that are in the same cluster, and, if that cannot be honored, prefers candidate volumes that are in the same SFI. Point-in-time copy volumes are requested when the ACCESSIBILITY parameter in the storage class is set to CONTINUOUS or CONTINUOUS PREFERRED. When allocating the target data set for data set fast replication, prefers volumes in the same cluster as the source data set, and, if that cannot be honored, prefers volumes in the same SFI as the source data set. To make efficient use of the fast replication function, ensure that enough volumes are in one SFI to accommodate the likely increase in allocations to volumes in these storage groups and SFIs. For optimum efficiency, ensure that enough volumes are in one cluster to accommodate the likely increase in allocations to volumes in these storage groups and clusters.

The preference for volumes that are in the same cluster or SFI applies to all multi-volume allocations with ACCESSIBILITY = CONTINUOUS or CONTINUOUS PREFERRED. However, there are a few situations where SMS volume selection will not take the cluster or SFI attribute into consideration. These situations are: The volumes specified by the user for a guaranteed space request are not in the same SFI. No clusters or SFIs have a sufficient number of volumes to meet the number of volumes required for allocation. No clusters or SFIs have a sufficient number of unique controllers to meet the stripe count for an striping allocation. During extend processing, the volumes that the data set resides on are not in the same SFI. Space Constraint Relief (SCR) processing is in effect.

Parent topic:[SMS volume selection for data set allocation](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Spreading allocations across multiple volumes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You might want to separate allocations and spread them across multiple volumes to reduce I/O contention and improve performance. After analyzing your data, you might find that certain data may not need some of the advanced storage features in your environment.

If you want to spread conventional allocations across volumes of different features, use one or all of the following suggested parameter settings in the assigned storage and data classes: Storage Class Parameters MSR = blankBIAS = blankACCESSIBILITY = NOPREFAVAILABILITY = NOPREFGUARANTEED SPACE = NData Class Parameter if EXT = blank

Parent topic:[SMS volume selection for data set allocation](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Space constraint relief

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If SMS cannot allocate the data set to any of the volumes, SMS will perform space constraint relief (if specified in the data class) and repeat the selection process. For more information about space constraint relief, see [Specifying attributes to handle space constraints during allocation](#).

Parent topic:[SMS volume selection for data set allocation](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using the multi-tiered storage group function

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Be familiar with conventional volume selection for allocation, which is described in [SMS volume selection for data set allocation](#).

You can specify Multi-Tiered SG (Y) in the storage class to enable the multi-tiered storage group function. When you specify this function, SMS allocates to enabled pool storage groups in the order they are specified in the ACS storage group selection routines.

Example: Multi-Tiered SG (Y) is specified in the storage class. An ACS storage group routine assigns the following three similar storage groups:

`SET &STORGRP = 'SG1',
'SG2', 'SG3'`

Result: SMS selects the enabled volumes that are below the high threshold in SG1 before selecting volumes in the SG2 storage groups that are listed second, third, and so forth, can also be used depending on the status of the underlying volumes when: All SG1 enabled volumes exceed high threshold, then SMS selects the enabled volumes in SG2. All SG1 and SG2 enabled volumes exceed the high threshold, SMS selects the enabled volumes in SG3. All enabled volumes exceed the high threshold, SMS selects the quiesced volumes in the same storage group order. Only volumes that reside in the first storage group assigned by the ACS routines (SG1 in the example) are eligible for the primary list.

Parent topic: [SMS volume selection for data set allocation](#)



©Copyright IBM Corporation 1990, 2014

Using the parallel access volume option

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Be familiar with conventional volume selection for allocation, which is described in [SMS volume selection for data set allocation](#).

If the Enterprise Storage Server® (ESS) PAV option is enabled, you can use the DFSMS storage class parallel access volume (PAV) option to ensure that data sets that require high performance are only allocated to volumes on which the ESS PAV option is enabled.

The DFSMS PAV capability option includes the following settings with these results:

RequiredOnly volumes with the PAV feature enabled are

selected.**Preferred**Volumes with the PAV feature enabled are eligible to be primary volumes. Volumes without the PAV feature enabled are only eligible to be secondary volumes.

StandardVolumes without the PAV feature enabled are preferred over volumes with the PAV feature enabled and are eligible to be primary volumes. Volumes with the PAV feature enabled are only eligible to be secondary volumes.

NopreferenceWhether the PAV feature is enabled or not for a volume is ignored and has no effect on the volume selection process. This is the default value for this option.

[Planning to use the PAV option in the storage class](#)

[Implementing the PAV option enhancement](#)

[Modifying storage classes to enable the PAV option](#)

[Defining new storage classes with the PAV option enabled](#)

[Displaying information about the PAV capability of a volume](#)

[Diagnosing problems with the PAV option](#)

Parent topic:[SMS volume selection for data set allocation](#)



Planning to use the PAV option in the storage class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: All storage classes defined by your installation are automatically assigned the default PAV value of Nopreference, which means that all PAV and non-PAV volumes are treated equally. To force volume selection to differentiate between them, you need to modify the PAV option in the storage class.

The following list comprises the planning steps that you must complete before you can successfully implement the PAV option in the storage class:

Verify that ESS devices have the PAV capability enabled using the DEVSERV command. Example: DS QPAVS, *devnum*

For information about using the DEVSERV command, see z/OS MVS System Commands.

Determine which data sets require high performance. You could assign a PAV value of Required or Preferred to those data sets. Data sets that require high performance could include frequently accessed data sets and critical data sets. For example, if certain jobs are taking too long because data sets used by these jobs are on non-PAV volumes, you might allocate those data sets on PAV volumes to determine if that improves their performance.

Decide whether to assign the PAV option to existing storage classes. If you do not specify the PAV option in the storage class, the default value is Nopreference.

Decide whether to create any new storage classes that use the PAV option. Now you are ready to implement the SMS PAV option in the storage class.

Parent topic:[Using the parallel access volume option](#)



Implementing the PAV option enhancement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Ensure that your system is running at z/OS® V1R6.

Migration actions: If you do not plan to exploit the PAV volume selection capability, you do not need to perform any migration actions. However, if you plan to exploit the PAV capability, you must make changes to the SMS configuration. Perform the following steps to implement the PAV volume selection capability in the storage class:

Modify the PAV option in the SMS storage classes that are assigned to data sets that require high performance. For the detailed steps, see [Modifying storage classes to enable the PAV option](#).

Define new storage classes with a value other than NOPREFERENCE specified for the PAV option. For the detailed steps, see [Defining new storage classes with the PAV option enabled](#).

Result: Now you have more control over the volume selection process for a SMS-managed data set, depending on whether a volume is PAV enabled. You can update automatic class selection (ACS) routines to use the new or modified storage classes with the PAV capacity enabled.

Restriction: Striped data sets and best-fit data set allocations support the PAV option in a limited way. If PAV capability is required, all volumes that do not have this capability are rejected. However, if you specify a PAV capability of Preferred, Standard, or Nopreference for a striped or best-fit data set, this PAV option is ignored.

Parent topic:[Using the parallel access volume option](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Modifying storage classes to enable the PAV option

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to set up the PAV option in existing storage classes:
Use ISMF to access the Storage Class application.

Select option 4, Alter a Storage Class.

On page 2 of the Storage Class Alter panel, set the Parallel Access Volume Capability value to one of the following values: R (required)P (preferred)S (standard)N (no preference)

Activate the updated configuration using one of the following methods: On the CDS Application Selection panel in ISMF, complete one of the following actions: Select the Activate option, or Enter the ACTIVATE operator command on the command line. Example: ACTIVATE SCDS *dsname*

From the operator console, enter the SETSMS command. Example: SETSMS SCDS *dsname*

Result: The PAV option is enabled for this storage class.

Recommendation: When you modify an existing storage class and activate the configuration, you are implicitly modifying the volume selection process for all data sets to which this storage class is assigned. If you do not want the "modified" storage class to apply to all data sets that it is being assigned to, you will need to change the storage class ACS routines.

Parent topic:[Using the parallel access volume option](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining new storage classes with the PAV option enabled

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define new storage classes with the PAV option enabled:

Use ISMF to access the Storage Class application.

Select option 3, Define a Storage Class.

On page 2 of the Storage Class Define panel, set the Parallel Access Volume Capability value to one of the following values: R (required)P (preferred)S (standard)N (no preference)

Complete the remaining entries on the Storage Class Define option panel.

Update the ACS routines to include the new storage class. For more information, see [Defining ACS routines](#) in the z/OS DFSMSdfp Storage Administration.

Activate the updated configuration using one of the following methods: On the CDS Application Selection panel in ISMF, complete one of the following actions: Select the Activate option, or Enter the ACTIVATE operator command on the command line. Example: ACTIVATE SCDS *dsname*

From the operator console, enter the SETSMS command. Example: SETSMS SCDS *dsname*

Result: The new storage class is activated.

NaviQuest tip: You can specify the PAV option using the NaviQuest sample JCL, ACJBAS1, to define, alter, or display a storage class.

Parent topic:[Using the parallel access volume option](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying information about the PAV capability of a volume

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to display information about the PAV capability of a volume:

Use ISMF to access the Storage Class application.

Select option 2, Display a Storage Class. Go to page 2 of the Storage Class Display panel. The Parallel Access Volume Capability field displays one of the following values: REQUIREDPREFERREDSTANDARDNOPREFERENCE You also can obtain information about the PAV capability of a volume using the following ISMF panels: Storage Class List (Parallel Access Vol field)Storage Class List Print (Parallel Access Vol field)Storage Class Sort Entry (Parallel Access Vol field)Storage Class List View (Parallel Access Vol field)

Parent topic:[Using the parallel access volume option](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Diagnosing problems with the PAV option

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following SMS messages indicate that volumes are rejected or not used because they do not allow the PAV requirement: IGD17268IIGD17269IIGD172791 ISMF issues the following new message if an invalid value is entered for the PAV option on the Storage Class Define/Alter panel: DGTSC084

For more information about these error messages, refer to z/OS MVS System Messages, Vol 4 (CBD-DMO) and z/OS MVS System Messages, Vol 8 (IEF-IGD).

Parent topic:[Using the parallel access volume option](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Possible reasons for volume selection failure

z/OS DFSMSdfp Storage Administration
SC23-6860-01

During the process of volume selection, volumes can be rejected for any one of many different reasons. Possible reasons for rejection include: The volume is not online to MVS. The SMS volume or storage group status is DISNEW, DISALL, or NOTCON. The volume was not initialized as being SMS-managed. The volume resides in a copy pool backup storage group. The volume does not contain enough space to satisfy the primary space requested (this is determined by DADSM). The volume control data sets (VTOC) does not contain enough space to accommodate the format 1 DSCB for the allocation request, as determined by DADSM. The volume was selected and DADSM returned an unsuccessful return/reason code for the allocation request. When DADSM fails to allocate the data set on a volume due to insufficient space on the volume, SMS tries to allocate the data set on a different volume. Other failures in DADSM can result in allocation failures. The volume does not meet the availability requirement specified in the storage class. The volume does not support concurrent copy, SnapShot, or FlashCopy® and the storage class specifies accessibility continuous (Required). The DASD controller does not support extended format and the data class specifies 'If EXT=R.' The volume is mountable DASD when an IART of zero or blank was specified in the storage class. The volume was listed on the provided exclude list or was not listed in the provided include list during the DFSMSdss COPY or RESTORE operations. The volume has an incorrect unit control block (UCB) type. The volume was not specified for a specific guaranteed space request. The volume belongs to a storage group and the storage group does not contain enough volumes to satisfy the volume count of a guaranteed space request. All volumes in the storage group are rejected. The volume is too fragmented to contain the primary space extent. During extend (EOV) processing, volume selection rejects volumes whose device types do not match the device type of the volume that the data set currently resides on. The controller was IMLed while online to MVS. This can result in the MVS device control blocks not reflecting the current state of the volume. The device or devices should be varied offline and back online to update the MVS control block status. SMS can reject the device because of incorrect MVS status. ISMF reports the correct status because ISMF queries the device/controller directly. To not affect I/O performance, SMS uses the status in the MVS device control blocks for all volumes in a storage group. The volume is not assigned to any of the storage groups selected by the storage group ACS routine. This results in the volume not being included on the candidate volume list, which has the same effect as the volume being rejected. Note: During the volume selection process, many volumes can be rejected because they do not possess the right attributes. With few exceptions, volume selection fails only after allocation has been attempted without success on each of the remaining volumes.

Parent topic:SMS volume selection for data set allocation



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining secondary lock tables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can define multiple secondary VSAM RLS lock structures to be associated with a single SMS storage class. A secondary lock structure is identified by an SMS storage class attribute called a lock set. You can define up to 256 lock sets per sysplex. Each lock set can contain a single lock structure name. When an application opens a VSAM data set, RLS processing checks the storage class defined for the data set to determine which lock structure to use. If the storage class specifies a secondary lock structure, RLS processing uses the secondary lock structure to serialize access to records in the data set.

If you do not define secondary lock structures, VSAM RLS uses the primary DFSMS CF lock structure, the IGWLOCK00 lock table, to hold the record locks. However, using a single lock structure forces DFSMS to combine all locking information for all instances of the SMSVSAM address space in the same lock structure. That might cause differing workloads to interfere with each other and affect system and application availability. You can use secondary lock structures to help prevent locking constraints and allow isolation of workloads.

Follow these steps to implement VSAM RLS multiple lock table:

Define the names, sizes and preference locations of the new lock tables in the CFRM policy using the IXM utility.

Activate the new CFRM policy.

Define the names of the new lock tables in the SMS control data set (CDS) using the ISMF Control Data Set function or Navquest. Specify the name of a lock set and one lock structure per lock set (you can use the same lock structure for multiple lock sets). If you do not specify the names of the new lock sets, RLS processing uses the primary lock structure, IGWLOCK00, for all record locking.

Use ISMF or Navquest to add the lock set name to each storage class for which you requested multiple lock table. Otherwise RLS processing uses IGWLOCK00 for all record locking.

Validate and activate the SMS configuration to allow the new definitions in the Source Control Data Set (SCDS) to take effect.

Allocate VSAM data sets using the storage classes that contain the new lock set names.

Update ACS routines.
Note:Secondary lock structures are used for record locks only. When secondary lock structures are used, IGWLOCK00 is still required for other types of locks. IGWLOCK00 is required for the initialization of the SMSVSAM address space.The number of lock structures that are concurrently connected varies and depends on the value of MAXCAD specified in the IEASYxx member that is used for system initialization. There might be 10-14 lock structures, including IGWLOCK00, connected at the same time.

For more information, refer to: [Defining the primary CF lock structure](#),[Quiescing or enabling a secondary lock structure](#),[Deleting a VSAM RLS lock structure](#),[Displaying information about a secondary lock structure](#).

Parent topic:[Defining storage classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining data classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SMS simplifies data set allocations by introducing data classes. This topic describes SMS data classes and explains how you can define data classes with the ISMF data class application.

[Understanding data classes](#)

[Planning data classes](#)

[Defining data class attributes](#)

[Assigning data classes](#)

[Defining additional data classes](#)

 | 

©Copyright IBM Corporation 1990, 2014

Understanding data classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A data class is a list of data set allocation attributes and their values. You cannot assign a data class to an object; however, data class may be used for allocation of a scratch tape to be used to write objects. When end users allocate a data set and refer to a data class either explicitly (for example, through JCL) or implicitly (through ACS routines), SMS allocates the data set using the attribute values of its associated data class. For data class attributes, explicit specifications by end users override any parameters derived from the data class ACS routine. If SMS is active, a data class can be assigned to any data set. For non-SMS-managed DASD data sets, the system uses the allocation attribute values of the data class, but it does not save the data class name. For tape data sets, only the expiration and retention values are applied. Specifying the Override Space attribute causes the space attributes from the data class to override the space information specified explicitly in the JCL.

Not all attributes apply to every data set organization. When SMS allocates a data set, it uses only those data class attributes that have meaning for the given data set organization. SMS saves the data class name for each SMS-managed data set. The actual data class definitions reside in the SCDS. If you alter a data class definition, SMS applies the changes to any new data sets that use the data class after you activate the changed configuration. However, SMS does not retroactively apply your changes to previously allocated data sets. To apply your changes to existing data sets, you need to delete and redefine the data sets.

A data class definition contains identification and allocation information. To identify and refer to your data classes, you assign each one a unique name that contains from one to eight alphanumeric characters. Each data class maintains an owner ID that identifies the storage administrator who originally created or last modified the data class. The owner ID can be viewed on the Data Class List panel. Also, each data class contains an optional 120 character description field for describing its contents.

The data class allocation attributes match the keywords that you use to allocate data sets. The attributes contain space-related, access-related, and organizational information that you typically find on JCL DD statements, TSO/E ALLOCATE commands, access method services DEFINE commands, dynamic allocation requests, and ISPF/PDF panels.

Parent topic:[Defining data classes](#)



Planning data classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You should create a data class based on service level agreements. For example, all data sets having a low-level qualifier of LIST, LISTING, OUTLIST, or LINKLIST probably belong to the same data class, because they are typical work data sets having similar allocation characteristics.

Before you actually define any data classes, gather information about the common types of data sets in your installation. You also need to determine if only the data class ACS routine can assign data classes to data sets, if end users can assign data classes to data sets, or if you want a combination of these two policies. If you intend to have only the data class ACS routine assign data classes, you need to develop methods to identify the data in your installation. However, if you allow only the data class ACS routine to assign data classes, you should be aware that users need to override some data class attributes. For example, you should probably not code one data class for each possible amount of space that users need. Finally, you need to identify the space requirements for some commonly used data sets. By gathering useful installation information, you can relieve your end users from specifying all of the allocation attributes on allocation requests. They can use the data class that most closely matches their needs and explicitly change the few attributes that are unique to their data sets.

You can create a data class with no attributes specified and use it to handle system-managed data sets that are allocated (but never opened) without all DCB attributes being specified. These are called "empty data sets." This might be a common situation with batch systems where the DSORG, for example, is specified in the program but not all allocated data sets are used for every run of the batch stream. The empty data sets then occupy space on DASD and cannot be migrated by DFMSHsm because they do not have a specified DSORG. To solve this, check in your data class ACS routine for allocations that do not specify a DSORG or a data class and assign them the blank data class. This causes the DSORG for the data set to default to physical sequential (PS), so that DFMSHsm can migrate the data set.

A data set that has been opened for output and closed without writing anything is called a "null data set."

Parent topic:[Defining data classes](#)



Defining data class attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to define data class attributes.

Select option 4, Data Class, from the ISMF Primary Option Menu for storage administrators. This displays the Data Class Application Selection panel.

Supply values on the Data Class Application Selection panel: **CDS Name**This must be the name of an SCDS. ISMF primes the CDS Name field with the last used SCDS name. The default is 'active'. This represents the currently active configuration, but you cannot define or alter the storage groups for the active configuration.**Data Class Name**This is the name of the data class. ISMF primes the field with the last used name.**Option**Select option 3, Define. This displays the Data Class Define panel.

Supply values on the Data Class Define panel. You can leave any of the pages of the Data Class Define panel at any time without saving the data class by issuing the CANCEL command. Use the DOWN command to display the next page of the panel. For more information, see: [Defining record and space attributes for data class](#)
[Defining volume and data set attributes for data class](#)
[Specifying attributes to handle space constraints during allocation](#)
[Defining VSAM attributes and specifying media types for data class](#)
[Defining the encryption management mechanism](#)
[Defining Shareoptions and RLS attributes for data class](#)
[Specifying attributes for data set reuse and loading](#)
[Specifying attributes for backup-while-open \(BWO\) and recovery](#)

[Defining record and space attributes for data class](#)

[Defining volume and data set attributes for data class](#)

[Specifying attributes to handle space constraints during allocation](#)

[Defining VSAM attributes and specifying media types for data class](#)

[Defining the encryption management mechanism](#)

[Defining Shareoptions and RLS attributes for data class](#)

[Specifying attributes for data set reuse and loading](#)

[Specifying attributes for backup-while-open \(BWO\) and recovery](#)

Parent topic:[Defining data classes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining record and space attributes for data class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

On the Data Class Define panel, the SCDS Name and Data Class Name are output fields that contain the SCDS and data class names you specified in the Data Class Application Selection panel.

All of the input fields are optional and have a default value of blanks. A blank indicates that no value is assigned to a parameter, and end users need to specify the value explicitly if it is required and they intend to use the data class. Otherwise, their jobs might fail. When users override some data class attributes, they should ensure that the remaining attributes in the data class do not conflict with the explicitly specified attributes.

You can specify the following attributes on page 1 of the Define Data Class panel:

DescriptionIs an optional description for the data class. It can be up to 120 characters.**Recfm**Specifies the data set record format. You can specify one of the following, with the option of appending either an A (ISO/ANSI control character) or an M (Machine control character): **U**Undefined**V**Variable**VS**Variable spanned**VB**Variable blocked**VBS**Variable blocked spanned**FF**Fixed**FS**Fixed standard**FB**Fixed blocked**FBS**Fixed blocked standard If you specify Recfm, you cannot specify the Recorg attribute.**LrecI**Specifies the logical record length in bytes. If you leave the Recorg field blank, you can specify a LRECL value from 1-32760 (or leave it blank). If you choose a Recorg value of KS, ES, or RR, you can specify a LRECL value from 1-32761 (or leave it blank). If the Recorg value is KS, the LRECL value must be greater than or equal to the value you specify for the Keylen attribute. Also if the Recorg value is KS and you specify values for both the Keylen and Keyoff attributes, then the LRECL value must be greater than or equal to the sum of the Keylen and Keyoff attributes. If the Recorg value is LS, the LRECL attribute is ignored and the Csize value is 4096.

Override SpaceSpecifies whether DATA CLASS attributes override attributes obtained from other sources (JCL, AMS control cards or LIKE=). Override Space flag has two values: **YES**data class override takes effect.**NO**no data class override. NO is the default.You can override the following JCL Space subparameters, including dynamic allocations, such as TSO ALLOCATE: Space type (CYL,TRK, Block length, or record length plus AVREC)Primary QuantitySecondary QuantityDirectory blocks

You can override the following IDCAMS DEFINE space parameters: Cylinders (Primary, Secondary)Tracks (Primary, secondary)Kilobytes (Primary, Secondary)Megabytes (Primary, Secondary)Records (Primary, Secondary)Controlintervalsize (CISIZE DATA)Freespace (CI-percent, CA-percent) Note: You must explicitly specify AVGREC, AVG VALUE, PRIMARY, SECONDARY, and DIRECTORY. ISMF primes these fields with blanks, and returns an error message for each blank field. 0 is a valid value.If you set RECORC to KS, ES, RR, or LS, then you must also explicitly specify CISIZE DATA. If you set RECORC to KS, then you must also explicitly specify %FREESPACE CI and %FREESPACE CA. ISMF primes these fields with blanks, and returns an error message for each blank field. 0 is a valid value.SMS gets PRIMARY SPACE, SECONDARY SPACE, allocation units, and directory blocks either from JCL or from the DATA CLASS. The SPACE information in the DATA CLASS must be all inclusive, otherwise jobs might fail or produce unexpected results.If you set Override Space to YES, the SPACE information specified in DATA CLASS overrides the SPACE information specified on JCL, even for non-SMS managed data sets. The SPACE information in the DATA CLASS must be all inclusive, otherwise jobs might fail or produce unexpected results.

SpaceSpecifies a request for space in records, eliminating track and cylinder space requests. **Avgrec**Specifies a scaling factor for primary and secondary record allocations. It can have the following factors: **U**Multiplies the allocation quantity by 1**K**Multiplies the allocation quantity by 1024**M**Multiplies the allocation quantity by

1048576Avg ValueSpecifies the average length of each record.**Primary**Specifies the primary allocation quantity of records as multiplied by the scaling factor (Avgrec).**Secondary**Specifies the secondary allocation quantity of records as multiplied by the scaling factor (Avgrec).**Directory**Specifies the number of directory blocks for a PDS. It is valid only when the following fields are blank:
RecorgKeyoffClsize Data% Freespace Cl% Freespace CAShareoptions
XregionShareoptions XsystemIf you specify the following values: Avgrec = KAvg Value = 80Primary = 500Secondary = 100Directory = 100 you request 40000 KB of primary space, 8000 KB of secondary space, and 100 directory blocks. The JCL used to accomplish the same task looks like: //DD1 ... SPACE=(80,(500,100,100)),AVGREC=K ...

Retpd or ExpdtSpecifies the retention period or expiration date that is associated with the data class being defined. Retention period is the number of days (0 to 93000) and expiration date is the date when you want the definition to expire. The default is 'blank' for Expdt and 0 for Retpd, no expiration time.

Volume CountSpecifies the maximum number of SMS-managed DASD or mountable volumes a data set can span, with the following exceptions. Valid values are 1 through 59.

For striped VSAM data set, the volume count can be overridden by calculations derived from the sustained data rate (SDR) or the dynamic volume count which ever is greater up to the maximum of 59. For non-VSAM striped data set, volume count is determined by sustained data rate(SDR), dynamic volume count has no affect and the maximum count is 59. For managed mountable volumes, this value can be overridden by the volume count specified in JCL up to a maximum of 255. Dynamic volume count has no affect to tape data sets.

Volume count is ignored for data sets to which no storage class is assigned. The default is 1. Note: In JES3 systems, volume count is also ignored for managed mountable data sets.

Add'l Volume AmountSpecifies whether primary or secondary allocation amounts are to be used when the data set is extending to a new volume. You can specify: P for primaryS for secondaryIf you leave the field blank, the system default of primary is used. This attribute is used during VSAM EOV processing, and is only applicable to any VSAM multivolume data sets allocated in the extended format.

Parent topic:[Defining data class attributes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining volume and data set attributes for data class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

From Page 1 of the Data Class Define panel, issue the DOWN command to view Page 2, on which you specify volume and data set attributes.

Data Set Name TypeSpecifies the format in which the data set is to be allocated. When you specify extended format, you can also select requirements for the data set, including the need for extended addressability. See z/OS DFSMS Using Data Sets for detailed information on assigning and allocating data classes for extended format data sets.

The available values are: **EXT**Specifies that the data set to be allocated is in the extended format. All VSAM data set types can be allocated in the extended format (with the exception of key range data sets, temporary data sets and system data sets). If you specify EXT, you must also specify a value for the If Ext subparameter. The other subparameters are primed with the following defaults: Extended Addressability = No, and Record Access Bias = USER.

» Sequential extended format data sets can be version 1 or 2. The user can specify the version number with the second value of DSNTYPE on the DD statement or dynamic allocation equivalent. The PS_EXT_VERSION keyword in the IGDSMSxx member in PARMLIB can change the default for the version number. [«](#)

» If the sequential extended format data set is striped or if it resides only on one volume, then the version number does not matter. If it is not striped and it resides on multiple volumes, then the version number affects whether the system can use FlashCopy. FlashCopy can handle version 2 in that case but not version 1. PTFs are available to allow proper handling on releases earlier than z/OS® 2.1. [«](#)

HFSSpecifies that the data set is a hierarchical file system (HFS) data set. **LARGE**Specifies that the data set is a large format data set. Large format data sets are physical sequential data sets with the ability to grow beyond the limit of 65 535 tracks per volume. The BLOCKTOKENSIZE=REQUIRE option in the IGDSMSxx member of SYS1.PARMLIB affects what programs can open these data sets. See the description of the IGDSMSxx member in z/OS MVS Initialization and Tuning Reference.

These data sets cannot be written on a system older than z/OS 1.7. These data sets cannot be read on a system older than z/OS 1.7 if they have more than 65535 tracks on each volume.

LIBSpecifies that the data set is a PDSE. **PDS**Specifies that the data set is a partitioned data set. **blank**Leaves the Data Set Name Type attribute value unspecified. This is like coding DSNTYPE=BASIC on the DD statement but you cannot specify BASIC on this panel.

If you specify a Data Set Name Type of EXT for data sets allocated in extended format, you can specify additional attributes: **If Ext**Specifies whether allocation in extended format is preferred or required. If you specified EXT for the Data Set Name Type attribute, you must also specify one of the following: **P (preferred)**The data set allocation is attempted in nonextended format if the necessary system resources for extended are not available. **R (required)**The data set allocation fails if the data set cannot be allocated in extended format. **Extended**

AddressabilitySpecifies that a VSAM data set in the extended format be allocated using extended addressability. This lets the data set grow beyond the four gigabyte (4GB) size. The data set can be a VSAM data set in any record organization but must be allocated in the extended format. The available values are: **Y**Extended addressability is used. **N**Extended addressability is not used. This is the default.

Record Access BiasSpecifies whether to let VSAM determine how many and which type of buffers to use when accessing VSAM extended format data sets by batch processing. This is known as system-managed buffering, and is available to VSAM data sets in any record organization which are allocated in the extended format. Tip: The buffering algorithms determined by VSAM can be overridden by using JCL.

The available values are: **System**Specifies that VSAM chooses how many buffers to use and how they are processed. System-managed buffering only takes effect if the application requests the use of non-shared resources (NSR). It is not effective for applications requesting local shared resources (LSR), global shared resources (GSR), or record-level sharing (RLS).

The number of buffers and the buffering technique are determined by the system or the user: The system, based on application specifications (ACB MACRF=(DIR,SEQ,SKP)) and the values for direct and sequential millisecond response (MSR) and bias specified in the storage class. The user, as specified on the JCL DD card (AMP='ACCBIAS=').

If sequential processing is to be used, the system optimizes the number of buffers and uses the NSR buffering technique. If direct processing is to be used, the system optimizes the number of buffers and uses the LSR buffering technique. You can change the defaults for space and the relative amount of hiperspace when direct optimization is used by using the following keywords in the AMP= parameter: SMBVSP=, SMBHWT=, and SMBDFR=.

UserSpecifies that system-managed buffering is not used. This is the default.**SO**Specifies that system-managed buffering with sequential optimization is to be used.**SW**Specifies that system-managed buffering weighted for sequential processing is to be used.**DO**Specifies that system-managed buffering with direct optimization is to be used.**DW**Specifies that system-managed buffering weighted for direct optimization is to be used.

RMODE31RMODE31 allows you to specify whether or not to allocate the buffers and control blocks in 31-bit addressable storage. The values you may specify for RMODE31 are: **ALL**Control blocks and buffers above the line.**BUFF**Buffers (only) above the line.**CB**Control blocks (only) above the line.**NONE**Control blocks and buffers below the line.

Dynamic Volume CountDynamic Volume Count is used during allocation processing to determine the maximum number of volumes a data set can span. The number can be in the range 1 through 59. The default value is 1. 59 is the z/OS volume limit. During define processing, Dynamic Volume Count allows for a larger number of volumes to be considered without increasing the number of candidate volumes stored in the catalog. During existing data set allocation, it provides a way to increase the number of TIOT/JFCB entries that are created, so that more volumes can be dynamically allocated as required during the lifetime of the allocation. For VSAM data sets, the Dynamic Volume count is the maximum number of volumes that all components in the sphere being allocated can span. Note: Dynamic volume count support when a data set extends is a function of access method end-of-volume (EOV) processing and not the data set type. For products that do not use the standard IBM® access method EOV interface for their data sets, DVC may not be supported. Please consult with your product representative to see if it supports dynamic volume count.

All the following must be true for the Dynamic Volume Count value to be valid: The Dynamic Volume Count is larger than the current volume count of the data set. The data set is SMS-managed. The Space Constraint Relief value is Y.

Dynamic Volume Count is not supported for the following: Non-SMS managed data sets or data sets with no associated data class System data sets: Page data sets Single-volume data sets such as a catalog BCS or VVDS, VSAM temporary data sets, or any system data set with ACBSDS=ON SAM striped data sets VSAM data sets with the IMBED or KEYRANGE options VSAM EOV Snapshot processing. The Dynamic Volume Count field has significance only when it is larger than administrator-specified data class Volume Count, and any user-specified volume

information provided by JCL, IDCAMS, or TSO.

For VSAM data sets associated with a data class with a dynamic volume count that is greater than 1, users must have ALTER access to the RACF resource that protects the data sets.

CompactionSpecifies whether data is to be compressed. You can compress data on tape, or on DASD if the data set is allocated in the extended format. This field is ignored for DASD data sets if the data set name type is not EXT. A compressed data set cannot reside on the same cartridge as a data set that is not compressed.

» For physical sequential data sets, you can specify one of the following:
YExtended format data sets are compressed and tape volumes are compacted, but the type of DASD compression depends on the COMPRESS option in the IGDSMSxx PARMLIB member.
NData sets are *not* compressed. Tape volumes are *not* compacted, unless requested by the user on JCL/dynamic allocation. This is the default, if left blank.
TExtended format data sets are compressed using tailored dictionaries.
GExtended format data sets are compressed using generic dictionaries.
ZRIndicates "zEDC Required", meaning that the system should fail the allocation request if the zEDC function is not supported by the system, or if the minimum allocation amount requirement is not met.
ZPIndicates "zEDC Preferred", meaning that the system should *not* fail the allocation request, but rather create either a tailored compressed data set if the zEDC function is not supported by the system, or a non-compressed extended format data set if the minimum allocation amount requirement is not met.

«
T, **G**, **ZR**, and **ZP** override the compression option set in the IGDSMSxx PARMLIB member, and let you select the type of compression on a data set level. Current users of generic compression can move to using tailored or zEDC compression one data set at a time, as new data sets are created.

Spanned / NonspannedSpecifies whether a data record can span control interval boundaries. This applies to both system-managed and non-system-managed data sets. Specify one of the following:
SpannedSpecifies that if the size of a data record is larger than a control interval, the record can be contained on more than one control interval. This lets VSAM select a control interval size that is optimum for the DASD. When a data record that is larger than a control interval is put into a cluster defined for spanned record format, the first part of the record fills a control interval. Subsequent control intervals are filled until the record is written into the cluster. Unused space in the record's last control interval is not available to contain other data records.

Restriction: Do not use this attribute for a variable-length relative record data set (VRRDS).

NonspannedSpecifies that a record must be contained in one control interval. VSAM selects a control interval size that accommodates the largest record in the data set. This is the default.
System Managed BufferingSpecifies the amount of virtual storage for SMB Direct Access Bias obtained for buffers when opening the data set. The possible values are: 1K to 2048000K 1M to 2048M

System Determined BlocksizeSpecifies whether DADSM or OPEN is to use system-determined block size processing:
YSystem-determined block size processing is used only when there is no block size specified on the JCL DD statement, DCB, DCBE and in the DCB OPEN exit routine.
NNo change in block size processing. N (No) is the default.
EATTRA data set level attribute specifying whether a data set can have extended attributes (format 8 and 9 DSCBs) and optionally reside in EAS.
NONo extended attributes. The data set can not have extended attributes (format 8 and 9 DSCBs) and cannot reside in EAS. This is the default behavior for non-VSAM data sets.
OPTExtended attributes are optional. The data set can have extended attributes (format 8 and 9 DSCBs) and can optionally reside in EAS. This is the default behavior for VSAM data sets.

Parent topic:Defining data class attributes

◀ □ | □ ▶

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying attributes to handle space constraints during allocation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can specify attributes on Page 2 of the Data Class Define panel to indicate whether or not to retry new data set allocations or extends on new volumes that fail due to space constraints.

During allocation, there might not be enough space on a volume to meet the requested space. SMS volume selection can sometimes solve this problem by trying all candidate volumes before failing the allocation. You can also use the Space Constraint Relief and Reduce Space Up To (%) attributes to request that an allocation be retried if it fails due to space constraints. SMS retries the allocation by combining any of the following:

Spreading the requested quantity over multiple volumes

Allocating a percentage of the requested quantity

Using more than 5 extents

Space Constraint ReliefSpecifies whether or not to retry an allocation that was unsuccessful due to space constraints on the volume. Before it is retried, the allocation is attempted on all candidate volumes. Space Constraint Relief applies only to system-managed data sets, and is limited to new data set allocations, and while extending the data set on new volumes. Specify one of the following:

YSpecifies that SMS retry the allocation.

NSpecifies that SMS does not retry the allocation, so that allocation is not attempted on multiple volumes. This is the default.

If you specify Y, SMS begins the retry process. This is a one- or two-step process, depending on the volume count you specified. For JCL allocations, SMS determines the volume count by taking the maximum of the unit, volume, or volser count. If these are not specified, SMS picks up a volume count from the data class. If there is no data class, SMS defaults the volume count to 1. If the volume count is 1 (one-step process) SMS retries the allocation after reducing the requested space quantity based on the Reduce Space Up To attribute. SMS simultaneously removes the 5-extent limit, so that SMS can use as many extents as the data set type allows.

If the volume count is greater than 1 (two-step process) First, SMS uses a *best-fit* volume selection method to spread the primary quantity over more than one volume (up to the volume count). If this fails, SMS continues with the best fit method after reducing the primary quantity and removing the 5-extent limit.

Tip: You can remove the 5-extent limit without reducing the primary quantity by specifying 0 for the Reduce Space Up To (%) attribute.

For extends to new volumes, space constraint relief is strictly a one-step process. If regular volume selection has failed to allocate space, SMS reduces space or removes the 5-extent limit, but does not try the best-fit method.

The maximum number of extents per volume and the maximum number of volumes per data set vary depending on data set type as follows: A basic-format or large-format sequential data set and a direct data set can have up to 16 extents per volume and up to 59 volumes. An extended-format sequential data set can have up to 123 extents per volume and up to 59 volumes. Either all or none of these volumes can be arranged into stripes for parallel processing. A non-system-managed VSAM data set can have up to 255 extents per component and up to 59 volumes. A system-managed VSAM data set can have up to 255 extents per stripe and up to 59 volumes. This extent limit can be removed if the associated data class has extent constraint removal specified. Up to 16 volumes at a time can be read or written in parallel due to striping. A PDS can have up to 16 extents and only one volume. A PDSE can have up to 123 extents and only one volume. An HFS data set can have up to 123 extents per volume and up to 59 volumes.

Reduce Space Up to (%)Specifies the amount by which you want to reduce the requested space quantity when the allocation is retried. You must specify Y for the Space Constraint Relief attribute. Valid values are 0 to 99. If you request space constraint relief but do not specify a percentage value (either 0 or blank), SMS does

not reduce the requested space quantity. This implies your application cannot tolerate a reduction in the space to be allocated, so only the 5 extent limit is relieved.

When you request space constraint relief in one or more data classes, you might notice any of the following: Very large allocations might succeed if a sufficiently large volume count is specified in the data class or through JCL. Existing data sets might end up with less space than initially requested on extents. The space allocated for new data sets might be less than requested. The number of extents used during initial allocation might result in fewer extents being subsequently available. For example, if the primary space allocation uses 10 extents when allocating a physical sequential data set, then only 6 extents are left for allocation of the secondary quantity. X37 abends should occur less frequently.

Parent topic:[Defining data class attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining VSAM attributes and specifying media types for data class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

From Page 2 of the Data Class Define panel, use the DOWN command to view Page 3, on which you specify VSAM attributes and media.

RecorgSpecifies the data set organization, and it resembles the RECORG DD attribute. If you specify a Recorg value, you cannot specify the Recfm attribute. A blank value specifies either a physical sequential or a partitioned organization (non-VSAM data set). If you do not specify a Recorg value for data sets with a data class, assigned either by JCL or ACS routine, the DSORG defaults to either physical sequential (PS) or partitioned organization (PO). Data class does not have a DSORG field. To specify a physical sequential data set, specify RECFM. To specify partitioned organization, specify Recfm and Space values for directory blocks. See z/OS DFSMS Using Data Sets.

[Table 1](#) summarizes which data class attributes apply to each record organization (RECORG). Table 1. Applying Data Class Attributes to Record Organization (Recorg)

See the following for more information: z/OS DFSMS Access Method Services Commandsz/OS DFSMS Using Data Setsz/OS MVS JCL Reference

KeylenSpecifies the key length in bytes. To use this attribute, you must specify either KS or blank for the Recorg attribute. If the Recorg value is KS, the Keylen attribute represents the length of the KSDS key field and ranges from 1-255, or you can leave it blank. If the Recorg value is blank, the Keylen value ranges from 0-255, or you can leave it blank. For either value of Recorg, you must assign a value to the Keylen attribute that is less than or equal to the LRECL value.**Keyoff**Specifies the displacement from the beginning of a record to the KSDS key field. It is valid only when the Recorg value is KS. The Keyoff value can range from 0 to the value of (LRECL - Keylen). If a Keylen value is specified, then the Keyoff attribute must also be specified, or 0 value is used. This value is merged only if the Keylen value is not specified in JCL.

ClsizE DATASpecifies the control interval size for the data component of data sets having Recorg values of KS, ES, RR, or LS. The default is 4096.%

FreespaceSpecifies the percentage of each control interval and control area to be set aside as free space when the cluster is initially loaded or when a mass insert is done. It applies only to the data component. The Freespace attribute is valid only if the Recorg value is KS or blank. The default is blank.**Media Interchange**Provides the capability to control the type of media created to make the data and media acceptable to other processors in the same or different locations. For example, you can create a data class with a name of BOSTON and define the media interchange parameters (Media Type and Recording Technology attributes) compatible with the tape hardware in the Boston location. When the hardware in Boston is changed, the BOSTON data class can be changed to the new media interchange parameters.**Media Type**Specifies mountable tape media cartridge type. The valid media types are MEDIA1, MEDIA2, MEDIA3, MEDIA4, MEDIA5, MEDIA6, MEDIA7, MEDIA8, MEDIA9, MEDIA10, or blank. This field is optional.

Recording TechnologySpecifies recording technology for mountable tape media cartridges. This field is optional and dependent on the value specified in the Media Type field. If MEDIA1 is specified, either 18TRACK or 36TRACK is valid. If MEDIA2 is specified, only 36TRACK is valid. If MEDIA3 or MEDIA4 is specified, either 128TRACK or 256TRACK is valid. If MEDIA5, MEDIA6, MEDIA7, or MEDIA8 is specified, EFMT1, EFMT2, EFMT3 and EEFMT3 are valid. If MEDIA9 or MEDIA10 is specified, EFMT2, EFMT3, and EEFMT3 are valid. Refer to z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries for further information on data class attributes and the Recording Technology field.

Performance ScalingSpecifies the performance scaling preference. The valid values are Y, N, or blank. This field is optional. If you select a value of Y, it means

that you want performance scaling. Performance scaling, however, is valid only when you select MEDIA5 or MEDIA9. This value is not valid if you select any other media type. Also, the appropriate recording technology must be used for the selected media type. If you select a value of N or blank, it means that you do not want performance scaling. Performance scaling and performance segmentation are mutually exclusive. **Performance Segmentation** Specifies whether the system should enable segmentation format of the tape. The value specified in this field is considered only for the devices that support the segmentation for performance feature. The valid values are Y, N, or blank. This field is optional. If you select a value of Y, it means that you want performance segmentation. Performance segmentation, however, is valid only when you select MEDIA5 or MEDIA9. This value is not valid if you select any other media type. Also, the appropriate recording technology must be used for the selected media type. If you select a value of N or blank, it means that you do not want performance segmentation. Performance segmentation and performance scaling are mutually exclusive. **Block Size Limit** Specifies the block size limit for new tape data sets on SMS and non-SMS managed storage. This value can be from 32760 - 2147483648, 32 KB - 2097152 KB, 1 MB - 2048 MB, or 1 GB - 2 GB. The default is blank. When a program opens a tape data set without a block size value for output, the system determines a block size. This system-determined block size value is less than or equal to the first value in the following: BLKSZLIM keyword on the DD statement or the dynamic allocation equivalent block size limit specified in data class (even for non-SMS-managed data sets) TAPEBLKSZLIM keyword in the DEVSUPxx member of SYS1.PARMLIB32760 Currently, OPEN chooses a block size value of no greater than 262144, or 256 KB. Any larger limit has no effect.

Parent topic:[Defining data class attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining the encryption management mechanism

z/OS DFSMSdfp Storage Administration
SC23-6860-01

From Page 3 of the Data Class Define panel, use the DOWN command to view Page 4, on which you specify the key labels and encoding mechanism.

Key LabelSpecifies the label for the key encrypting key used by the Encryption Key Manager. The key encrypting key is used to encrypt the data (encryption) key. To specify the key label value, you can use up to 64 characters with blanks padding the field on the right. The characters can be alphanumeric, national, or special characters with some additional characters also allowed. It is treated as a free form field on input and validity checked by the control unit when the key label is first used and converted from EBCDIC to ASCII. The characters specified through ISMF must map to ASCII characters X'20' to "7E".**Encoding Mechanism**Specifies how the label for the key encrypting key specified by the key label (input) is encoded by the Encryption Key Manager and stored on the tape cartridge. **L** = encoded as the specified label**H** = encoded as a hash of the public key

Parent topic:[Defining data class attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining Shareoptions and RLS attributes for data class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

From Page 4 of the Data Class Define panel, use the DOWN command to view Page 5, on which you specify Shareoptions and RLS attributes.

ShareoptionsSpecifies how end users can share a component or cluster. The Xregion value specifies the amount of sharing allowed among regions within the same system, or within multiple systems using global resource serialization (GRS). You can specify the following values: **1**Any number of users can read the data set at one time, *or*only one can write to it. This setting does not allow any type of non-RLS access when the data set is already open for RLS processing.**2**Any number of users can read the data set at one time, *and*only one can write to it. If the data set is already open for RLS, non-RLS users can read the data set but cannot write to it. If the data set has been opened for non-RLS output, an RLS open fails.

Requirement: You must apply APARs OW25251 and OW25252 to allow non-RLS read access to data sets already open for RLS processing.

3Any number of users can share the data set, and each is responsible for maintaining read and write integrity. This setting does not allow any type of non-RLS access when the data set is already open for RLS processing.**4**Any number of users can share the data set, and buffers used for direct processing are refreshed for each request. This setting does not allow any type of non-RLS access when the data set is already open for RLS processing.

The Xsystem value specifies the amount of sharing allowed among systems. You can specify the following values: **3**Any number of users can share the data set, and each is responsible for maintaining read and write integrity.**4**Any number of users can share the data set, and buffers used for direct processing are refreshed for each request. If values are not specified explicitly or in the data class, VSAM defaults are used.

RLS CF Cache ValueSpecifies the amount of data that will be cached. This keyword is honored only when RLS_MaxCfFeatureLevel(A) is active in the sysplex. DFSMS CF cache structures are connected to the system when the first VSAM RLS instance is opened on the system. At this time, message IGW500I is issued to indicate whether the RLS CF Cache Value keyword is honored.

You can specify the following values:

ALLIndicates that VSAM index and data components will be cached. **ALL** is the default.**NONE**Indicates that only the VSAM index data will be cached. The data components will not be placed in the cache structure.**UPDATESONLY**Indicates that only WRITE requests will be placed in the cache structure.**DIRONLY**Indicates that RLS will not cache the data or index parts of the VSAM data set in the coupling facility cache structure. In this case, RLS will use the XCF cache structure to keep track of data that resides in permanent storage (DASD) and in local storage but data or index CIs are not stored in the cache structure itself.**RLS Above the 2-GB Bar**Specifies whether the SMSVSAM address space can take advantage of 64-bit addressable virtual storage during VSAM RLS buffering. You can specify the following values:

YIndicates that VSAM RLS buffers can reside above the 2-gigabyte bar. This setting is recommended for best performance of high-volume applications that use VSAM RLS buffering.**N**Indicates that VSAM RLS buffering is limited to storage below the bar. **N** is the default.

Extent Constraint RemovalSpecifies whether data sets are allowed to be extended beyond 255 extents.

CA ReclaimSpecifies whether the DASD space for empty control areas (CAs) will be reclaimed for key-sequenced data sets (KSDS). Specify one of the following:

YIndicates that DASD space for empty CAs will be reclaimed. This is the default.

NIndicates that DASD space for empty CAs will not be reclaimed. The value for CA Reclaim is always saved in the catalog, but is used only if the CA reclaim function is

enabled in the IGDSMS xx member of PARMLIB or with the SETSMS command. The setting for CA Reclaim for a data class is processed only when a catalog entry for a KSDS is defined. If you need to change the CA reclaim setting for an existing KSDS based on the contents of the data class, either use the ALTER command for the KSDS or delete and define the KSDS to use the value in the data class. For more information, see the topic about [CA reclaim](#) in z/OS DFSMS Using Data Sets.

Parent topic:[Defining data class attributes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying attributes for data set reuse and loading

z/OS DFSMSdfp Storage Administration
SC23-6860-01

On Page 4 of the Data Class Define panel, you can also specify attributes to indicate whether the VSAM cluster can be reused and how the data set is to be loaded.

ReuseSpecifies whether the VSAM cluster can be opened again as a new data set. Specify one of the following: **Y (Yes)**The cluster is reusable.**N (No)**The cluster is not reusable.**Initial Load**Specifies how the data set is to be loaded. Specify one of the following: **S (Speed)**The data set is loaded without being preformatted.**R (Recovery)**The data set is preformatted when it is loaded. This is the default.

Parent topic:[Defining data class attributes](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying attributes for backup-while-open (BWO) and recovery

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can specify whether a data set is eligible for backup-while-open (BWO) processing. You can also indicate whether a data set is recoverable or not, and if so, provide the name of the forward recovery log stream.

Restriction: These attributes are only available to system-managed data sets.

BWOSpecifies whether BWO is to be used. This applies only to system-managed VSAM data sets, and is not available for linear data sets. Specify one of the following: **TYPECICS**BWO processing is used for CICS® VSAM file control data sets. SMS rejects dynamic or JCL allocations if, as the result of an alter function, the value for LOG is changed to ALL without a logstream ID being available, or if the logstream ID is nullified.**TYPEIMS**BWO processing is used for IMS™ VSAM data sets. Note: Support for this option is only available with IMS 6.1 or above.

NOBWO is not used for CICS VSAM file control or IMS VSAM data sets. This is the default.**FRlog**Specifies whether VSAM batch logging is to be performed for your VSAM data set. Specify one of the following: **ALL**Tells VSAM to do both forward and backward recovery logging. Changes made by applications are written to the MVS log stream indicated on the logstream ID parameter.**NONE**Disables the VSAM batch logging function for your VSAM data set. Changes made by applications are **not** written to the MVS log stream indicated on the logstream ID

parameter.**REDO**Tells VSAM to do forward recovery logging. Changes made by applications are written to the MVS log stream indicated on the logstream ID parameter.**UNDO**Tells VSAM to do backward recovery logging. Changes made by applications are written to the MVS log stream indicated on the logstream ID parameter.**blank**The FRlog value in the catalog is used. If you specify FRlog(ALL), FRlog(REDO), or FRlog(UNDO), you must specify the logstream ID parameter for the VSAM data sets. If you do not specify the logstream ID, message IEC161I is issued. There is no default JCL value. If FRlog is omitted, the catalog value is used.

LogSpecifies whether the data set is considered recoverable or not. For data sets defined using access method services, the Log and Logstream ID attributes in the data class are merged with those defined in the DEFINE command. If a single logstream is to be used for each VSAM data set, the logstream ID should be specified on the DEFINE command. Specify the logstream ID in the data class only if the same logstream ID is to be used for many data sets. Otherwise, this will result in too many data classes.

Additionally, SMS rejects dynamic or JCL allocations if, as the result of an alter function, the value for LOG is changed to ALL without a logstream ID being available, or if the logstream ID is nullified. Specify one of the following:

NONEIndicates that neither an external backout nor a forward recovery capability is available, so the data set is not considered recoverable**UNDO**Indicates that changes can be backed out using an external log, so the data set is considered recoverable**ALL**Indicates that changes can be backed out and forward recovered using an external log If you specify Log(ALL), you must specify a logstream ID, either on the access method services DEFINE command or in the Logstream ID field in the data class.

blankThe data set is not recoverable. This is the default.**Logstream ID**Identifies the CICS forward recovery log stream. It applies to all components of the sphere. If you specify Log(ALL) or FRlog(REDO), you must specify a logstream ID. A logstream ID is made up of 1-to-26 characters, including separators. This name is made up of one or more segments, each containing one to eight alphabetic, numeric, or national characters. The first character of each segment must be an alphabetic or national character. Segments are joined by periods.

For data sets defined using access method services, the attributes in the data class are merged with those defined in the DEFINE command. If a single logstream is to be used for each VSAM data set, the logstream ID should be specified on the

DEFINE command. Specify the logstream ID in the data class only if the same logstream ID is to be used for many data sets. Otherwise, this will result in too many data classes.

Log ReplicateSpecifies whether the data set is eligible for replication:
YYes, data set is eligible for VSAM replication.
NNo, data set is not eligible for VSAM replication. This is the default.

Parent topic:[Defining data class attributes](#)



 Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Assigning data classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can define an ACS routine to determine data classes, or end users can explicitly specify a data class name on the following: JCL DD statements TSO/E ALLOCATE commands Access method services ALLOCATE and DEFINE commands Dynamic allocation requests such as through ISPF/PDF data set allocation panels End users can explicitly specify data set attributes, and these specifications take precedence over the data set attributes assigned through the data class ACS routine.

[Specifying data classes outside ACS routines](#)

[Processing data class attributes in JCL](#)

Parent topic:[Defining data classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying data classes outside ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The syntax for specifying a data class on a JCL statement is: `DATACLAS=data-class-name`
The format for specifying a data class on a TSO/E command is: `DATACLAS(data-class-name)`
The format for specifying a data class on an access method service command is:
`DATACLAS(-)`

Parent topic:[Assigning data classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Processing data class attributes in JCL

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The order of precedence for data class attributes in JCL is as follows:

Explicit specifications
LIKE and REFDD keywords
Data class definitions (explicit or derived) in REFDD statement
Data class definitions in the referencing DDA data class does not need to be a self-contained, complete data organization. You can partially define the data set attributes in the data class definition (as a base) and the user can explicitly specify the remaining attributes. However, the merging of all attributes according to the order specified above must result in a valid data organization.

In the example below, //DD1 DD ... ,DATACLAS=DC2,RECORG=ES,...

//DD2 DD ... ,DATACLAS=DC1,LRECL=180,REFDD=DD1,...

For the attributes for DD2, LRECL of 180 is used first, and then the RECORG of ES in DD1 is used, regardless of the values specified in either the DC1 or the DC2 data class. For the remaining attributes that are not explicitly specified on the DD statements, SMS uses the values defined in the data class definition of DC2 and then DC1.

In this next example, the attributes of the data set referenced by the LIKE keyword are used after all other explicit specification but before data class DC3 attributes:

//DD3 DD ... ,DATACLAS=DC3,LRECL=180,LIKE=SAMPLE.DATA,...

Here, an LRECL of 180 is used. Then, SMS uses the Data Set Control Block (DSCB) information from the SAMPLE.DATA data set. Finally, the remaining attribute values are drawn from the DC3 data class.

A final example illustrates the use of REFDD: //DD4 DD ... ,DATACLAS=DC4,...

//DD5 DD ... ,DSN=DS1,REFDD=DD4,DATACLAS=DC5,LRECL=180,...

In this example, the REFDD keyword specifies that the explicit attributes on the DD4 JCL statement are to be used second, because explicit attributes on the DD4 JCL statement are used after the explicitly specified attributes. Next, the attributes from data class DC4 referenced in DD4 should be used. The remaining attributes are taken from data class DC5.

You can use the DATACLAS keyword to specify data class attributes for SMS-managed or non-SMS-managed data sets. Or, you can use the DCB=*.ddname or DCB=dsname keywords to copy the attributes of existing data sets for new non-SMS-managed data set allocations. You can specify the attributes of new SMS-managed data sets with the LIKE or REFDD keyword. The LIKE and REFDD keywords are mutually exclusive. Restriction: You cannot use the REFDD keyword to copy DCB attributes from the DSCB. Use the LIKE keyword to copy from the DSCB.

For more information on using the LIKE and REFDD keywords, refer to z/OS MVS JCL Reference. For information on determining data classes through ACS routines, see [Defining ACS routines](#). For information on access method services DEFINE defaults and data class defaults, refer to z/OS DFSMS Access Method Services Commands.

Parent topic:[Assigning data classes](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining additional data classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can copy existing data classes and modify them to create new ones by using the COPY line operator from the List panel, which is explained in [Copying SMS components](#).

Parent topic:[Defining data classes](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining aggregate groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Individual SMS data sets in pool storage groups are backed up based on the backup attributes that are in a management class. Using ISMF's aggregate group application, you can group and back up data sets according to application or backup requirements. This topic describes aggregate groups and shows you how to define them using the ISMF aggregate group application.

[Understanding aggregate groups](#)

[Planning aggregate groups](#)

[Defining aggregate groups](#)

[Defining additional aggregate groups](#)

 | 

©Copyright IBM Corporation 1990, 2014

Understanding aggregate groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

An aggregate group is an SMS construct that uses control information and data set lists to define an application or other group. The data stored in the aggregate group is the principal input to DFSMSHsm application backup and recovery.

An aggregate group consists of backup criteria and a group of data sets selected for backup by the storage administrator according to application or other requirements. See *z/OS DFSMSHsm Storage Administration* for further information on backing up or recovering aggregate groups.

The aggregate group application allows you to:

- Generate a list of aggregate groups
- Display the attributes of a single aggregate group
- Define or alter the attributes of a single aggregate group
- Delete aggregate groups
- Edit and browse the selection data sets associated with aggregate groups
- Edit and browse the instruction data sets associated with aggregate groups
- Back up the selected aggregate group
- Recover an aggregate group that has already been backed up
- Specify one to fifteen local copies of aggregate backup (ABACKUP) output files to be created
- Assign aggregate backup attributes to an aggregate group by specifying a management class name

Restriction: You cannot assign an object to an aggregate group.

ABACKUP allows specifying the number of copies of the Aggregate Backup and Recovery Support (ABARS) output files to be created using ISMF. The Copies value can be 1-15. The default is 1.

In order to support the creation of multiple copies, a unique name must be created for each copy. This entails using a new suffix convention for the output data sets:
.D.CccVnnnn for the DFMSdss data file.
.O.CccVnnnn for the internal data file.
.C.CccVnnnn for the control file.
.I.CccVnnnn for the instruction or activity log file
where **cc**Represents the copy number (a number from 1-15 of the copy created)
nnnnRepresents the version number The name supports the elimination of the GDG support.

Parent topic:[Defining aggregate groups](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Planning aggregate groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

First, you should identify those applications that are vital to continuing operation. Then, you must identify the application's associated components. These include: JCL and proceduresSource, object, and load module data setsProcedures and run booksRequired system data setsApplication data

Next, you should identify the required primary and recovery locations to ensure that the operating environments are compatible and the necessary resources are available. A primary location is the location that does an aggregate group backup of a specific application. A recovery location is the location where the aggregate group recovery of that specific application is performed.

Last, you should consider data set naming conventions to avoid duplicate data set names, and if system-managed data sets are to be recovered, you must ensure that the SMS constructs and attributes are compatible. Once the data sets needed to recover an application are identified, you can define the aggregate group.

Parent topic:[Defining aggregate groups](#)



©Copyright IBM Corporation 1990, 2014

Defining aggregate groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use ISMF to define, alter, list, display, back up or recover an aggregate group, or edit the selection or instruction data sets associated with an aggregate group by selecting option 9, Aggregate Group, from the ISMF Primary Option Menu for Storage Administrators. A selection data set is a data set which contains lists of data sets to be included in the backup of an application or other group. An instruction data set is a data set which contains instructions, commands, and so on, that are copied into the control file volume after the backup control file.

This topic describes defining aggregate groups. See the following for further information on the other options: [Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF](#) for additional information on option 1 [Altering aggregate groups](#) for additional information on option 4 [Backing up and recovering an aggregate group](#) and [z/OS DFSShsm Storage Administration](#) for additional information on options 5 and 6.

Perform the following steps to define storage groups.

Select option 9, Aggregate Group, from the ISMF Primary Option Menu for storage administrators. This displays the Aggregate Group Application Selection panel.

Supply values on the Aggregate Group Application Selection panel: **CDS**

Name This is the name of an SCDS. ISMF primes the CDS Name field with the name last used for an aggregate group. The default is 'Active'. This represents the currently active configuration, but you cannot define or alter aggregate groups to the 'Active' configuration. **Aggregate Group Name** This is the name of the aggregate group. ISMF primes the field with the name last used. **Option** Select option 3, Define. This displays the Aggregate Group Define panel

Supply values on the Aggregate Group Define panel. You can leave the Aggregate Group Define panel at any time without saving the aggregate group attributes or changes by issuing the Cancel command. For more information, see: [Defining aggregate group attributes](#) [Editing aggregate group attributes](#)

[Defining aggregate group attributes](#)

[Editing aggregate group attributes](#)

[Backing up and recovering an aggregate group](#)

Parent topic: [Defining aggregate groups](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining aggregate group attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Page one of the Aggregate Group Define panel contains aggregate group define attributes. The SCDS Name and Aggregate Group Name fields are output fields that contain the SCDS and aggregate group names you specified on the Aggregate Group Application Selection panel.

DescriptionDescribes the aggregate group, and can be up to 120 characters. It is optional.**Number of Copies**Specifies the number of aggregate backup output files to be created. The valid values are 1-15. It is required.**Management Class**

NameSpecifies the management class name from which the Aggregate Backup attributes are obtained. The valid values are 1-8 alphanumeric characters (first character not a digit) or a blank.**Output Data Set Name Prefix**Identifies the output data sets created by aggregate backup. It is required.

Parent topic:[Defining aggregate groups](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Editing aggregate group attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

After specifying backup attribute values, issue the DOWN command to view page two of the Aggregate Group Define panel. Page two of the Aggregate Group Define panel contains the selection and instruction data set names for the aggregate group. The SCDS Name and Aggregate Group Name fields are output fields that contain the SCDS and aggregate group names you specified on the Aggregate Group Application Selection panel. **Edit a Data Set**Select the number of a selection or instruction data set that you want to edit. When you select a data set number, it allows you to allocate or modify the data set by invoking PDF Edit. The PDF edit screen is shown in [Figure 1](#). See z/OS ISPF User's Guide Vol I for more information on the PDF Edit commands.

Selection Data Set NameName of the data set containing lists of data sets to be included in the application backup. You can specify up to five selection data set names. One data set name is required. There is no default. If you want to enter a fully qualified data set name, enclose the name in single quotation marks. If you do not enclose the name in single quotation marks, the TSO prefix is added to the name as the first high level qualifier. There is no default.

Member NameName of the data set member containing lists of data sets to be included in the application backup if the selection data set is partitioned. This name must be a valid TSO data set member name. Enter a valid member of the partitioned data set specified in the Selection Data Set Name field. This is required if the data set specified in the Selection Data Set Name field is a partitioned data set. If you want to enter a fully qualified data set name, enclose the name in single quotation marks. If you do not enclose the name in single quotation marks, the TSO prefix is added to the name as the first high level qualifier. There is no default.

Instruction Data Set NameName of the data set containing instruction, commands, etc., that are copied into the control file volume after the backup control file. This data set can only be a sequential data set. You must use a valid TSO data set name. The data set name, including the TSO prefix, can be no more than 44 characters long. This is an optional field and has no default.

If you select the option to edit a Selection or Instruction Data Set on the Aggregate Group Define panel, you get the PDF Edit screen shown in [Figure 1](#).

Figure 1. Editing an Aggregate Group Selection Data Set- Example

```
SELECT.DATASET.ONE ----- LINE 000000 COL 001 072
COMMAND ==> SCROLL ==> HALF
***** ***** TOP OF DATA *****
000001 INCLUDE( XMP.** )
000002 EXCLUDE( XMP.USER.TAPE1,
000003           XMP.INPUT.MASTER )
000004 ACCOMPANY( DATA.MASTER,          /* MASTER */
000005           XMP.USER.TAPE1 )
000006 ALLOCATE( XMP.OLD.DASD1 )
***** ***** BOTTOM OF DATA *****
```

You can specify the following keywords with parameters using PDF edit:

INCLUDESpecifies which data sets are included in the backup. No distinction is made whether they are tape or DASD. If a partitioned data set is indicated, all members are included in the application backup.

EXCLUDESpecifies the data sets that are specifically excluded from the backup process.

ACCOMPANYSpecifies data sets that are physically removed from the backup site, transported to the recovery site, and only need to be cataloged during application recovery.

ALLOCATEAllocates and defines data sets at the recovery site without copying the data from the source. For VSAM data sets, only the base cluster is defined, and the AIXs and pathnames are not. See z/OS DFSMSHsm Storage Administration for further information.

When you allocate or edit selection data sets, there are several rules you must follow: Records must be 80 bytes in length and of a fixed format. Entries must be entered between columns 1 and 72. INCLUDE, EXCLUDE, ACCOMPANY, and ALLOCATE can be specified only once. A comment is a string of characters preceded by "/" and followed by "/*" and might span multiple records. A separator consists of a comma (,), one or more blanks, or a comment. Parameters 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 of a record is optionally specified by a hyphen (-) or plus sign (+) as the rightmost nonblank character, preceded by one or more blanks. Continuation characters are not required. 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. Data set names specified in the INCLUDE list cannot be specified in the ALLOCATE or ACCOMPANY list. The minimum truncation allowed for keywords is: I for INCLUDE, E for EXCLUDE, AC for ACCOMPANY, and AL for ALLOCATE.

Parent topic:[Defining aggregate groups](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Backing up and recovering an aggregate group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To back up an aggregate group, select option 5, ABACKUP, on the Aggregate Group Application Selection panel, or enter the ABACKUP line operator on the Aggregate Group List panel, and press Enter. You get the Aggregate Group Backup panel.

To recover an aggregate group, select option 6, ARECOVER, and press Enter. You get the Aggregate Group Recover panel.

DFSMShsm handles your request for backup or recovery. You are prompted to enter information for backing up or recovering aggregate groups.

Parent topic:[Defining aggregate groups](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining additional aggregate groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can copy existing aggregate groups and modify them to create new aggregate groups by using the COPY line operator, which is explained in [Copying SMS components](#).

Parent topic:[Defining aggregate groups](#)

|

©Copyright IBM Corporation 1990, 2014

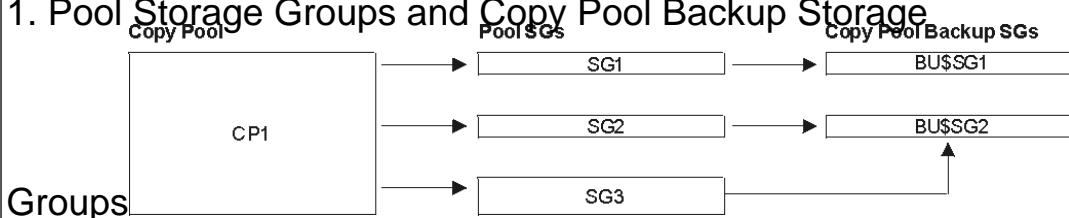
Defining copy pools

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Copy Pool Name specifies the name of the copy pool. The maximum length is 23 alphanumeric or national characters, or any combination. With one exception where the underscore character ('_') is also allowed in the combination. The first character can not be numeric or underscore.

DFSMShsm manages the use of volume-level fast replication functions, such as FlashCopy® and SnapShot. These functions provide point-in-time copy services that can quickly copy data from a source location to a target location. Using a copy pool, you can specify the pool storage groups that you want DFMSHsm to process collectively for fast replication.

Each pool storage group in a copy pool contains the name of the associated target *copy pool backup storage group*. A copy pool backup storage group is a type of SMS storage group that contains eligible target volumes that DFMSHsm can select for fast replication backup versions. [Figure 1](#) shows this relationship. [Figure 1. Pool Storage Groups and Copy Pool Backup Storage](#)



After you define a copy pool and a copy pool backup storage group, you can use DFMSHsm commands to prepare, create, recover, and delete fast replication backup versions. See [Defining storage groups](#) to learn how to define a copy pool backup storage group.

This topic shows you how to define a copy pool. Before you begin:For information about using DFMSHsm commands to perform fast replication functions, see [z/OS DFMSHsm Storage Administration](#).For information about how DFMSdss supports DFMSHsm fast replication functions, see [z/OS DFMSdss Storage Administration](#). .For information about FlashCopy and SnapShot, see [z/OS DFSMS Advanced Copy Services](#).

The following table shows the subtasks and associated procedures you perform in order to use DFMSHsm fast replication commands with copy pools.

- [Planning a copy pool](#)
- [Defining a copy pool](#)
- [Defining a copy pool backup storage group](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Planning a copy pool

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Identify the storage groups for which you want DFSMShsm to manage the fast replication backups. Then determine which of those storage groups you want to put into the same copy pool. Consider these recommendations: If a single storage group contains data that you want to assign to separate copy pools, you must place the data into separate storage groups. DFSMS does not verify that the storage groups listed in the copy pool include any associated extend and overflow storage groups. Therefore, if you have implemented extend or overflow storage groups, ensure you include them in the appropriate copy pools. A pool storage group defined to a copy pool must contain FlashCopy® volumes, SnapShot volumes, or any combination of the two. Determine whether auto dump should be enabled for any of the copy pools. If so, you must determine: Which host, if any, the copy pool has been defined on (auto dump affinity) Whether to adjust the auto dump window to allow for the new workload Whether any storage groups in a copy pool should be processed by auto dump separately from the copy pool. In this case, you might need to define new dump classes to better separate copy pool dumps from non-copy pool dumps.

For DFSMShsm fast replication recovery processing to recover a deleted or moved data set, user catalogs must exist on volumes that are defined in storage groups that are defined in the copy pool definition. In addition, give careful consideration to the ALLOWPPRCP settings in the copy pool. For more information, see *z/OS DFSMShsm Storage Administration*.

Parent topic:[Defining copy pools](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining a copy pool

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use ISMF to define, alter, list, or display copy pool attributes. This topic describes the ISMF panels that you use to define copy pool attributes, and outlines the steps you follow to define a copy pool.

[Steps for defining a copy pool](#)

Parent topic:[Defining copy pools](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Steps for defining a copy pool

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Read the information in [Planning a copy pool](#). Back up the SMS Source Control Data Set (SCDS).

Perform the following steps to define a copy pool.

Select option P, Copy Pool, on the ISMF Primary Option Menu for Storage Administrators. This displays the Copy Pool Application Selection panel.

Supply values on the Copy Pool Application Selection panel:

CDS Name Specifies the name of the SCDS where the copy pool is located. Valid values include: A data set name that follows TSO naming conventions. The quoted word 'ACTIVE', which specifies the currently active configuration. The CDS Name field is required. There is no default.

Copy Pool Name Specifies the name of the copy pool. The maximum length is 23 alphanumeric or special characters. The first character cannot be numeric. The Copy Pool Name field is required. There is no default.

Guideline: For DB2®, use the required DB2 naming convention when you name the copy pool.

Option Select option 3, Define. This displays the Copy Pool Define panel.

Supply values on the Copy Pool Define panel. The SCDS Name and Copy Pool Name fields are primed by ISMF with the values that are specified on the Copy Pool Application Selection panel.

Description Describes the copy pool. You can use up to 120 characters. Description is optional.

Auto Dump Specifies with Y or N whether volumes in this copy pool are to be eligible for automatic dump processing. Auto Dump is a required field. The default is N.

Dump Sys/Sys Group Name Specifies the 1-8 character name of the system or system group where volumes in this copy pool are to automatically dump to backup volumes (the auto dump affinity). Dump Sys/Sys Group Name is an optional field.

Dump Class Specifies the 1-8 character names of up to five dump classes. ISMF does no validity checking of the values you enter in these fields. Dump Class is an optional field.

Number of DASD Fast Replication Backup Versions with Background

Copy Specifies the number of fast replication backup versions of the copy pool that you want to be maintained by DFSMSHsm. Valid values range from 0 to 85. You can leave the field blank. If you specify 0 (zero), DFSMSHsm creates the DASD backup copy with the NOCOPY option. Number of DASD Fast Replication Backup Versions with Background Copy is an optional field. The default is 2.

Each backup version that you specify requires a unique target volume for each source volume. Target volumes are defined in the copy pool backup storage group. For more information, see [Defining a copy pool backup storage group](#).

Recommendation: Specify a minimum of two backup versions.

FRBACKUP to PPRC Primary Volumes allowed Specifies whether DFSMSHsm will target PPRC primary volumes, if available, for FRBACKUP processing. The values are:

NO Do not use PPRC primary volumes as FlashCopy® target volumes. This is the default.

PN PPRC primary volumes can be FlashCopy target volumes. If the target volume is a PPRC primary device, do not consider Preserve Mirror when determining FlashCopy eligibility and when performing fast replication backup.

PP If the target volume is a PPRC primary volume, a Preserve Mirror operation is preferred.

PR If the target volume is a PPRC primary volume, a Preserve Mirror operation is required.

blank Is the same as NO. For more information about FlashCopy, Preserve Mirror, PPRC, also known as synchronous Peer-to-Peer Remote Copy (PPRC), and other copy services functions, refer to z/OS DFSMS Advanced Copy Services.

For more information, refer to the topic about [using Metro Mirror primary volume during fast replication backup](#) in z/OS DFSMSHsm Storage Administration

FRRECOV to PPRC Primary Volumes allowedSpecifies whether DFSMSHsm will target PPRC primary volumes, if available, for FRRECOV processing. The values are: **N**Do not use PPRC primary volumes as FlashCopy target volumes. This is the default.**P**PPRC primary volumes can be FlashCopy target volumes. If the FlashCopy target volume is a PPRC primary device, do not consider Preserve Mirror when performing fast replication recovery."**P**If the FlashCopy target volume is a PPRC primary volume, a Preserve Mirror operation is preferred.**R**If the FlashCopy target volume is a PPRC primary volume, a Preserve Mirror operation is required.**blank**Is the same as NO. For more information about FlashCopy, Preserve Mirror, PPRC and other copy services functions, see z/OS DFSMS Advanced Copy Services.

For more information, refer to the topic about [Using Metro Mirror primary volume during fast replication recovery](#) in z/OS DFSMSHsm Storage Administration

» **FlashCopy Consistency Group** ◊ Specifies whether DFSMSHsm will use consistency groups for FlashCopy. The values are: **N (No)**Do not use consistency groups for FlashCopy. This is the default.**Y (Yes)**Use consistency groups for FlashCopy, to create data-consistent copies. **blank**Is the same as N (No).For more information, refer to the topic about [Creating consistent copies using FlashCopy consistency groups](#) in z/OS DFSMSHsm Storage Administration

«

Use the DOWN command to view page 2 of the panel.

Specify the values on page 2: **Catalog Name**Specifies the names of one or more valid catalogs (up to 10). Use fully qualified names, without quotation marks.

Catalog Name is required if Capture Catalog Information for Data Set Recovery is R or P. There is no default.**Capture Catalog Information for Data Set**

RecoverySpecifies, to FRBACKUP processing, options for collecting catalog information. The options are: **R** (Required). If not able to capture catalog information, fail the backup version.**P** (Preferred). If not able to capture catalog information, issue a warning and do not fail the backup version.**N** (do not collect catalog information). **N** is the default. Note: It is not uncommon for mature or highly utilized catalogs to contain logical errors. However, the catalog capture function in DFSMSHsm Fast Replication requires the catalogs to be free of all errors to ensure that each cataloged data set can be recovered. If errors are present in a catalog when capture catalog information has been requested, message ARC1812I, indicating a catalog failure, may be issued. If the Required option was specified, the fast replication backup request will fail. To allow fast replication backup to succeed, you should correct all catalog errors. DFSMSHsm uses the Catalog Search Interface (CSI) to capture the catalog information. Thus, if the errors in a catalog cause CSI to return with an error, DFSMSHsm issues an ARC1812I message. To allow Fast Replication Backup to succeed, you should correct all catalog errors. Alternatively, specify the Preferred option, which will fail the capture catalog function but allow the FRBACKUP request to continue processing.

Allow Fast Reverse RestoreIndicates whether to allow recovery of a FlashCopy source from the FlashCopy target without waiting for the background copy to complete. The options are: **Y** (Yes). Allow fast reverse restore.**N** (No). Do not use fast reverse restore. This is the default.Use the DOWN command to view page 3 of the panel.

Specify the names of one or more valid pool storage groups. You can specify up to 256 valid pool storage group names in the fields that appear on pages 3, 4 and 5 of the Copy Pool Define panel. Pages 4 and 5 of the panel are not shown here. Use

the DOWN command to display them. Storage Group Names is a required field. You must specify at least one storage group name. There is no default. Rule: You must specify all associated extend and overflow storage groups to ensure that they are included in the copy pool.

Use the END command to save and exit the panel. Result: When you are done, you have defined a new copy pool.

Parent topic:[Defining a copy pool](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining a copy pool backup storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

After you define a copy pool, you must define a copy pool backup storage group. A copy pool backup storage group contains the target volumes for fast replication backup versions. For information on how to use ISMF to define a copy pool backup storage group, see [Defining storage groups](#).

Each copy pool backup storage group must be associated with a pool storage group. You must specify the name of the copy pool backup storage group on the Pool Storage Group Define/Alter panel.

You can use the BACKUPSTORAGEGROUP parameter to override the copy pool backup storage group with an alternate storage group. You can specify this optional keyword with the EXECUTE and PREPARE keywords of the FRBACKUP command for DFMSMShsm: Requesting a Fast Replication Backup or Dump Version. For more information, see [z/OS DFMSMShsm Storage Administration](#).

Parent topic:[Defining copy pools](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This topic documents intended Programming Interfaces that allow you to write programs to obtain the services of DFSMS.

This topic helps you define ACS routines. Specifically, it explains how you can write ACS routines for an SMS configuration using the ISMF Automatic Class Selection application.

[Understanding ACS routines](#)

[Creating ACS routines](#)

[Translating ACS routines](#)

[Validating ACS routines or an entire SCDS](#)

[Translating and validating in a sysplex environment](#)

[Testing ACS routines](#)

[ACS routine environments](#)

[Displaying ACS object information](#)

[Deleting an ACS object from an SCDS](#)

[Using security labels in ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Understanding ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACS routines can be used to determine the SMS classes and storage groups for data sets and objects in an SMS complex. For storage administrators, ACS routines automate and centralize the process of determining SMS classes and storage groups. ACS routines also help convert data sets to an SMS environment. An object is assigned to a storage group when it is stored and remains in that storage group throughout its lifetime. The initial storage class and management class might be determined by defaults defined by an ACS routine or by explicit request. Storage class and management class assignments might be changed by the OSREQ CHANGE function or by automatic class transition. The OSREQ CHANGE request causes invocation of ACS routines that might override the requested assignments. During the automatic class transition, ACS routines are invoked to determine the new storage class and management class assignments. When a data set is allocated, it is assigned an initial management class, storage class, and storage group by defaults defined by an ACS routine or by explicit request. These assignments can be changed through the DFSMSHsm class transition processing that occurs during the automatic space management functions. During the automatic class transition, ACS routines are invoked with the ACS environment of SPMGCLTR to determine the new management class, storage class, and storage group assignments.

[Writing ACS routines](#) lists the rules for programming in the ACS language.

Tip: You can use the DFSMS NaviQuest tool to help you design and test your ACS routines. First, you can create test cases to perform extensive testing against test data representing actual data sets. Then you can test ACS routines in batch, freeing the workstation for other work. See [Using NaviQuest](#) for further information. Through ISMF, you can create and maintain as many as four ACS routines in an SCDS, one for each type of SMS class and one for storage groups. After you have activated an SMS configuration, SMS executes ACS routines for the following operations: JCL DD statements (DISP=NEW, DISP=MOD)Dynamic allocation requests (DISP=NEW, DISP=MOD for a nonexistent data set)DFSMSdss COPY, RESTORE, and CONVERTV commandsDFSMSHsm RECALL, RECOVER, and class transitionsAccess method services ALLOCATE, DEFINE, and IMPORT commandsOAM processing for STORE, CHANGE, and class transitionLocal data set creation by remote application through Distributed FileManager/MVSMVS data sets or z/OS® UNIX System Service (z/OS UNIX) files created by remote application through the z/OS Network File System server

As a storage administrator, you write ACS routines using the ACS programming language, a high-level programming language. The language follows a logical, procedural flow of implementation that consists mainly of filtering criteria, IF/THEN statements, and SELECT/WHEN statements. Using these relational statements, ACS routines determine SMS classes and storage groups according to allocation parameters, data set sizes, object or data set names, and other variables.

All allocations directed to units that are neither tape nor DASD should be excluded from SMS management. Do this by testing for UNIT in the storage class routine and ensuring that the storage class is set to NULL in these cases.

Ensuring that no storage class is assigned for such allocations avoids potential errors with allocations that require specific types of units. For example, assigning a storage class to a VTAM® channel-to-channel (CTC) adaptor allocation results in sense errors when VTAM attempts to use the CTC.

Requirements: For system-managed data sets, the storage group is required because there is no way to explicitly specify storage groups. The other routines are optional for system-managed data sets. For objects, the storage group, storage class and management class ACS routines are required. For tape, the storage group, storage class and data class ACS routines are required.

[Using ACS routines for data sets created by z/OS Network File System and](#)

Using ACS routines for data sets created by z/OS Network File System and
Distributed FileManager/MVS

Restrictions on using ACS routines

Parent topic:Defining ACS routines



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using ACS routines for data sets created by z/OS Network File System and Distributed FileManager/MVS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACS routines can be used to determine the SMS classes for MVS data sets and z/OS® UNIX files created by z/OS Network File System. When data sets and files are created by z/OS Network File System, any attributes related to SMS classes not specified by remote client user are defaulted using the ACS routines for the data set. If the remote user does not specify a storage class and if the ACS routines decide that the data set should not be SMS-managed, Distributed FileManager/MVS ends and returns an error to the remote workstation. Therefore, it is important that on the MVS target system, you consider the potential data set creation requests of remote users when constructing the ACS routines.

When data sets are created by Distributed FileManager/MVS, any attributes not specified by the source requester are defaulted using the attributes specified in the data and management classes selected by the ACS routines for the data set. If the system requires attributes that neither the remote user nor the data class specified, the creation process fails.

See [Determining Distributed FileManager/MVS data set creation requests](#) for information on using ACS routines to determine the SMS classes for data sets created by Distributed FileManager/MVS.

Parent topic:[Understanding ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Restrictions on using ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACS routines are not invoked for of data sets that cannot be system-managed. These include the following: Dummy data setsData sets that have SUBSYS or PATH coded on the JCL DD statementSYSIN and SYSOUT data setsVVDS data sets: SYS1.VVDS.*VTOC Index data sets: SYS1.VTOCIX.*

Parent topic:[Understanding ACS routines](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Creating ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: You must allocate either a sequential data set, a member of a partitioned data set (PDS), or a member of a partitioned data set extended (PDSE) for each of the ACS routines that you intend to write. As a general guideline, use an LRECL of 80. The maximum LRECL you can use is the maximum that can be specified on JCL.

Perform the following steps to create ACS routines.

Select option 7, Automatic Class Selection, from the ISMF Primary Option Menu. This displays the ACS Application Selection panel.

Select option 1, Edit, on the ACS Application Selection panel. This displays the Edit Entry panel.

Supply the required values on the Edit Entry panel, then press Enter to invoke the ISPF Editor.

Supply the source code for the ACS routine.

Use the END command to save the ACS routine and return to the ISPF Edit Entry panel.

Use the END command again to return to the ACS Application Selection panel.

Parent topic:[Defining ACS routines](#)



Translating ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

After creating an ACS routine, you must translate it into executable form. The translation process checks your source code for syntactic and semantic errors, generates an object form if no errors exist, and places the object form into the SCDS you specified on the translate panel. If the ACS routine that you are translating already exists in the SCDS, the new object form replaces the existing object form.

Before you begin: You must display the ACS Application Selection panel. See [Creating ACS routines](#).

Perform the following steps to translate ACS routines.

Select option 2, Translate, from the ACS Application Selection panel. This displays the Translate ACS Routines panel.

Supply values on the Translate ACS Routines panel: **SCDS Name** is the name of an SCDS. A successful translation places the ACS routine object table into this SCDS. **SCDS Name** is primed with the last SCDS that you have referenced or translated into. **ACS Source Data Set** is the name of the data set containing the ACS routine that you want to translate. This is a required field that is primed with the last used value. **ACS Source Member** is the name of the member of the source data set that contains the ACS routine. It is required only if the ACS source data set is a partitioned data set or PDSE. The field is primed with the last used value. The default is blanks. **Listing Data Set** is the name of a sequential data set to contain the translation results in Listing Data Set. If you specify a data set that already exists, the translation process replaces the existing data set contents with the results of the translation. If you specify a nonexistent data set, the translation process allocates space for it. If you leave this field blank, which is the default, you receive the results of the validation but you do not get a listing.

[Browsing the results of a translation](#)

[Browsing the results of an unsuccessful translation](#)

Parent topic: [Defining ACS routines](#)



Browsing the results of a translation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you specify an ACS Source Data Set that contains no errors, you see a PDF browse panel after translation.

The Browse panel displays the SCDS name, the ACS routine source data set name, an output listing of the ACS routine, and a translation return code. SCDS Name, ACS Source Data Set, and ACS Source Member reflect the values that you specified on the Translate ACS Routines panel. The output listing contains the source code of the ACS routine and diagnostic messages. See *z/OS® MVS System Messages, Volumes 1–10* for an explanation of the diagnostic messages.

The Translation Return Code displays one of the following codes: **0000**Successful translation.**0012**Unsuccessful translation. The ACS routine contains one or more semantic or syntactic errors. No object form was created and no updates were made to the SCDS.**0020**Internal error in translator.

When you exit from this panel, you see the Output Listing Disposition panel. The short message area displays the results of the translation.

Supply these values on the Output Listing Disposition panel: **Print Output Listing**Specify Y (Yes) to submit a batch job to print the data set. The default is N (No).**Delete Output Listing**Specify Y (Yes) to delete the data set after it is printed. The default is N (No).**Table 1** summarizes the possible outcomes for each combination of Output Listing Disposition values.

Table 1. Results of Output Listing Disposition

Use the END command to return to the Translate ACS Routines panel.

Parent topic:[Translating ACS routines](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Browsing the results of an unsuccessful translation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you specified an ACS source data set that contains errors, and you specified a data set in the Listing Data Set field, the Browse panel shown in [Figure 1](#) is displayed after translation. [Figure 1. Browsing the Results of an Unsuccessful ACS Translation](#)

```
BROWSE -- STORGRP.LISTING----- LINE 00000000 COL 001 080
COMMAND ===> SCROLL ===> PAGE
***** TOP OF DATA *****
ACS TRANSLATOR ***** TIME 11:47:09 DATE 07/15/2004 PAGE 0001 *****
SCDS NAME: 'SMS.TEST.SCDS'
ACS SOURCE DATA SET: SMS.ACS.ROUTINES
ACS SOURCE MEMBER: SG1

0001 PROC STORGRP
0002   SELECT
0003     WHEN (&STORCLAS = DB*)
0004       SET &STORGRP = 'DATABASE'
0005     WHEN (&MAXSIZE > 9999999KB)
0006       SET &STORGRP = 'LARGE'
0007     WHEN &DSTYPE = 'TEMP'
***** IGD03114I A LEFT PARENTHESIS WAS EXPECTED BUT NOT FOUND
0008       SET &STORGRP = 'VIO', 'PRIMARY'
0009     OTHERWISE
0010       SET &STORGRP = 'PRIMARY'
0011   END
0012 END

SUMMARY OF ERROR MESSAGES:

0007 IGD03114I A LEFT PARENTHESIS WAS EXPECTED BUT NOT FOUND

TRANSLATION RETURN CODE: 0012
***** BOTTOM OF DATA *****
```

Scroll down to see a display of all the error messages together.

Parent topic:[Translating ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Validating ACS routines or an entire SCDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SMS validates the entire SCDS when it is saved. You can validate the individual ACS routines of an SMS configuration after successfully translating them. You should also validate the entire configuration yourself so that you can see any error messages that result. Separate validation of the ACS routines does not produce all of the possible messages.

[Validating an ACS routine](#)

[Validating an entire SCDS](#)

Parent topic:[Defining ACS routines](#)

|

©Copyright IBM Corporation 1990, 2014

Validating an ACS routine

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Display the ACS Application Selection panel. See [Creating ACS routines](#).

Perform the following steps to validate an ACS routine.

Select option 3, Validate, on the ACS Application Selection panel. This displays the Validate ACS Routines or Entire SCDS panel.

Supply values on the Validate ACS Routines or Entire SCDS panel: **SCDS**

NameIdentifies the SCDS containing the ACS routine you are validating. It is a required field that is primed with the last used value. The default is blank.**ACS**

Routine TypeIdentifies which of the four ACS routine types you are validating.

When validating an individual ACS routine, the valid values are: DC, SC, MC, and SG. It is a required field that is primed with the last used value. The default is an asterisk, *, which specifies that you want to validate the entire SMS configuration in the SCDS, and not just one of the ACS routines.**Listing Data Set**Identifies the name of a sequential data set to contain the results of the validation. If you specify a data set that already exists, the validation process replaces the existing data set contents with the results of the validation. If you specify a nonexistent data set, the validation process allocates space for it. If you leave this field blank, which is the default, you receive the result of the validation, but you do not get a listing.

Press Enter to perform the validation. See [Validation results](#) for more information.

[Validation results](#)

[Browsing validation results](#)

Parent topic: [Validating ACS routines or an entire SCDS](#)



Validation results

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Validation fails if certain conditions are not satisfied, as described below (except when, as noted, only a warning is issued). **For the storage group ACS routine**, the conditions are: All of the defined pool storage groups should be possible outcomes of the routine. No dummy storage group can result from the storage group ACS routine. All storage groups that are possible outcomes of the storage group ACS routine must exist (be defined). All VIO storage groups are possible results of the storage group ACS routine. Note: If you have no pool storage groups, you get a warning. If you have any pool type storage groups that are not possible outcomes of the storage groups ACS routine (that is, are not used in any SET statement), you get a warning. This is because the storage group cannot be explicitly requested and therefore represents a wasted resource.

For the management class ACS routine: All management classes that are possible outcomes of the management class ACS routine must exist. **For the storage class ACS routine:** All storage classes that are possible outcomes of the storage class ACS routine must exist. **For the data class ACS routine:** All data classes that are possible outcomes of the data class ACS routine must exist.

If an ACS routine uses the SET statement to assign an SMS class or storage group that does not exist, validation fails. If an ACS routine references an SMS storage class, management class, or data class that does not exist, SMS issues a warning. For example, you might delete a storage class. If you continue to reference the deleted storage class in the storage class ACS routine of an SCDS, SMS issues a warning when you validate the SCDS. If only warnings exist, the SCDS becomes valid despite the warning. This allows you to check for the deleted storage class and replace it with an existing storage class within the storage class ACS routine.

For example:

```
WHEN (&STORCLAS = 'OLDSC')
      SET &STORCLAS = 'NEWSC'
```

Parent topic: [Validating an ACS routine](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Browsing validation results

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you specified a data set name in the Listing Data Set field, you get the Browse panel shown in [Figure 1](#) after translation.

Figure 1. Browsing the Results of an ACS Routine Validation

```
BROWSE -- SGVAL.LISTING -----
----- LINE 0000000 COL 001 080
COMMAND ===> SCROLL ===> PAGE
***** TOP OF DATA *****
VALIDATION RESULTS

VALIDATION RESULT: ERRORS DETECTED
SCDS NAME: 'SMS.SCDS1.SCDS'

ACS ROUTINE TYPE: SG
DATE OF VALIDATION: 2004/11/19
TIME OF VALIDATION: 10:01

IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE DATABASE
IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE LARGE
IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE VIO
IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE PRIMARY
IGD06025I THE STORAGE GROUP ACS ROUTINE SETS NON-EXISTENT VALUE PRIMARY
***** BOTTOM OF DATA *****
```

VALIDATION RESULT can be VALIDATION SUCCESSFUL, ERRORS DETECTED or WARNINGS DETECTED. The SCDS Name and ACS Routine Type fields reflect the values you specified on the Validate ACS Routines or Entire SCDS panel. Browse also provides the date and time of validation. See [z/OS® MVS System Messages, Volumes 1–10](#) for an explanation of the diagnostic messages. When you leave Browse, you receive the Output Listing Disposition panel. See [Browsing the results of a translation](#) and [Table 1](#) for information about printing and deleting.

Parent topic: [Validating an ACS routine](#)



©Copyright IBM Corporation 1990, 2014

Validating an entire SCDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For its contents to become the active storage management policy for an installation, an SCDS must be valid. Activating an SCDS validates its contents and copies them into the ACDS identified by IGDSMSxx. If the SCDS is not valid, activation fails.

To define a valid minimal SMS configuration, the configuration must contain: A fully defined base configurationA storage class definitionA pool storage group containing at least one volume, or a VIO, object, or tape storage groupA storage class ACS routineA storage group ACS routine See z/OS DFSMS Implementing System-Managed Storage for more information on defining a minimal SMS configuration.

Before you begin: Display the CDS Application Selection panel (select option 8 from the ISMF Primary Option Menu for Storage Administrators.)

Perform the following steps to validate an entire SCDS.

On the command line of the CDS Application Selection panel, type the VALIDATE command. This displays the Validate ACS Routines or Entire SCDS panel.

Supply values on the Validate ACS Routines or Entire SCDS panel: **SCDS**

NameIdentifies the data set you want to validate. It is a required field that is primed with the last used value. The default is blank.

ACS Routine TypeType an * in this field, because you are validating an entire SCDS.

Listing Data SetIdentifies the name of a sequential data set to contain the results of the validation. If you specify a data set that already exists, the validation process replaces the existing data set contents with the results of the validation. If you specify a nonexistent data set, the validation process allocates space for it. If you leave this field blank, which is the default, you receive the result of the validation, but you do not get a listing.

Press Enter to perform the validation. See [Validation results](#) for more information.

[Validation results](#)

Parent topic: [Validating ACS routines or an entire SCDS](#)



Validation results

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Validation fails if any of the following conditions are not satisfied (except where it is noted that only a warning is issued). Base configuration information must exist. If you specify a default management class, you need to have defined the management class. Each possible outcome of the data class, storage class, management class, and storage group ACS routines must have a corresponding definition in the SCDS. For example, if you have five unique storage classes that can be determined by the storage class ACS routine, then you must have five corresponding storage class definitions in the SCDS. All classes and storage groups referenced in all ACS routines exist (issues a warning only). The storage group ACS routine exists. At least one VIO, pool, object type storage or tape group exists. All pool and dummy storage groups have at least one volume. At least one storage class exists. All pool and VIO storage groups are possible outcomes of the storage group ACS routine (issues a warning only). No dummy storage group is set by storage group ACS. All libraries associated with storage groups must exist in the configuration. Every optical library must have at least one drive associated with it. The DATABASE 2 (DB2®) qualifier specified for each object or object backup storage group must be unique. Object and object backup storage groups can have up to eight real or pseudo optical libraries, but not both. An object or object backup storage group cannot reference a tape library. A tape storage group cannot reference an optical library. A tape storage group must have at least one tape library. If a default entry data class is specified, the data class must exist in the configuration. All cache sets specified in storage classes must be defined in the base configuration.

Parent topic:[Validating an entire SCDS](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Translating and validating in a sysplex environment

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When running SMS in a sysplex with mixed levels of DFSMS, you must do the following on the system running the highest level of DFSMS: Define new SMS constructs and ACS routines Modify existing SMS constructs and ACS routines Translate your ACS routines Validate your SCDS See [Altering the SCDS on different DFSMS releases](#) for further information on altering the SCDS.

You can then activate and share the validated SCDS among systems with mixed levels of DFSMS. If you do not translate your ACS routines and validate your SCDS on the highest level of DFSMS, the translation and validation might fail. This failure occurs because some read-only variables are only known to higher levels of DFSMS. It can also occur due to changes in validation rules between releases.

Guideline: Coexistence PTFs allow you to share CDSs among mixed DFSMS levels and switch back to a prior release. However, once the CDSs have been formatted by the higher-level system, the CDS control blocks reflect the new rather than the down-level lengths.

When sharing the same SMS CDSs among mixed releases, or when switching back to a lower-level release using the same SMS CDSs that were formatted by the higher-level system, all coexistence PTFs *must* be applied to avoid corrupting the CDSs.

Parent topic:[Defining ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Testing ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

After completing your ACS routines, you can write and execute test cases using the ISMF Automatic Class Selection application. After testing the individual routines of a new or modified configuration, you can activate it with greater confidence. ACS installation exits are not invoked during ACS routine testing. Restriction: During ACS testing, only those parameters that are passed to the ACS routines at ACS time are tested. For some data processing, what is specified in the JCL is not necessarily what is passed to the ACS routines. A good example of this is how the RETPD and EXPDT parameters work: if EXPDT is specified in the JCL, during normal processing the RETPD read-only variable is set based on the EXPDT. If your ACS routine logic bases decisions on RETPD variable, and you specify EXPDT as a test parameter, you might get unexpected test results. ACS testing is not designed to simulate all of the possible processing that might occur before ACS processing. You must take care to know what is passed to the ACS routines at ACS time to ensure that you are testing what you expect to occur on your system.

[ACS test usage](#)

[Creating ACS test cases](#)

[Running ACS test cases](#)

[ACS routines invoked for copying and importing data sets](#)

[ACS routines invoked for restoring, recalling, recovering, and converting data sets](#)

Parent topic:[Defining ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ACS test usage

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Because a DSNTYPE of EXT cannot be specified in JCL, ACS test for an data set allocated in extended format has some considerations.

To test data class only, you should not specify DSNTYPE = EXT for extended format data sets, because the value of DSNTYPE in the Data Class ACS routine is never set to EXT during actual ACS processing.

To test more than just the Data Class for extended format data sets, you can choose to test the Data Class ACS routine with a separate test case that has no DSNTYPE specified. To test other constructs, such as management class, storage class and storage group, you should use a test case with DSNTYPE set to EXT. Note: ACS read-only variables are not defaulted from Data Class when the ACS test case is run (unlike actual ACS processing).

Parent topic:[Testing ACS routines](#)



©Copyright IBM Corporation 1990, 2014

Creating ACS test cases

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Allocate a partitioned data set for the ACS test cases. The partitioned data set must have a logical record length of 80 or greater and a record format of F or FB. Then, display the ACS Application Selection panel. See [Creating ACS routines](#).

Perform the following steps to create ACS test cases.

Select option 4, Test, from the ACS Application Selection panel. This displays the ACS Test Selection panel.

Supply values on the ACS Test Selection panel: **ACS Test Library and ACS Test Member** These identify the data set and member that contain the test case. You can specify one test case per member. **Option** Select option 1, Define, to specify test criteria. This displays the ACS Test Case Define panel.

Supply values on the ACS Test Case Define panel, which has multiple pages. On these pages, you can specify test values that correspond to the ACS variables listed in [Writing ACS routines](#). On page 3, you can specify values corresponding to the MSxxxx parameters that you expect to be returned from your tape management system through the pre-ACS routine. See [Tape management system support](#) for a description of these parameters, and use the documentation provided by your tape management vendor for more specifics.

Use the END command to save your test values and return to the ACS Test Selection panel. On the ACS Test Selection panel, you can add more test case members to the same test library, or you can build another test library.

Parent topic: [Testing ACS routines](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Running ACS test cases

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to run ACS test cases.

Display the ACS Test Selection panel, as described in [Creating ACS test cases](#).

Supply values on the ACS Test Selection panel: **Option**Select option 3, Test.**ACS Test Library**Supply the library name.Press Enter to see the Test ACS Routines panel.

Supply values on the Test ACS Routines panel to indicate which routines you want to test. **ACS Test Library and Member**These identify the data set and member that contain the test case. Specify an asterisk in the Member field to run all test cases in the partitioned data set library. **Listing Data Set**Specify a data set name to create a listing of the results of the testing. Leave this field blank to prevent the creation of a listing.

Press Enter to test the routines. If you specified a listing data set, the results of the testing are displayed in the PDF Browse panel. After examining the results, issue the END command and specify on the ACS Output Listing Disposition whether or not to keep the output listing.

Parent topic:[Testing ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ACS routines invoked for copying and importing data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Table 1 shows which ACS routines are invoked when performing initial allocation, importing, copying, restoring and recalling data sets.

Table 1. Allocation, IMPORT, and COPY Conditions

[Initial allocation](#)

[IMPORT \(Access Method Services\)](#)

[COPY \(DFSMSSdss\)](#)

[COPY BYPASSACS \(DFSMSSdss\)](#)

Parent topic: [Testing ACS routines](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Initial allocation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Because the data set is new in an initial allocation, the data class ACS routine is invoked to determine the characteristics of the data set. The storage class ACS routine is then invoked to determine whether the data set is system-managed. The management class ACS and storage group ACS routines are invoked only if the data set is system-managed and a storage class was assigned. Each time a dynamic allocation is issued by application programs, such as DFSMSHsm RECALL, SMS follows this initial allocation path (except for the data class ACS routine which is not invoked).

Parent topic:[ACS routines invoked for copying and importing data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

IMPORT (Access Method Services)

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The IMPORT (IDCAMS) command has three conditions: The VSAM data set already existsThe VSAM data set is preallocated (allocated but empty)The target data set is created by the IDCAMS IMPORT job
The data is copied from a backup copy or another VSAM data set, and is imported to the target data set. If a storage class is assigned, the data is then copied into a system-managed target data set. Otherwise it is copied into a non-system-managed target data set. The data class ACS routine is not invoked, because either the data set already exists, or the characteristics of the data set are derived from the IDCAMS IMPORT job.
The storage class ACS routine is invoked to determine whether the data set is system-managed. The imported data might come from a non-system-managed source data set that was then copied into a system-managed target data set.
The management class ACS and storage group ACS routines are invoked only if the data set is system-managed and a storage class was assigned.

Parent topic:[ACS routines invoked for copying and importing data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

COPY (DFSMSdss)

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The COPY command has three conditions: The target data set already existsThe target data is preallocated (allocated but empty)The target data is created by the DFSMSdss COPY job

The data is either copied from a backup copy of the data set, or from another data set to the target data set.

The data class ACS routine is not invoked, because either the data set already exists, or the characteristics of the data set are derived from the DFSMSdss COPY job.

The storage class ACS routine is invoked to determine whether the data set is system-managed. The copied data might come from a non-system-managed source data set that was then copied to a system-managed target data set.

The management class ACS and storage group ACS routines are invoked only if the data set is system-managed.

See *z/OS DFSMSdss Storage Administration* for more information on the DFSMSdss COPY command.

Parent topic:[ACS routines invoked for copying and importing data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

COPY BYPASSACS (DFSMSdss)

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The COPY BYPASSACS command has three conditions: The target data set already existsThe target data is preallocated (allocated but empty)The target data is created by the DFSMSdss COPY job

The data is copied from either a backup copy or another data set to the target data set. If a storage class is assigned, the data is then copied into a system-managed target data set. Otherwise it is copied into a non-system-managed target data set. The data class, storage class, and management class ACS routines are not invoked because the BYPASSACS keyword is coded in the DFSMSdss COPY job. The storage class ACS routine is invoked only if it is specified in the DFSMSdss COPY job.

The storage group ACS routine is invoked only if SC is specified in the DFSMSdss COPY job, or if the data set is system-managed.

Requirement: You need RACF® authorization to use the BYPASSACS keyword.

The storage or security administrator must define BYPASSACS to RACF as a facility and then tell RACF who is authorized to use it.

Parent topic:[ACS routines invoked for copying and importing data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ACS routines invoked for restoring, recalling, recovering, and converting data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

[Table 1](#) shows which ACS routines are invoked for restoring, recalling, recovering, and converting data sets.

Table 1. RESTORE, RECALL, RECOVER, CONVERTV, and FORCENONSMS Conditions

[DFSMSdss RESTORE](#)
[DFSMSdss RESTORE BYPASSACS](#)
[DFSMSdss CONVERTV](#)
[DFSMShsm RECALL/RECOVER](#)
[DFSMShsm Class Transitions](#)
[DFSMShsm FORCENONSMS](#)
Parent topic: [Testing ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMSdss RESTORE

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When using the DFSMSdss RESTORE command, the data set either exists or is deleted. The data set is recovered by restoring data from a backup copy to the target data set.

The data class ACS routine is not invoked, because either the data set already exists, or the characteristics of the data set are derived from the backup data set or DFSMSdss RESTORE job.

The storage class ACS routine is invoked to determine whether the data set is system-managed.

The management class ACS and storage group ACS routines are invoked only if the data set is system-managed and a storage class is assigned.

Parent topic:[ACS routines invoked for restoring, recalling, recovering, and converting data sets](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMSdss RESTORE BYPASSACS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When using the DFSMSdss RESTORE BYPASSACS command, the data set either exists or has been deleted. The data set is recovered by restoring data from a backup copy to the target data set.

The data class, storage class, and management class ACS routines are not invoked because the BYPASSACS keyword is coded in the DFSMSdss RESTORE job.

The storage group ACS routine is only invoked if: Storage class is specified in the DFSMSdss RESTORE job. The data set is system-managed. Requirement: You need RACF® authorization to use the BYPASSACS keyword. Define BYPASSACS to RACF as a facility and then tell RACF who is authorized to use it.

Parent topic:[ACS routines invoked for restoring, recalling, recovering, and converting data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMSdss CONVERTV

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When using the DFSMSdss CONVERTV command, the data class ACS routine is not invoked because the data sets on the volume already exist.

The storage class ACS routine is invoked because the data sets and volume are to become system-managed.

The management class ACS routine is invoked, if the storage class is assigned, to determine the appropriate management class names assigned to the data sets.

The storage group ACS routine is not invoked because all the data sets on the system-managed volume are assigned to the storage group to which the volume belongs.

Parent topic:[ACS routines invoked for restoring, recalling, recovering, and converting data sets](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMShsm RECALL/RECOVER

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When using the RECALL/RECOVER command, if you want to RECALL a data set, then the data set already exists. If you want to RECOVER a data set, then either the data set exists or has been deleted. The data set is recovered by restoring data from a DFSMShsm backup copy to the target data set.

The data class ACS routine is not invoked, because either the data set already exists, or the characteristics of the data set are derived from the backup copy. The storage class ACS routine is invoked to determine whether the data set is system-managed. The restored data might come from a non-system-managed backup copy, and is allocated as system-managed or non-system-managed as determined by the ACS routines.

The management class ACS and storage group ACS routines are invoked only if the data set is system-managed.

Parent topic:[ACS routines invoked for restoring, recalling, recovering, and converting data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMShsm Class Transitions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before recalling a data set, DFSMShsm will check if a class transition was missed while that data set was migrating. If a class transition was missed, then DFSMShsm will run the ACS routines with ACS environment SPMGCLTR to determine any new SMS construct names. Using the newly retrieved management class name, DFSMShsm will repeat the check to see if a class transition was missed, until the appropriate management and storage classes and storage group(s) are finally found. These new management class, storage class, and storage group(s) are then used for the RECALL.

Note: To avoid circularity, DFSMShsm will repeat the check to see if a class transition was missed five times. DFSMShsm will *not* check if any class transitions were missed during RECOVER, ARECOVER, or FRRECOV of a data set.

Parent topic: [ACS routines invoked for restoring, recalling, recovering, and converting data sets](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMShsm FORCENONSMS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When using the FORCENONSMS keyword, if you want to RECALL a data set, then the data set already exists. If you want to RECOVER a data set, then either the data set exists or is deleted. The data set is recovered by restoring data from a DFMSHsm backup copy to the target data set. Because the data set is not supposed to be system-managed, no ACS routines are invoked.

Parent topic:[ACS routines invoked for restoring, recalling, recovering, and converting data sets](#)



©Copyright IBM Corporation 1990, 2014

ACS routine environments

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Depending on the environment, SMS invokes some or all of the ACS routines in the following order:

Data classStorage classManagement classStorage group DFSMSrmm supports the SMS pre-ACS interface. The SMS subsystem calls DFSMSrmm before the data class ACS routine obtains control. Then, DFSMSrmm optionally sets the initial value for the ACS routine MSPOOL and MSPOLICY read-only variables if the pre-ACS installation exit has not done so. However, DFSMSrmm does not use the installation exit.

See [z/OS DFSMSrmm Implementation and Customization Guide](#) for detailed information on the DFSMSrmm support for the SMS pre-ACS interface. Or, check with your tape management vendors for information on support for the variables. When UNIT=AFF is specified on the JCL DD statement and the ACS routine determines that the AFF'ed and AFF'ing DDs use different SMS constructs, an unexpected SMS construct might be used.

[JCL DD statement \(batch\), and dynamic allocation](#)

[Volume reference](#)

[Data set stacking](#)

[Non-data set stacking allocations](#)

[Tape management system support](#)

[Access Method Services](#)

[DFSMShsm](#)

[DFSMSdss](#)

[DFSMSrmm](#)

[ISMF](#)

[OAM](#)

[Processing of SMS classes and storage groups](#)

Parent topic:[Defining ACS routines](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

JCL DD statement (batch), and dynamic allocation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For the &ACSENVIR='ALLOC' environment, the data class ACS routine and then the storage class ACS routine are executed. If the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed.

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Volume reference

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can code VOL=REF on a DD statement to refer to a DD statement in the same or an earlier step to allocate the data set on the same volume as the earlier data set. You can also code VOL=REF to an existing data set (VOL=REF=A.B.C,...) where A.B.C is a cataloged data set. If VOL=REF is coded, SMS invokes ACS routines as follows: SMS invokes the data class ACS routine. Storage class is copied from the referenced data set if the referenced data set has a storage class assigned to it. However, if the referenced data set is an SMS-managed tape data set and it is not cataloged or DISP is not OLD or MOD, the storage class is not available. In that case, the ACS routine is called and must specify a valid storage class for the referencing data set to be an SMS-managed tape data set. Note: Non-SMS managed data sets do not have storage classes to copy. As a result, the storage class routine is invoked and either allows the non-SMS allocation or fails it. If it does anything else, SMS fails it. When VOL=REF is coded to refer to a DD statement and the referenced DD allocates a new data set (DISP=NEW), the storage class is copied from the referenced DD if one is assigned. When VOL=REF is coded to refer to an SMS-managed tape data set, the storage class from the referenced data set will not be available. The storage class ACS routine is invoked and a valid storage class must be assigned; otherwise, SMS fails it. For example, in the case of VOL=REF to an SMS-managed tape data set, the storage class ACS routine must use the read-only variables &LIBNAME,&ANYVOL,&ALLVOL to derive a valid storage class that will allow this referencing data set to be allocated to SMS-managed tape. Additionally, read-only variable &LABEL can be used to help derive the valid storage class.

The storage class ACS routine is also invoked when VOL=REF and UNIT=AFF is coded on the DD statement. For information on what is passed to the storage class ACS routine, refer to [Non-data set stacking allocations](#). If you use VOL=REF processing to refer to a temporary data set, you might get different results in storage group assignments than expected. This is because temporary data sets are assigned a storage group by the system, based on a list of eligible storage groups, such as: VIO, PRIME, STANDARD, etc. Data sets that use VOL=REF are assigned a storage group based on this list of eligible storage groups, not on the name of the storage group used to successfully allocate the first data set being referenced. This might result in the data sets being allocated in different storage groups.

Management class ACS routine is called if storage class is not null. The storage group from the referenced data set is passed as input to the storage group ACS routine. For data sets on SMS-managed tape volumes, the ACS routine must assign the same storage group to the referencing data set. For other SMS-managed data sets, any pool or VIO storage group can be assigned to the referencing data set.

When you specify VOL=REF in the JCL, the system retrieves the volume serial numbers from the referenced DD. In the case of NEW to NEW referencing, since DD is not allocated yet, the default of 1 is passed in the &NVOL parameter.

When VOL=REF is used, the &ALLVOL and &ANYVOL ACS read-only variables are set to 'REF=SD,' 'REF=ST' or 'REF=NS' as appropriate. Additionally, if the reference is to a data set on an SMS-managed volume, the storage group of the referenced data set is provided in the &STORGRP read-write variable, if it is available. (For some references to data sets on SMS-managed tape, it might not be, in which case the ACS routine should use the value of the &LIBNAME read-only variable instead.) If the referenced data set is new, it can still have multiple candidate storage groups because it has not been allocated yet. In that case only the *first* candidate storage group is passed in.

[Examples: using volume reference](#)

[Parent topic:ACS routine environments](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Examples: using volume reference

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The &ALLVOL and &ANYVOL ACS read-only variables contain the following values when you use VOL=REF: 'REF=SD' (the volume reference is to an SMS-managed DASD or VIO data set)'REF=ST' (the volume reference is to an SMS-managed tape data set) [Figure 1](#) illustrates these values:

Figure 1. Example of REF=ST Values when Using VOL=REF PROC STORGRP

```
SELECT(&ANYVOL)
WHEN('REF=SD')
  IF &DSTYPE = 'TEMP' &&DSORG ^= 'VS' THEN
    SET &STORGRP = 'VIOSG','MAIN3380','MAIN3390','SPIL3380','SPIL3390'
  ELSE
    SET &STORGRP = 'MAIN3380','MAIN3390','SPIL3380','SPIL3390'
WHEN('REF=ST')
  SET &STORGRP = &STORGRP
OTHERWISE
.
.
.
END
END
```

'REF=NS' (the volume references is to a non-SMS-managed data set) [Figure 2](#) illustrates these values:

Figure 2. Example of REF=NS Values when Using VOL=REF PROC 0 STORCLAS

```
FILTLIST AUTH_USER INCLUDE('SYSPROG1','SYSPROG2','STGADMIN','SYSADMIN')

IF &ANYVOL = 'REF=NS' & &HLQ ^= 'SYS1' THEN
  IF &USER ^= &AUTH_USER THEN
    DO
      WRITE 'INVALID USE OF VOL=REF TO A NON-SMS-MANAGED DATA SET'
      WRITE 'AS OF 12/31/04 ALLOCATION WILL BE FAILED FOR ' &DSN
    END
  .
  END
```

As of the date shown in [Figure 2](#), you might change the ACS routine to fail the uses of VOL=REF that are not valid, as shown in [Figure 3](#).

Figure 3. Example of Failing VOL=REF Values That Are Not Valid PROC 1 STORCLAS

```
FILTLIST AUTH_USER INCLUDE('SYSPROG1','SYSPROG2','STGADMIN','SYSADMIN')

IF &ANYVOL = 'REF=NS' & &HLQ ^= 'SYS1' THEN
  IF &USER ^= &AUTH_USER THEN
    DO
      WRITE 'INVALID USE OF VOL=REF TO A NON-SMS-MANAGED DATA SET'
      WRITE 'DATA SET ' &DSN ' MUST BE SMS-MANAGED'
      EXIT CODE(4)
    END
  .
  END
```

Parent topic:[Volume reference](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data set stacking

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Data set stacking is the function used to place several data sets on the same tape volume or set of tape volumes. It increases efficiency when using tape media and reduces the overall number of tape volumes needed by allocation. It also allows an installation to group related data offsite. The data set sequence number subparameter on the JCL LABEL parameter is used in conjunction with VOL=REF or VOL=SER to accomplish this function.

Under certain conditions, SMS invokes the ACS routines more than once. This section describes those conditions as well as the values of the read-only variables for the initial invocation and any subsequent invocations of the ACS routines.

Because data set stacking might cause a second or third invocation of the ACS routines, you might want to take special care when using WRITE statements to avoid duplicates in the job log.

[Using VOL=SER within a job step](#)

[Using VOL=REF](#)

[Possible values for the &UNIT read-only variable](#)

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

Using VOL=SER within a job step

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When data set stacking is specified using VOL=SER within a job step, the system ensures that all the data sets making up the data set collection (a group of data sets intended to be allocated on the same tape volume or set of tape volumes as a result of data set stacking) are directed to the same device category. If the ACS routines initially directed the stacked allocations to different device categories, the system detects this and reinvokes the ACS routines, passing additional information to those routines. The ACS routines can then do one of the following: Correct the problem and route the allocations to consistent device categoriesFail the stacked allocation (if the ACS routine exits with a nonzero return code)Fail to correct the inconsistency, in which case SMS fails the allocationRecommendation: The system cannot detect data set stacking if VOL=SER is used to stack data sets across jobs or job steps. In these instances, you can change your JCL to specify VOL=REF instead of VOL=SER. For more recommendations on when to use VOL=REF versus VOL=SER for data set stacking, see [z/OS MVS JCL User's Guide](#).

The system reinvokes the ACS routines only when all of the following conditions are true: The request is part of a data set collection based on a data-set-sequence-number greater than one specified on the LABEL parameter, and a VOL=SER, where at least one of the volume serial numbers matches one of the volume serial numbers for a previous request in the same step. The request is currently directed to a different device category than the other requests in the data set collection. The request is DISP=NEW (or DISP=MOD treated as NEW).

Parent topic:[Data set stacking](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using VOL=REF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When data set stacking is requested with the VOL=REF parameter, the ACS routines are passed information that indicates that volume reference is used. Therefore, the ACS routines can direct the requests within a data collection to the same device category.

Parent topic:[Data set stacking](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Possible values for the &UNIT read-only variable

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Information is passed to the ACS routines in the &UNIT ACS read-only variable, so that the ACS routines know when data set stacking or unit affinity is used. In a tape environment, unit affinity is a JCL keyword (UNIT=AFF) used to minimize the number of tape drives used in a job step.

[Table 1](#) shows the values to which the &UNIT read-only variable can be set. The first value is applicable only when UNIT=AFF is used. The others are applicable only when data set stacking is being done, whether or not the UNIT=AFF keyword is present.

Table 1. Values for &UNIT ACS Read-Only Variable

[Examples: storage class ACS routines for read-only variables](#)

Parent topic:[Data set stacking](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Examples: storage class ACS routines for read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

[Figure 1](#) shows an example of a storage class ACS routine for read-only variables:

Figure 1. Example of a Storage Class ACS Routine for Read-Only Variables_{PROC}

```
&STORCLAS
SELECT(&UNIT)
  WHEN('STK=SMSD')
    SET &STORCLAS = 'POOLSC'
  WHEN('STK=NSMS')
    SET &STORCLAS =
  OTHERWISE
```

The storage group ACS routine could then do something like what is shown in

Figure 2:

Figure 2. Example of a Storage Group ACS Routine for Read-Only Variables_{PROC}

```
&STORGRP
SELECT(&UNIT)
  WHEN('STK=SMSD')
    SET &STORGRP = 'S1P03'
  WHEN('STK=NSMS')
    SET &STORGRP = 'POOLSG'
  OTHERWISE
```

Parent topic:[Possible values for the &UNIT read-only variable](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Non-data set stacking allocations

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following considerations apply when UNIT=AFF is used to reduce the number of units for a job, instead of data set stacking.

When unit affinity is specified on a DD statement, three new values are set depending on the unit of the AFF'ed DD. The following example explains how these values are set. In this example, DD1 is directed to SMS DASD, and DD2 is directed to SMS tape.

```
//DD1 DD UNIT=SYSDA,DISP=NEW...
//DD2 DD UNIT=AFF=DD1,DISP=NEW, ...
//DD3 DD UNIT=AFF=DD2,DISP=NEW, ...
//DD4 DD UNIT=AFF=DD1,DISP=NEW, ...
```

With the exception of the JES3 environment, ACS routines are called multiple times. When the ACS routines are invoked by the JES3 PRESCAN processing, the &UNIT read-only variable are set to AFF= for DD2, DD3, and DD4. The ACS routines are invoked again later during the allocation process with the values shown in the example.

Table 1 illustrates how &LIBNAME, &STORGRP, &ALLVOL, &ANYVOL, and &STORCLAS are set. These read-only variables are set depending on the value of the AFF'ed DD (for example, DD1 if DD2 is being processed) or the VOLSER value on the AFF'ing DD (DD2).

Table 1. AFFing DD Volser Values

Note: When both the AFFing DD and the AFFed DD are DISP=NEW, the storgrp and the library values will not be available when ACS routines are being processed for the AFFing DD.

[Volume reference and unit affinity](#)

Parent topic:[ACS routine environments](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Volume reference and unit affinity

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When both VOL=REF and UNIT=AFF are specified, VOL=REF processing takes precedence over UNIT=AFF. The &UNIT read-only variable may contain blanks or 'AFF= ', depending on the disposition of the referenced DD. Recommendation: When specifying both VOL=REF and UNIT=AFF, ensure that the ACS routines check for &ANYVOL='REF=ST', 'REF=SD' or 'REF=NS' *before* checking the &UNIT read-only variable. This ensures that the ACS routines select the correct classes.

Unlike volume reference processing, in which the storage class is produced from the REF'ed DD, unit affinity processing allows you to break the affinity by setting a different storage class. Even though the AFF'ed DD's storage class is passed as input to the storage class routine, the storage class routine must set a valid storage class for the job to run successfully.

[Example: storage class ACS routine specifying both volume reference and unit affinity](#)

Parent topic:[Non-data set stacking allocations](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Example: storage class ACS routine specifying both volume reference and unit affinity

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In [Figure 1](#), the storage class routine ensures that even if only UNIT=AFF is specified, the AFF'ed DD storage class will be assigned. Figure 1. Example of a Storage Class ACS Routine Specifying Both Volume Reference and Unit Affinity IF

```
&ANYVOL = 'REF=ST' OR &ANYVOL = 'REF=SD' THEN
  SET &STORCLS = &STORCLS
ELSE
  IF &UNIT = 'AFF=SMST' OR &UNIT = 'AFF=SMSD' THEN
    DO
      IF &STORCLS = '' THEN
        SET &STORCLS = 'xxxxxxxx' /* TO BREAK AFFINITY */
    ELSE
      SET &STORCLS = &STORCLS /* TO HONOR AFFINITY */
    END
  WRITE 'STORCLAS SET TO &STORCLS'
```

Parent topic:[Volume reference and unit affinity](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Tape management system support

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To support your tape management system's need to coordinate complex vaulting requirements with data set allocation in system-managed environments, you can use a pre-ACS routine exit to set values for read-only variables, which SMS then uses as input to ACS routines. This support is useful when the choice of a storage group needs to be influenced by the data set's vaulting requirements. For example, you might use this support when excluding data from redirection under tape mount management or directing data to a specific system-managed library.

The four read-only variables used to support tape management system-driven tape allocations are:

&MSPDEST used to specify a destination, in data set name format. This format lets you specify a sequence of destinations to be identified, where each qualifier is a specific destination. For example, a data set vaulted first at location OUTD and then sent to OLTS could have an MSPDEST of 'OUTD.OLTS'. The actual values depend on the support provided by your tape management system. **&MSPARM** used to specify any additional information related to a system-driven tape management allocation. This is a variable length field, and its value is specified through an external exit. **&MSPOLICY** used to specify a management policy related to a system-driven tape management allocation. This is an 8 character field, and its value is specified through an external exit. **&MSPOOL** used to specify a tape pool name associated with the data set being allocated. In a system-managed tape environment with scratch pool support, this variable might be used to specify a default storage group, where the tape storage group is equivalent to the tape pool specified in the variable. DFSMSrmm allows you to move your pooling decisions to the SMS ACS routines so that you can assign a scratch pool based on ACS input variables. A scratch pool selected in this way equates to a storage group and includes all assigned volumes in the DFSMSrmm control data set. DFSMSrmm passes scratch pooling decisions using the pre-ACS interface in the MSPOOL variable so that it can be the base for assigning a storage group. For non-system-managed tapes, DFSMSrmm also calls the management class and storage group ACS routines so that a storage group can be assigned for use as a scratch pool name.

You can assign a storage group using the DFSMSrmm RMMPOOL environment call. DFSMSrmm uses the assigned storage group as the pool name to validate that a mounted scratch volume is from the requested pool.

DFSMSrmm also allows you to move your VRS policy assignments to the SMS routines so that you can use any of the existing ACS input variables as the base for assigning a management class. You can assign a VRS policy by name to a system-managed or non-system-managed tape data set, and DFSMSrmm uses this management class name to identify the VRS policy. DFSMSrmm passes VRS management value decisions using the pre-ACS interface in the MSPOLICY variable so that it can be the base for management class assignments. DFSMSrmm also calls ACS routines for assigning a management class to non-system-managed tapes.

When DFSMSrmm calls ACS routines for a management class or storage group name, the &ACSENVIR variable is set to RMMVRS for the required management class and RMMPOOL for the required storage group. When RMM calls the ACS routines with the &ACSENVIR variable set to either RMMPOOL or RMMVRS, &STORCLAS is set to USERRMM.

See z/OS DFSMSrmm Managing and Using Removable Media and z/OS DFSMSrmm Implementation and Customization Guide for further information on the SMS pre-ACS interface and the updates to the ACS input variables.

Parent topic:ACS routine environments

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Access Method Services

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For the ALLOCATE and DEFINE commands (&ACSENVIR='ALLOC'), the data class ACS routine and then the storage class ACS routine are executed. If the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed.

For the IMPORT command, the storage class ACS routine is executed first. If the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed.

The STORCLAS ACS routine is not redriven or invoked on data set rename.

Redriving the MGMTCLAS ACS routine on data set rename invokes the management class ACS routine when an SMS managed cluster, Generation Data Set (GDS) or non-VSAM data set is renamed. Catalog management invokes the MGMTCLAS ACS routines during rename processing. You can reassign a different management class based on the following data set attributes: New data set nameData set typeData set organizationExpiration date in the catalogOld management classData classStorage classRecord organizationUser informationGroup information

ACS routines support RENAME. The following read-only variables are set for the management class ACS routine: &ACSENVIR:RENAME&DSN of the new data set name&DSORG&DSTYPE&EXPDT&STORCLAS&DATACLAS&MGMTCLAS&GROUP of the job&USER of the job&HLQ&LLQ

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMShsm

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The management policy during a data set recall or recover is determined by the value set for &ACSENVIR in the management class ACS routine. The storage class ACS routine is then invoked to apply performance and availability criteria. The data class routine is not invoked. Additionally, during DFSMShsm RECALL processing, the storage group routine gets a different value for the &ACSENVIR field, depending on the data mover that is used. If DFSMSdss is the data mover, RECOVER is passed as the environment; if DFSMShsm is the data mover, RECALL is passed as the environment. Because of this, do not use the storage group ACS routine to test the value in &ACSENVIR for recall or recover, as this could yield inconsistent results. 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.

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMSdss

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For the COPY command (&ACSENVIR='ALLOC') and the RESTORE command (&ACSENVIR='RECOVER'), the storage class ACS routine is executed first. If the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed.

An application program may set its own ACS environment for the logical data set COPY and RESTORE commands. The ACS environment may be set when the application is presented with the EIREC22 data area in Exit 22. The specification of an ACS environment will override the default &ACSENVIR values for the logical data set COPY and RESTORE commands. The read only variables available will be no different than the ones available for the COPY command (&ACSENVIR='ALLOC') and the RESTORE command (&ACSENVIR='RECOVER'). For CONVERTV SMS TEST and CONVERTV SMS, (&ACSENVIR='CONVERT'), the storage class ACS routine is executed first. If the storage class ACS routine determines that the storage class is not null, the management class ACS routine is executed.

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DFSMSrmm

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DFSMSrmm calls the ACS routines to request the assignment of storage group and management class names for non-system-managed tape data sets. [Table 1](#) lists the read-only variables that are set for DFSMSrmm requests.

Table 1. DFSMSrmm Read-Only Variables

When the &ACSENVIR variable is set to RMMPOOL, DFSMSrmm requests you to return a storage group name. DFSMSrmm requests that both the management class and storage group ACS routines are run. With a combination of these routines, you can decide whether or not you want to return a storage group value and what the value will be. If a storage group name is returned, it must be a valid tape storage group name. Using different &ACSENVIR values helps you differentiate between DFSMSrmm requests for a storage group name and allocation requests for system-managed data sets.

When the &ACSENVIR variable is set to RMMVRS, DFSMSrmm requests you to return a management class name. DFSMSrmm requests that only the management class ACS routine is run. Using different &ACSENVIR values helps you differentiate between DFSMSrmm requests for a management class name and allocation requests for system-managed data sets.

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ISMF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you are testing ACS routines, the data class ACS routine and then the storage class ACS routine are executed. If the storage class ACS routine determines that the storage class is not null, the management class ACS routine and then the storage group ACS routine are executed. You can also execute each ACS routine separately from the others when performing tests.

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

OAM

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For the OSREQ CHANGE command (&ACSENVIR='CHANGE'): If storage class, or both storage class and management class, are specified, both the storage class and management class ACS routines are executed, in that order. If only management class is specified, only the management class ACS routine is executed, with the old storage class used as input.

For the OSREQ STORE command (&ACSENVIR='STORE'): If the object is the first object in a collection, the storage class, management class and storage group ACS routines are executed, in that order. These routines determine the defaults for the object collection. Whenever storage class or management class is specified, the storage class and management class ACS routines are executed for the object with the specified classes as input. Then the ACS routines derive the initial storage class and management class for the object rather than having the object use the default initial storage and management classes for the collection.

At class transition time, as defined by the management class associated with an object, (&ACSENVIR='CTRANS'), the storage class routine and then the management class routine are executed.

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Processing of SMS classes and storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For each of the SMS classes, the processing is as follows:

If you have an ACS routine in your CDS to determine the SMS class, SMS executes the routine. Next, SMS executes the corresponding *ACS installation exit*. An ACS installation exit is an assembler language program you can write to perform processing beyond the scope of the standard ACS routines. Such processing might involve: Calling other programsWriting SMF recordsWriting generalized trace facility (GTF) trace recordsPerforming arithmetic calculationsMaintaining large tables of information for quick searchesWriting dumpsInvoking ACS routines only once. Each exit can override the corresponding SMS class, whether explicitly specified or previously determined by an ACS routine. The exit can also invoke the corresponding ACS routine a second time, but this does not cause the installation exit to be reinvoked. See *z/OS DFSMS Installation Exits* for additional information about ACS installation exits.

Finally, if RACF® is installed and the storage administrator has defined STORCLAS and MGMTCLAS as resources and permitted end users to specify them, the system verifies that the class is defined in the currently active configuration. The system then checks to verify that end users are allowed to use a selected management class or storage class. For storage groups, SMS invokes only the storage group ACS routine. Storage group does not have a corresponding ACS installation exit.

Parent topic:[ACS routine environments](#)



©Copyright IBM Corporation 1990, 2014

Displaying ACS object information

z/OS DFSMSdfp Storage Administration
SC23-6860-01

First, display the ACS Application Selection panel. See [Creating ACS routines](#). Then, to display information about the ACS objects stored in a control data set, Select option 5, Display. This displays the ACS Object Display panel, which includes the following information:

CDS NAME Can be the name of the SCDS that you entered on the Automatic Class Selection Application Selection panel or 'ACTIVE'. It identifies the data set that contains the ACS objects that are to be displayed.

ACS RTN TYPE Always lists the four types of ACS objects that can belong to an SCDS, even if they do not actually exist in the SCDS being displayed. If they do not exist, the corresponding display fields contain dashes.

SOURCE DATA SET ACS ROUTINE TRANSLATED FROM Contains the name of the data set that has the source code responsible for creating the ACS object. The figure shows that the data set name is folded if it is more than 23 characters in length. If an ACS object of a particular type does not exist in the SCDS being displayed, then this field contains dashes.

MEMBER NAME Displays the member name within the data set that contains the ACS source code used to create the corresponding ACS object. If an ACS object of the particular type does not exist in the SCDS being displayed, or if the SOURCE DATA SET ACS ROUTINE TRANSLATED FROM is not a PDS or PDSE, the field contains dashes.

LAST TRANS USERID Displays the TSO user ID of the person who last translated the ACS routine. If an ACS object of the particular type does not exist, the field contains blanks.

LAST DATE TRANSLATED Displays the date when the corresponding ACS object was created. If an ACS object of the particular type does not exist, the field contains blanks.

LAST TIME TRANSLATED Displays the time when the corresponding ACS object was created. If an ACS object of the particular type does not exist, the field contains blanks.

Parent topic: [Defining ACS routines](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Deleting an ACS object from an SCDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Display the ACS Application Selection panel. See [Creating ACS routines](#).

Perform the following steps to display information about the ACS objects stored in a control data set.

Select option 6, Delete, on the ACS Application Selection panel. This displays the Delete ACS Object panel, on which: **SCDS Name** Contains the name of the SCDS from which the ACS object is to be deleted. It is a required field and is primed with the last referenced SCDS. The SCDS name cannot be 'ACTIVE'. The default is blank. **ACS Routine Type** Contains the type of ACS object. It is a required field and it is primed with the last used value. The default is blank.

Press Enter to see the Confirm Delete Request panel to confirm that you want to delete the ACS object.

Parent topic: [Defining ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using security labels in ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use an automatic class selection (ACS) routine read-only security label variable, &SECLABL, as input to the ACS routine. The value of the security label (&SECLABL) is defined for specific users or data sets.

The security label is a name used to represent the association between a particular security level and a set of security categories. It indicates the minimum level of security required to access a data set protected by this profile. You can set the security label by entering it in the RACF® profile of the data set.

Before you begin: To understand the concepts of security labels, read *z/OS Planning for Multilevel Security and the Common Criteria*.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

[Planning and installing](#)

[Administering](#)

Parent topic:[Defining ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Planning and installing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This section describes the planning and installation tasks that you must perform to implement security labels in ACS routines.

[Planning to use security labels in ACS routines](#)

[Installing security labels in ACS routines](#)

Parent topic:[Using security labels in ACS routines](#)

 | 

©Copyright IBM Corporation 1990, 2014

Planning to use security labels in ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: An installation can define its own security labels. You can use security labels to associate a specific security level with a set of (zero or more) security categories. Consider using security labels if you do not want to mix data for different classified projects. Security labels allow you to segregate data of specific classifications on specific sets of volumes. You can assign an appropriate storage group dedicated to data of that classification.

Perform the following steps to help you decide whether and how to use security labels in ACS routines:

Decide whether you want to segregate data of specific classifications on specific sets of volumes.

Create a list of security label names that you can use to segregate groups of data.

Decide which storage groups should use security labels.

Decide which storage group ACS routines to update with the &SECLABL read-only variable.

Now you are ready to implement security labels in ACS routines.

Parent topic:[Planning and installing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Installing security labels in ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to implement security labels in ACS routines:
Create the security label and give users access to it. For more information, see [Creating a security label in the RACF data set or user's profile](#).

Specify the DD SECMODEL or DD PROTECT=YES parameter in the JCL or dynamic allocation to extract the security label from the RACF® user's profile. Otherwise, the security label is extracted from the data set profile. For more information, see [Using JCL or dynamic allocation to extract the security label from the RACF user's profile](#).

Update the storage group ACS routine with the &SECLABL read-only variable.

Use the ISMF ACS Test Case Define/Alter application to test the security labels in the storage group ACS routines. For more information, see [Writing ACS test cases using SECLABEL](#).

Validate and activate the SCDS. Select option 8 from the ISMF Primary Option Menu for Storage Administrators, to invoke the Control Data Set (CDS) Application Selection panel on ISMF. Specify the name of the SCDS to validate in the CDS NAME field. Select option 4, Validate the SCDS.

Now you are ready to use security labels in storage group ACS routines to segregate different types of data.

Parent topic:[Planning and installing](#)



Administering

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Administration steps include creating security labels, extracting security label values from a RACF® profile, and updating ACS routines with the &SECLABL variable. First, you set the ***seclabel*** value from either the user's RACF profile or from the RACF data set profile as input to the ACS routines. The ***seclabel*** value is set to null if the RACF SECLABEL class is not active. If you specify the DD SECMODEL or PROTECT=YES parameter in the JCL or dynamic allocation, the SECLABEL is extracted from the RACF data set profile. Your installation's ACS routines can assign storage groups depending on the &SECLABL value along with other read-only ACS variables.

Tip: If overflow storage groups or extended storage groups are also defined, ensure that the security levels do not conflict.

Restriction: For z/OS® V1R6 DFSMS, customers cannot use &SECLABL in ACS routines if they are using automatic data set protection (ADSP).

[Creating a security label in the RACF data set or user's profile](#)

[Using JCL or dynamic allocation to extract the security label from the RACF user's profile](#)

[Specifying the &SECLABL read-only variable in the ACS routine](#)

[Writing ACS test cases using SECLABEL](#)

Parent topic:[Using security labels in ACS routines](#)



Creating a security label in the RACF data set or user's profile

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: To understand security labels, read z/OS Security Server RACF Security Administrator's Guide. For information about RACF® commands, see z/OS Security Server RACF Command Language Reference.

Perform the following steps to create a security label:

Define the SECLEVEL profile to the SECDATA class using the RACF RDEFINE command. Example: RDEFINE SECDATA SECLEVEL UACC(NONE)

Define security levels as members of the SECLEVEL profile in the SECDATA class.

Example: RALTER SECDATA SECLEVEL ADDMEM(seclevel-name/seclevel-number ...)

Define the CATEGORY profile to the SECDATA class using the RDEFINE command. Example: RDEFINE SECDATA CATEGORY UACC(NONE)

Define categories as members of the CATEGORY profile in the SECDATA class.

Example: RALTER SECDATA CATEGORY ADDMEM(category-1 category-2 ...)

For each security label, define a profile in the SECLABEL class. Example: RDEFINE SECLABEL security-label SECLEVEL(seclevel-name) ADDCATEGORY(category-1 category-2 ...)

Provide READ access authority to each user of the security label. In this example, EAGLE is the name of the security label. Example: PERMIT EAGLE CLASS(SECLABEL) ACCESS(READ) ID(AHLEE GROUP1)

When you are ready to start using security labels, activate the SECLABEL class and activate SETROPTS RACLIST processing for the class. Example: SETROPTS CLASSACT(SECLABEL) RACLIST(SECLABEL)

Now you are ready to assign the security label to the &SECLABL read-only variable to use in the storage group ACS routine.

Parent topic:[Administering](#)



Using JCL or dynamic allocation to extract the security label from the RACF user's profile

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Specify the DD SECMODEL or DD PROTECT=YES parameter in the JCL or dynamic allocation to extract the security label from the RACF® user's profile. Otherwise, the security label is extracted from the data set profile. For information about using JCL statements, see *z/OS MVS JCL Reference*. For information about using dynamic allocation, see *z/OS MVS Programming: Authorized Assembler Services Guide*.

Perform the following step to extract the security label from the user's profile:
Specify the DD SECMODEL or DD PROTECT parameter in the JCL or dynamic allocation, then submit the job.

Example using the SECMODEL parameter:

```
//STEP20 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=A
//DD3   DD DSN=USER#1.S16SL001.DATASET3,
//        DISP=(NEW,CATLG),SPACE=(TRK,(2,5)),
//        STORCLAS=S1P01S01,UNIT=3390,
//        SECMODEL=USER#1.S16SL001.MODEL.DATASET
```

The above example specifies the DD SECMODEL parameter in JCL to extract a security label from the discrete profile. The USER#1.S16SL001.MODEL.DATASET data set has been assigned a security label. The security label of the newly allocated data set USER#1.S16SL001.MODEL.DATASET is extracted from the RACF discrete profile of user USER#1.

Example using the PROTECT parameter:

```
//STEP16 EXEC PGM=IEFBR14
//SYSPRINT DD SYSOUT=A
//DD1   DD DSN=USER#1.S16SL002.DATASET1,
//        DISP=(NEW,CATLG),SPACE=(TRK,(2,5)),
//        STORCLAS=S1P01S01,UNIT=3390,PROTECT=YES
```

The above example specifies the DD PROTECT=YES parameter in JCL to extract a security label from the discrete profile. The security label of the newly allocated data set USER#1.S16SL002.DATASET1 is extracted from the RACF discrete profile of user USER#1.

Submit the job. The security label is extracted from the specified user's profile.

Result: Now you are ready to assign the security label to the &SECLABL read-only variable to use in the storage group ACS routine.

Parent topic:[Administering](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Specifying the &SECLABL read-only variable in the ACS routine

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Before you begin: Read-only variables contain data set and system information, and they reflect what is known at the time of the allocation request. You can use read-only variables in comparison operations, but you cannot change their values. To understand how to test ACS routines and use read-only variables, read [Testing ACS routines](#) and [Writing ACS routines](#).

Perform the following steps to use the &SECLABL read-only variable in the ACS routine:

Update the storage group ACS routine with the &SECLABL read-only variable. This example assumes that a RACF® security label ALERT is already defined to the system. Example:

```
PROC &STORGRP
SELECT
WHEN (&SECLABL = 'ALERT')
DO
  SET &STORGRP = 'S1P01'
  WRITE 'ASSIGN DATA SETS WITH SECLABEL ALERT TO STORAGE GROUP: S1P01'
  EXIT CODE(0)
END
OTHERWISE
DO
  SET &STORGRP = 'S1P02'
  WRITE 'ASSIGN DATA SETS WITHOUT SECLABEL ALERT TO STORAGE GROUP: S1P02'
  EXIT CODE(0)
END
END
END
```

Validate the storage group ACS routine. On the ISMF Primary Option menu, select option 7, Automatic Class Selection. On the ACS Application Selection panel, select option 3, Validate to validate the ACS routine. On the Validate ACS Routines or Entire SCDS panel, specify the name of the SCDS where the ACS routine resides. Specify the ACS routine type, which is SG (storage group). Specify the name of the listing data set to where the validation results are printed. Press Enter to validate the storage group ACS routine. On the CDS Application Selection panel, select the Validate option.

Test your storage group ACS routines. For more details, see [Writing ACS test cases using SECLABEL](#).

Validate the SCDS with the updated storage group ACS routine.

Activate the validated SCDS. On the CDS Application Selection panel in ISMF, complete one of the following actions: Select the Activate option, or Enter the ACTIVATE operator command on the command line. Example: ACTIVATE SCDS *dsname*

From the operator console, enter the SETSMS command. Example: SETSMS SCDS *dsname*

Now you are ready to use security labels in storage group ACS routines to segregate different types of data.

Parent topic:[Administering](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Writing ACS test cases using SECLABEL

z/OS DFSMSdfp Storage Administration
SC23-6860-01

After completing your ACS routines, you can write and execute test cases using the ISMF Automatic Class Selection application. After testing the individual routines of a new or modified configuration, you can activate it with greater confidence. To write an ACS test case using the SECLABEL value, follow these steps:

Select ISMF option 7.4, ACS Test application to display the ACS Test Case Define/Alter panel.

Specify the security label value in the SECLABEL field. You also can specify other values that you want to test in the ACS routines.

To run the test cases in a test library, specify the library name and select option 3 in the ACS Test Selection panel. On the Test Case Routines panel, indicate which routines you want to test.

Tip: If you are using NaviQuest, select option 11, selection 1.1 to specify the security label value in the Test Case Generation from Saved ISMF List Entry panel.

Parent topic:[Administering](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Activating Storage Management Subsystem configurations

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can activate an SMS configuration manually, or automatically at IPL. This topic shows you how to perform the initial activation of an SMS configuration using a four-step manual approach. It also explains how you can activate an SMS configuration automatically at IPLs. In addition, it explains how you can change individual SMS parameters with the SETSMS operator command.

Prerequisite: When you activate an SMS configuration, ensure that all of the DASD volumes that belong to the configuration are initialized as SMS volumes. Otherwise, attempted allocations to an improperly initialized volume will fail. However, initialization for tape volumes is no different for SMS-managed and non-SMS-managed volumes.

[Manually activating the first Storage Management Subsystem configuration](#)
[Automatically activating a Storage Management Subsystem configuration](#)
[Displaying storage management subsystem information](#)
[Controlling DFSMStvs processing](#)
[Changing Storage Management Subsystem parameters](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Manually activating the first Storage Management Subsystem configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

IGDSSIIN is the subsystem initialization routine module for SMS. By omitting it from the SMS entry of IEFSSNxx for each system in the SMS complex, you can manually control the activation of an SMS configuration. Refer to [Preparing for the Storage Management Subsystem](#).

Step one: IPL each system in the SMS complex

Step two: prepare one system

Step three: activate the configuration from one system

Step four: activate SMS on the other systems

Parent topic:[Activating Storage Management Subsystem configurations](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Step one: IPL each system in the SMS complex

z/OS DFSMSdfp Storage Administration
SC23-6860-01

After defining SMS as a subsystem to z/OS®, IPL each system in the SMS complex. The presence of the SMS entry in IEFSSNxx tells z/OS to recognize SMS as a valid subsystem within each system. The absence of the IGDSSIIN module name in IEFSSNxx tells the system that you want to manually start SMS.

Parent topic:[Manually activating the first Storage Management Subsystem configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Step two: prepare one system

z/OS DFSMSdfp Storage Administration
SC23-6860-01

From one system in the SMS complex, issue the T SMS=xx command, where xx identifies IGDSMSxx as the SMS initialization control member of SYS1.PARMLIB. SMS uses the ACDS and COMMDS identified in IGDSMSxx to manage storage. Because the initial ACDS and COMMDS are empty, the system is activated with a *null configuration*. Keep in mind that a null configuration is only intended as a migration path.

For additional information, see [z/OS MVS Initialization and Tuning Guide](#).

Requirement: All systems in the SMS complex must be running in the same mode. When an SMS control data set that supports only eight names is accessed for update on a system running in 32-name mode, you must convert the data set to a new, incompatible format in order to support 32 names. Confirm this conversion using the operator console or the ISMF. This conversion is permanent, so make copies of your control data sets before the system mode is converted.

Parent topic:[Manually activating the first Storage Management Subsystem configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)**Step three: activate the configuration from one system**

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The configuration is only activated once for the SMS complex. It is not necessary to activate a configuration from every system in the SMS complex. After activating SMS with a null configuration, activate an SMS configuration contained in a valid SCDS on the same system. You can use either the ISMF ACTIVATE command or the SETSMS operator command. Both procedures copy the contents of the SCDS to the ACDS specified in IGDSMSxx.

When an SMS control data set that supports only eight names is accessed for update on a system running in 32-name mode, you must convert the data set to a new, incompatible format in order to support 32 names. Confirm this conversion, using the operator console or ISMF. This conversion is permanent, so you should make copies of your control data sets before the system mode is converted.

[Activating with the ISMF ACTIVATE command](#)

[Activating with the SETSMS operator command](#)

Parent topic:[Manually activating the first Storage Management Subsystem configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Activating with the ISMF ACTIVATE command

z/OS DFSMSdfp Storage Administration
SC23-6860-01

On the Control Data Application Selection panel, specify the name of an SCDS and issue the ACTIVATE command from the command line. A *Write to Programmer* message indicates if the activation is successful, provided you have WTPMSG in the TSO/E PROFILE.

Parent topic:[Step three: activate the configuration from one system](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Activating with the SETSMS operator command

z/OS DFSMSdfp Storage Administration
SC23-6860-01

From the operator console, issue the command: `SETSMS SCDS(dsname)`
where dsname identifies the name of the SCDS to be activated. Using the SCDS defined earlier in this manual, the command would be: `SETSMS SCDS(SMS.SCDS1.SCDS)`

Parent topic:[Step three: activate the configuration from one system](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Step four: activate SMS on the other systems

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For the other systems in the SMS complex, use the T SMS=xx command to start SMS on those systems, using the SMS configuration identified in Step Three. In each system, IGDSMSxx specifies the name of the ACDS containing the SMS configuration. All of the IGDSMSxx members must point to the same ACDS. Because the ACDS is no longer empty, the systems use it (and the COMMDS) to manage storage.

Parent topic:[Manually activating the first Storage Management Subsystem configuration](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Automatically activating a Storage Management Subsystem configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For each system in the SMS complex, update the SMS entry in IEFSSNxx to include the IGDSSIIN module name.

Make certain that each ID field identifies the IGDSMSxx member containing the name of the last used ACDS. All of the IGDSMSxx members must point to the same ACDS. At future IPLs, the SMS configuration contained in the ACDS is activated by all systems in the SMS complex. For example, the following command indicates that the ACDS specified in IGDSMS02 contains the SMS configuration to be activated at future IPLs:

```
SMS,IGDSSIIN,'ID=02,PROMPT=DISPLAY'
```

Parent topic:[Activating Storage Management Subsystem configurations](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying storage management subsystem information

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the DISPLAY SMS MVS command to display information about the storage management subsystem, such as the following information:
Active SMS configurationSMSVSAM status of sharing control data setsSMSVSAM server nameCoupling facility cache and lock structuresFor a description of the DISPLAY SMS command, see z/OS MVS System Commands.

Parent topic:[Activating Storage Management Subsystem configurations](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Controlling DFSMStvs processing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

While DFSMStvs is processing, you can monitor the performance of applications programs and maintain data integrity during backup-while-open (BWO) processing.

[Monitoring application programs that use DFSMStvs](#)

[Changing DFSMStvs status](#)

Parent topic:[Activating Storage Management Subsystem configurations](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Monitoring application programs that use DFSMStvs

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For information about monitoring application programs that use DFSMStvs and tuning performance, see [z/OS DFSMStvs Planning and Operating Guide](#).

Parent topic:[Controlling DFSMStvs processing](#)



©Copyright IBM Corporation 1990, 2014

Changing DFSMStvs status

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the VARY SMS MVS command to change the status for DFSMStvs in these ways:

Change the state of a DFSMStvs instance or change the state of all DFSMStvs instances in the sysplex
Change the state of a log stream to which DFSMStvs has access
Change the state of a data set for VSAM record-level sharing (RLS) and DFSMStvs access
Start or stop peer recovery processing for a DFSMStvs instance
Restriction: You cannot use the VARY SMS command to change the state of a DFSMStvs instance while it is initializing. Any attempt to do so is suspended until the initialization completes.

The possible states of a DFSMStvs instance follow:

ENABLEEnables DFSMStvs to begin accepting new units of recovery for processing.
DISABLEPrevents DFSMStvs from processing new work requests. DFSMStvs does not process new work requests from units of recovery that are currently in progress.
QUIESCEPrevents DFSMStvs from accepting any new units of recovery for processing. DFSMStvs completes the processing of any units of recovery in progress.

The possible states of a data set follow:
ENABLEUnquiesces a data set for VSAM RLS and DFSMStvs access.
DISABLEQuiesces a data set for VSAM RLS and DFSMStvs access. For a description of the VARY SMS command, see z/OS MVS System Commands. For information about the effects of DFSMStvs, log stream, and data set states on DFSMStvs processing, see z/OS DFSMStvs Planning and Operating Guide.

Parent topic:[Controlling DFSMStvs processing](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Changing Storage Management Subsystem parameters

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the SET SMS or SETSMS MVS command to change SMS parameters. Use the SET SMS command to initialize SMS parameters from PARMLIB member IGDSMSxx and start SMS (or restart SMS if it was already active). Use the SETSMS command when SMS is active (running) to change a subset of SMS parameters by entering them from the console without changing IGDSMSxx.

For descriptions of SET SMS and SETSMS, see [z/OS MVS System Commands](#).
Exception: To reactivate a null configuration, you must perform an IPL.

Restriction: Some of the parameters contained in the IGDSMSxx parmlib member cannot be changed using the SETSMS operator command. The section

[Parameters of the SETSMS operator command](#) lists only those parameters that can be changed using the SETSMS operator command. For additional information, see [z/OS MVS Initialization and Tuning Guide](#)

[Parameters of the SETSMS operator command](#)

[SETSMS command to alter the setting of BreakPointValue](#)

[SETSMS command to alter the setting of USEEAV](#)

[Considerations when changing Storage Management Subsystem configurations](#)

[OAM considerations when changing SCDSs](#)

Parent topic: [Activating Storage Management Subsystem configurations](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Parameters of the SETSMS operator command

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can change SMS parameters by using the SETSMS operator command. For descriptions of the parameters, see *z/OS MVS System Commands* and *z/OS MVS Initialization and Tuning Guide*.

Parent topic:[Changing Storage Management Subsystem parameters](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SETSMS command to alter the setting of BreakPointValue

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the SETSMS command to change the setting of BreakPointValue without having to re-IPL. This modified setting is in effect until the next IPL when it reverts to the value specified in the IGDSMSxx member of PARMLIB. To make the setting change permanent, you must alter the value in SYS1.PARMLIB.

The syntax of the operator command is: `SETSMS BreakPointValue(0-65520)`

Note: You cannot use the SETSMS command to alter the value inside any individual Storage Group. You must use the ALTER command in the ISMF Storage Group Application.

Parent topic:[Changing Storage Management Subsystem parameters](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SETSMS command to alter the setting of USEEAV

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the SETSMS command to change the setting of USEEAV without having to re-IPL. This modified setting is in effect until the next IPL when it reverts to the value specified in the IGDSMSxx member of PARMLIB. To make the setting change permanent, you must alter the value in SYS1.PARMLIB.

The syntax of the operator command is: `SETSMS USEEAV(YES|NO)`

The USEEAV(NO) setting indicates that no new primary space allocations or extended data sets to a new volume are allowed on an EAV.

Parent topic:[Changing Storage Management Subsystem parameters](#)



©Copyright IBM Corporation 1990, 2014

Considerations when changing Storage Management Subsystem configurations

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When activating a new SMS configuration, you have two options for keeping the currently active SMS configuration information: Keep, but never modify, the original SCDS from which the current SMS configuration was activated. If you choose this method, you need to maintain a log of all status changes, such as VARY storage group commands, that you make to the currently active SMS configuration. If in the future you activate a different SMS configuration but then decide you want to fall back to your original, you can reactivate the SCDS. You lose all the status changes you have made since activating the SCDS and must reenter them, but you return to the original SMS configuration.

Save the current active SMS configuration using the SETSMS operator command:

SETSMS SAVEACDS(ACDS.FALLBACK)

This is the recommended alternative for keeping the SMS configuration information. ACDS.FALLBACK must be an existing, already allocated data set. By using this command, you not only save the current storage management policy in ACDS.FALLBACK, but you also save the status changes you have made since the original SCDS was activated. You can then activate the new SMS configuration. If in the future you decide you want to fall back to the original SMS configuration, you can use the SETSMS operator command to reactivate it:

SETSMS ACDS(ACDS.FALLBACK)

This alternative is useful if you have altered the SCDS that you originally activated.

Restriction: Any SMS status that is changed using the SETSMS command is overridden by the MVS status if a new configuration is activated. See *z/OS MVS System Commands* for information on the SETSMS operator command.

Requirement: All systems in the SMS complex must be running in the same mode. When an SMS control data set that supports only eight names is accessed for update on a system running in 32-name mode, you must convert the data set to a new, incompatible format in order to support 32 names. Confirm this conversion using the operator console or the ISMF. This conversion is permanent, so you should make copies of your control data sets before the system mode is converted.

Parent topic:[Changing Storage Management Subsystem parameters](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

OAM considerations when changing SCDSs

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SMS notifies OAM when a new SCDS has been activated. OAM takes action depending on the RESTART parameter that is specified on the OAM address space. If RESTART=YES is specified, or defaulted, the OAM address space automatically restarts, rebuilding its configuration to match the newly activated SCDS. If RESTART=NO is specified, OAM does not automatically restart, but issues a message acknowledging that an activation has taken place. In this case, you must determine if an OAM restart is necessary. If a restart is necessary, use the MODIFY OAM,RESTART command.

When RESTART=YES is specified, the length of the delay between the SCDS activation and the OAM restart depends on the value specified in the INTERVAL keyword in the active IGDSMSxx PARMLIB member. During this reinitialization, all optical libraries and drives defined to the new SCDS are reset to the initial status values specified in the SCDS. After the OAM restart completes, display all optical libraries and drives, and tape libraries, and then set them to the desired online or offline status before the reinitialization occurred.

For more information, see z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

Parent topic:[Changing Storage Management Subsystem parameters](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Maintaining the Storage Management Subsystem

z/OS DFSMSdfp Storage Administration
SC23-6860-01

After activating SMS, you need to monitor and adjust it over time. This topic explains how to maintain SMS. Maintenance activities include listing, altering and ejecting optical and tape volumes. Maintenance activities also include listing, altering, copying, and deleting the following SMS components: Storage groupsManagement classesStorage classesData classesCopy poolsAggregate groupsOptical librariesOptical drivesTape libraries

Tip: You can use the DFSMS NaviQuest tool to help you maintain your SMS configuration. With DFSMS NaviQuest, you can perform many ongoing storage administration activities in batch, thereby freeing the workstation for other work. For example, you can update configuration values, create reports, and use NaviQuest's cross-referencing capabilities to help you verify changes to the configuration.

See [Using NaviQuest](#) for more information on DFSMS NaviQuest.

[Displaying SMS and OAM information](#)

[Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF](#)

[Altering data set associations](#)

[Altering SMS components](#)

[Copying SMS components](#)

[Deleting DASD volumes from the system](#)

[Deleting DASD volumes from storage groups](#)

[Deleting storage groups](#)

[Deleting SMS classes](#)

[Converting volumes to SMS](#)

[Deleting DASD volume residual data](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying SMS and OAM information

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the DISPLAY SMS operator command to determine the status of SMS, storage groups, DASD volumes, tape volumes, tape libraries, OAM, OSMC, optical libraries, optical volumes and optical drives.

You can also enter the LISTSYS and LISTVOL line operators on the ISMF Storage Group List panels to get information about the currently active configuration, storage groups and volumes.

See z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support and z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries for specific information on using the OAM storage management component commands.

[Displaying information about the active configuration](#)

[Displaying SMS TRACE information](#)

[Displaying storage group status using the DISPLAY SMS command](#)

[Displaying OAM status in a parallel sysplex](#)

[Displaying the caching statistics](#)

[Displaying storage group status using ISMF](#)

[Displaying volume status using ISMF](#)

[Displaying volumes using the DISPLAY SMS command](#)

[Displaying the device status](#)

Parent topic:[Maintaining the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

Displaying information about the active configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To display information about the currently active configuration, issue the following command:

D SMS,ACTIVE or D SMS

where D is an abbreviation for DISPLAY. (The default for the DISPLAY command is to display the active configuration when no options are specified.) An example of the generated output appears in [Figure 1](#).

Figure 1. Displaying Information about the Active Configuration

09.23.36

IGD002I 09:23:36 DISPLAY SMS 401

```
SCDS = SMS.SSMS1.SCDS
ACDS = SMS.ACDS1.ACDS
COMMDS = SMS.COMMDS1.COMMDS
DINTERVAL = 150
REVERIFY = NO
ACSDEFAULTS = NO
      SYSTEM    CONFIGURATION LEVEL      INTERVAL SECONDS
      SYSTEM01  2004/12/01 09:23:29        15
      SYSTEM02  ----- -----          N/A
      SYSTEM03  ----- -----          N/A
      SYSTEM04  ----- -----          N/A
      SYSTEM05  ----- -----          N/A
      SYSTEM06  ----- -----          N/A
      SYSTEM07  ----- -----          N/A
      SYSTEM08  ----- -----          N/A
```

The display shows the names of the current control data sets. The naming convention used here specifies a first level qualifier of SMS for all SMS control data sets. The second level qualifier identifies the type of SMS data set. The third level qualifier uniquely identifies the data set.

The DINTERVAL, REVERIFY ACSDEFAULTS, and OVRD_EXPDT fields contain the values given them when SMS was initialized. For an explanation of these values, see [z/OS MVS Initialization and Tuning Guide](#)

The last portion of the output shows configuration levels about the system from which you issued the DISPLAY command. The configuration level indicates the date and time the ACDS was last updated. This last portion also contains the synchronization time interval (the number of seconds that SMS allows before this system checks the COMMDS for news from other systems in the SMS complex) for each system. The synchronization time interval can be changed using the SET SMS command.

Parent topic:[Displaying SMS and OAM information](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying SMS TRACE information

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To display information about the SMS trace options in effect, issue the command: `D SMS,TRACE`

where D is an abbreviation for DISPLAY. An example of the generated output appears in [Figure 1](#).

Figure 1. Displaying Trace Information

14.57.24

IGD002I 14:57:24 DISPLAY SMS 879

```
TRACE      = {ON|OFF}    SIZE =nnnK      TYPE {ERROR|ALL}
  JOBNAME = {jjj|*}     ASID = {asid|*}
  TRACING EVENTS:
MODULE     = {ON|OFF}    SMSSJF = {ON|OFF}   SMSSI  = {ON|OFF}   ACSINT = {ON|OFF}
OPCMD      = {ON|OFF}    CONF C = {ON|OFF}   CDSC    = {ON|OFF}   CONFS   = {ON|OFF}
MSG         = {ON|OFF}    ERR      = {ON|OFF}   CONF R = {ON|OFF}   CONFA   = {ON|OFF}
ACSPRO     = {ON|OFF}    IDAX     = {ON|OFF}   DISP     = {ON|OFF}   CATG    = {ON|OFF}
VOLREF     = {ON|OFF}    SCHEDP  = {ON|OFF}   SCHEDS  = {ON|OFF}   VTOCL   = {ON|OFF}
VTOCD      = {ON|OFF}    VTOCR   = {ON|OFF}   VTOCC   = {ON|OFF}   VTOCA   = {ON|OFF}
RCD        = {ON|OFF}    DCF     = {ON|OFF}   DPN     = {ON|OFF}   TVR     = {ON|OFF}
DSTACK     = {ON|OFF}
```

The display shows the status of the SMS trace option, the size of the SMS trace table, and the type of SMS trace entries. ERR means only error type trace entries are traced. ALL indicates all types of trace entries are traced. This section also includes the JOBNAME which indicates the tracing scope in relation to jobs being run. ASID indicates the tracing scope in relation to address spaces. ASID means tracing is limited to a particular address space and * indicates tracing is performed for all address spaces.

The last section, TRACING EVENTS, indicates which SMS events are selected for tracing.

You can use this tracing display if you have been requested to collect information by the IBM® Support Center for diagnosing problems. See [z/OS DFSMSdfp Diagnosis](#) for additional information.

Parent topic: [Displaying SMS and OAM information](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying storage group status using the DISPLAY SMS command

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the STORGRP parameter to display the status of storage groups including tape and object storage groups. [Figure 1](#) shows the syntax of the DISPLAY SMS,STORGRP command. See z/OS MVS System Commands for the complete syntax of the DISPLAY SMS command.

Figure 1. DISPLAY SMS,STORGRP Command Syntax

```
{DISPLAY} SMS{ ,STORGRP(storgrp)}
[LISTVOL|DETAIL] [,L={a}]
{D}           [,SG      (ALL)]
[             {cc}]
[             {cca}]
[             {name}]
[             {name-a}]
```

STORGRP({storgrp|ALL}) If STORGRP(storgrp) is specified, the system displays the status of one storage group for each z/OS® system connected to that storage group. If STORGRP(ALL) is specified, the system displays the status of all the storage groups defined in the SMS configuration. You can abbreviate STORGRP as SG.**LISTVOL** Displays the status and volume serial numbers of all the volumes in the storage group. The LISTVOL parameter is ignored for tape, object, and object backup storage groups. This parameter is mutually exclusive with the DETAIL parameter.**DETAIL** Displays detail status and is only valid for tape, object, and object backup storage groups. This parameter is mutually exclusive with the LISTVOL parameter and overrides the LISTVOL parameter if the storage group is tape or object related.**L=a|cc|cca|name|name-a** Specifies where the results of the inquiry are to be displayed: the display area (a), the console (cc), or both (cca). The name parameter is routed to the console referred to by "name" and the name-a parameter is routed to the console referred to by "name" and the screen referred to by "a". The name parameter can be an alphanumeric character string. To display information about the status of a storage group, issue the following command:

D SMS,STORGRP(storgrp)

For storgrp you specify the name of a single storage group or you specify ALL to display the status of all storage groups. An example of the generated output for the single storage group SG1 appears in [Figure 2](#).

Figure 2. Displaying Storage Group Status Information

```
15.12.25          IGD0021 15:12:25 DISPLAY SMS 900          15.12.25          d sms,storgrp(slp01)
STORGRP TYPE          SYSTEM= 1 2 3 4 5 6 7 8
SG1   {DUMMY|POOL|TAPE|VIO} + . . . : . .
***** LEGEND *****
. THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
+ THE STORAGE GROUP OR VOLUME IS ENABLED
- THE STORAGE GROUP OR VOLUME IS DISABLED
* THE STORAGE GROUP OR VOLUME IS QUIESCED
D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
SYSTEM 1 = sysname1   SYSTEM 2 = sysname2   SYSTEM 3 = sysname3
SYSTEM 4 = sysname4   SYSTEM 5 = sysname5   SYSTEM 6 = sysname6
SYSTEM 7 = sysname7   SYSTEM 8 = sysname8   SYSTEM 9 = sysname9
SYSTEM 10= sysname10  SYSTEM 11= sysname11  SYSTEM 12= sysname12
SYSTEM 13= sysname13  SYSTEM 14= sysname14  SYSTEM 15= sysname15
SYSTEM 16= sysname16  SYSTEM 17= sysname17  SYSTEM 18= sysname18
SYSTEM 19= sysname19  SYSTEM 20= sysname20  SYSTEM 21= sysname21
SYSTEM 22= sysname22  SYSTEM 23= sysname23  SYSTEM 24= sysname24
SYSTEM 25= sysname25  SYSTEM 26= sysname26  SYSTEM 27= sysname27
SYSTEM 28= sysname28  SYSTEM 29= sysname29  SYSTEM 30= sysname31
SYSTEM 31= sysname31  SYSTEM 32= sysname32
```

If you have more than 16 systems in your SMS complex, a second message follows, and then the legend follows that.

[Displaying the status of volumes in the storage group](#)

[Displaying storage group detail status](#)

Parent topic:[Displaying SMS and OAM information](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying the status of volumes in the storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Specifying the optional LISTVOL parameter provides the status of the volumes associated with the storage group. If you have volume 123456 in storage group SG1, the following command generates output similar to that shown in [Figure 1](#).

D SMS,STORGRP(SG1),LISTVOL

Figure 1. Displaying Storage Group Volume Status Information

```
15.12.25      IGD0021 15:12:25 DISPLAY SMS 900
SMS,storgrp(slp01),listvol
STORGRP TYPE          SYSTEM= 1 2 3 4 5 6 7 8
SG1    {DUMMY|POOL|TAPE|VIO}   + . . .
VOLUME   UNIT   SYSTEM= 1 2 3 4 5 6 7 8 STORGRP NAME
123456      + . . . . . SG1
***** LEGEND *****
. THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
+ THE STORAGE GROUP OR VOLUME IS ENABLED
- THE STORAGE GROUP OR VOLUME IS DISABLED
* THE STORAGE GROUP OR VOLUME IS QUIESCED
D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
SYSTEM 1 = sysname1   SYSTEM 2 = sysname2   SYSTEM 3 = sysname3
SYSTEM 4 = sysname4   SYSTEM 5 = sysname5   SYSTEM 6 = sysname6
SYSTEM 7 = sysname7   SYSTEM 8 = sysname8   SYSTEM 9 = sysname9
SYSTEM 10= sysname10  SYSTEM 11= sysname11  SYSTEM 12= sysname12
SYSTEM 13= sysname13  SYSTEM 14= sysname14  SYSTEM 15= sysname15
SYSTEM 16= sysname16  SYSTEM 17= sysname17  SYSTEM 18= sysname18
SYSTEM 19= sysname19  SYSTEM 20= sysname20  SYSTEM 21= sysname21
SYSTEM 22= sysname22  SYSTEM 23= sysname23  SYSTEM 24= sysname24
SYSTEM 25= sysname25  SYSTEM 26= sysname26  SYSTEM 27= sysname27
SYSTEM 28= sysname28  SYSTEM 29= sysname29  SYSTEM 30= sysname31
SYSTEM 31= sysname31  SYSTEM 32= sysname32
```

Parent topic:[Displaying storage group status using the DISPLAY SMS command](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying storage group detail status

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you specify the keywords DETAIL and STORGRP with a single storage group name, you get detailed information from OAM about the requested object, object backup, or tape storage group. If you specify DETAIL with ALL for the storage group name, you get detailed information from OAM about all object, object backup and tape storage groups.

See z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support and z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries for sample commands and outputs.

Parent topic:[Displaying storage group status using the DISPLAY SMS command](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying OAM status in a parallel sysplex

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Specifying the OAMXCF parameter displays the status of this instance of OAM within a Parallel Sysplex®. You get the status of each member within the specific OAMplex, as well as the number of transactions waiting for a response from other instances of OAM in the Parallel Sysplex.

D SMS,OAMXCF

See z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support for more information on object support.

Parent topic:[Displaying SMS and OAM information](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying the caching statistics

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you have an IBM® 3990 Storage Control with cache with at least one SMS volume attached, you can display cache statistics, hit ratio, and DASD Fast Write bypasses, by issuing the following command: D SMS,CACHE

An example of the generated output appears below. Figure 1. Displaying Cache and DASD Fast Write Information

```
IGD002I 09:28:38 DISPLAY SMS 411
SSID DEVS READ WRITE HIT RATIO FW BYPASSES
0002 32 80 60 68% 62
0004 14 92 100 76% 30
000E 20 84 100 84% 44
0011 08 100 100 92% 16
001C 08 100 100 86% 24
***** LEGEND *****
SSID THE SUBSYSTEM IDENTIFIER OF THE IBM 3990 STORAGE CONTROL
MODEL 3 TO WHICH THE INFORMATION APPLIES
DEVS THE NUMBER OF SMS-MANAGED DEVICES ATTACHED TO THE IBM 3990
STORAGE CONTROL MODEL 3 WITH THIS SSID
READ THE PERCENTAGE OF CYLINDERS FOR WHICH DYNAMIC CACHING IS
PERMITTED
WRITE THE PERCENTAGE OF CYLINDERS FOR WHICH DYNAMIC DASD FAST
WRITE IS PERMITTED
HIT RATIO THE READ HIT RATIO THAT IS ACHIEVED BY THE IBM 3990 STORAGE
CONTROL MODEL 3 WITH THIS SSID
FW BYPASSES THE NUMBER DASD FAST WRITES PER MINUTE THAT ARE EXECUTED
DIRECTLY WITH DASD BECAUSE OF MVS OVER-COMMITMENT
```

Exception: The read and write percentages are displayed as N/A if the subsystem has the extended platform available. Since the enhanced dynamic cache management algorithm is used, the percentages are no longer valid.

Parent topic:[Displaying SMS and OAM information](#)



©Copyright IBM Corporation 1990, 2014

Displaying storage group status using ISMF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can display a storage group's status in each system in the SMS complex by using the LISTSYS line operator in ISMF. The storage group must be a pool type, and it must be part of the active configuration.

Perform the following steps to display a storage group's status.

Display the Storage Group Application Selection panel (select option 6 from the ISMF Primary Option Menu for storage administrators).

Supply values to create a list of storage groups. For example, you could use the following to create a list of storage groups with names that begin with SGNAME:
CDS Name: 'ACTIVE' Storage Group Name: SGNAME*Option: 1, List. This displays the Storage Group List panel.

Type LISTSYS in the Line Operator column next to the desired pool type storage group on the Storage Group List panel. This shows the SMS Storage Group Status Display. Requirement: If a volume has been initialized with LABEL as an SMS device, the storage group must be defined in the active configuration in order for space information to be displayed when the LISTSYS line operator is issued.

Note: The information provided by LISTSYS differs from that provided by LISTVOL, because the information for the two commands comes from different sources.

The Total Space for Storage Group value is the total capacity of all the online volumes belonging to storage group SGNAME01, of all the volumes that have a status of ENABLE.

MB-TOTALTotal number of MB belonging to the storage group**MB-FREE**Total number of free MB belonging to the storage group**% FREE**Percentage of free storage group spaceRequirement: Space information exists for a volume only if the volume has at least one SMS-managed data set allocated on it.

The Group Space Available for Allocation value represents the sum of all volume space that is currently available to each system online to MVS. The fields contain values only if the SMS SG Status value is ENABLE. **MB-TOTAL**Total available space in MB**MB-FREE**Total available space in MB that is free**% FREE**Percentage of total available space that is free

Parent topic:[Displaying SMS and OAM information](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying volume status using ISMF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to display a volume's status.

Display the Storage Group Application Selection panel (select option 6 from the ISMF Primary Option Menu for storage administrators).

Select option 4, Volume. This displays the Storage Group Selection panel.

Supply values on the Storage Group Selection panel: The single volume or range of volumes you wish to display. Option 1, Display, to display the SMS Volume Status Display panel.

[Values on the SMS volume status display panel](#)

Parent topic:[Displaying SMS and OAM information](#)



©Copyright IBM Corporation 1990, 2014

Values on the SMS volume status display panel

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The SMS Volume Status Display panel shows status information.

Physical Volume Status This field shows the current SMS status of, and the amount of free space on, the volume. The volume status can be:
INITIAL The volume has been defined to a storage group, but all the data sets are not yet under SMS control.
CONVERTED The volume is fully converted to SMS.
NONSMS The volume is not under SMS control.
UNKNOWN The status of the volume is not known.
MB-free This is the total number of free megabytes belonging to the volume.
% Free This is the percentage of free volume space. Requirement: If you do not specify 'ACTIVE' in the CDS Name field, then the Physical Volume Status, MB-free, % Free and MVS Vol Status fields display dashes. The space information only exists for a volume if the volume has at least one SMS-managed data set allocated on it.

SMS Vol Status This field shows the relationship between the volume and SMS. The six possible relationships are:
DISALL SMS does not permit data sets on this volume to be accessed.
DISNEW SMS does not permit the allocation of new data sets on this volume.
ENABLE SMS permits access to data sets on this volume.
NOTCONS SMS does not attempt to access this volume.
QUIALL SMS does not schedule any more jobs that access data sets on this volume.
QUINEW SMS does not schedule any jobs that create new data sets on this volume.

MVS Vol Status This column shows relationships between MVS and the volume whose status is being displayed. Possible relationships are:
BOXEDA status encountered when an application cannot disconnect from a malfunctioning volume. BOXED means that MVS is simulating I/O errors in response to the application's I/O request.
NOTREADY The volume cannot send nor receive I/O now.
OFFLINE I/O is not possible because the storage management subsystem cannot find the address where the volume's VOLSER is mounted.
ONLINE The volume is physically connected to MVS; I/O can proceed.
PENDOFF (Pending Offline) MVS varies the volume offline as soon as persons currently accessing data sets on it have disconnected from it.

Parent topic: [Displaying volume status using ISMF](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying volumes using the DISPLAY SMS command

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To display the status of specific DASD, tape or optical disk volumes issue the following command:

D SMS,VOLUME(serial-number)

If the volume serial number is of a DASD volume in a pool storage group, you get the generated output on the Operator Display panel shown in [Figure 1](#). In this example, the volume serial number is 123456. [Figure 1. Displaying Status Information for Individual DASD Volumes](#)

VOLUME	UNIT	SYSTEM=	1	2	3	4	5	6	7	8	STORGRP	NAME	15.38.00	IGD002I 15:38:00 DISPLAY SMS 918
123456	123		+	SG1			
***** LEGEND *****														
.	THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM													
+	THE STORAGE GROUP OR VOLUME IS ENABLED													
-	THE STORAGE GROUP OR VOLUME IS DISABLED													
*	THE STORAGE GROUP OR VOLUME IS QUIESCED													
D	THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY													
Q	THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY													
SYSTEM 1 = sysname1	SYSTEM 2 = sysname2	SYSTEM 3 = sysname3												
SYSTEM 4 = sysname4	SYSTEM 5 = sysname5	SYSTEM 6 = sysname6												
SYSTEM 7 = sysname7	SYSTEM 8 = sysname8	SYSTEM 9 = sysname8												
SYSTEM 10= sysname10	SYSTEM 11= sysname11	SYSTEM 12= sysname12												
SYSTEM 13= sysname13	SYSTEM 14= sysname14	SYSTEM 15= sysname15												
SYSTEM 16= sysname16	SYSTEM 17= sysname17	SYSTEM 18= sysname18												
SYSTEM 19= sysname19	SYSTEM 20= sysname20	SYSTEM 21= sysname21												
SYSTEM 22= sysname22	SYSTEM 23= sysname23	SYSTEM 24= sysname24												
SYSTEM 25= sysname25	SYSTEM 26= sysname26	SYSTEM 27= sysname27												
SYSTEM 28= sysname28	SYSTEM 29= sysname29	SYSTEM 30= sysname30												
SYSTEM 31= sysname31	SYSTEM 32= sysname32													

See [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support](#) and [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries](#) for information about displaying the status of optical and tape volumes.

Parent topic:[Displaying SMS and OAM information](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying the device status

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the DEVSERV command to request and display basic status information on a device, a group of devices, or storage control units. See z/OS MVS System Commands for a detailed description of the syntax and parameters of the DEVSERV command.

[DEVSERV SMS](#)

[DEVSERV QPAVS](#)

[DEVSERV PATHS](#)

[DEVSERV QDASD](#)

Parent topic:[Displaying SMS and OAM information](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DEVSERV SMS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the DEVSERV SMS or DEVSERV S command to display the volume and storage group status of nn devices that SMS manages, starting with the specified device number.

Use the following syntax for the DEVSERV SMS command:

```
{DEVSERV} {SMS},{ddd[,nn][,ONLINE][,L=(a)][{DS}{S}][,ON][{cc}][,CCA][,NAME]}
```

Where:

dddSpecifies the device number, in hexadecimal, for which the system is to display information.

nnSpecifies the decimal number (1 - 32) of devices for which the system is to display the information, in ascending order beginning with the device you specify. If you do not code nn, the system displays information about the one device you specify.

ONLINE/ONDisplays information about only those specified devices that are online. If you do not specify ONLINE or OFFLINE, the system displays information about both online and offline devices.

OFFLINE/OFFDisplays information about only those specified devices that are offline.

L=a|cc|cca|name|name-aSpecifies where the results of the inquiry are to be displayed: the display area (a), the console (cc), or both (cca). The name parameter is routed to the console referred to by "name" and the name-a parameter is routed to the console referred to by "name" and the screen referred to by "a".

[Figure 1](#) shows the output of the DEVSERV S,430 command for the target device 430.

Figure 1. DEVSERV S,430 Output

<pre>UNIT,DTYPE ,M,VOLSER,VOLSTAT 15.44.26 IGD001I 15:44:26 DEVSERV SMS 921 430 ,3380 ,O,XP0101,ENABLED STORGRP,SGSTAT SXP01,QUIESCED</pre>
<pre>*****LEGEND***** A = ALLOCATED F = OFFLINE M = MOUNT PENDING N = NOT ALLOCATED O = ONLINE P = PENDING OFFLINE</pre>

Parent topic: [Displaying the device status](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DEVSERV QPAVS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DEVSERV QPAVS supports the parallel access volume (PAV) capability of the IBM® Enterprise Storage Server® (ESS). You can use the DEVSERV QPAVS command to perform the following tasks: Describe how a logical subsystem configuration is defined to z/OS®Highlight the inconsistencies, if any, between the IODF and the LSS configurationDisplay unbound alias device types with the UCB parameter and, if necessary, "unbox" a boxed alias device with the UNBOX parameterShow information on both a PAV-base address and its PAV-aliases by specifying the VOLUME parameterDisplay information on devices

Use the following syntax for the DEVSERV QPAVS command:

Where: **QPAVS**Is a required positional keyword. May be abbreviated as QPAVS or QP.
deviceSpecifies the device or devices to be displayed. device can be specified in either of the following formats: sccuu, where s is either 0 or 1 to indicate the desired subchannel and ccuu specifies a 4 hex digit device number (3-digit device numbers must be padded with a leading zero), orccuu, which specifies just the device number. In this case, the active subchannel set is used to return information about the device.
numSpecifies the number of devices to be displayed. num can be a decimal number from 1 to 256. 1 is the default.
DCEDisplays the device class extension block (DCE) of the BASE UCB. If DCE is specified, only one device may be displayed (that is num must be 1).
UCBDisplays the unit control block (UCB) information associated with the device. If UCB is specified, only one device may be displayed (that is num must be 1).
UNBOXCauses QPAVS to unbox the unbound alias device if it is in a BOX state. num can be a decimal number from 1 to 256.
VOLUMEDisplays the parallel access volume (PAV) relationship information for the logical volume, including the PAV base device number and all PAV alias device numbers bound to that base. May be abbreviated as VOL.
HPAVDisplays the number of alias pool devices or the alias pool device numbers associated with a base device number (sccuu) that is the target of the QPAV command. If the sccuu number is issued to a HyperPAV alias device, only that device will be in the output.
SSID=ssidSpecifies the subsystem identification number (SSID) of the subsystem whose information DEVSERV is to display. Note: If a specific set of devices is specified after SSID with the device and num parameters, then IO will be issued only to those devices.

Figure 1 shows the display content when you issue the DEVSERV QPAVS command with parameters other than DCE and UCB. If you specify the DCE or UCB parameter (or both), additional DCE or UCB information (or both) is displayed for the device. This additional information is formatted the same way as the output resulting from the QDASD parameter.

Figure 1. DEVSERV QPAVS Display Content

```

--- DS QP Display Content ---
IEEE459I (time) DEVSERV QPAVS
----- Host Subsystem -----
Configuration Configuration
----- UNIT UA -----
NUM. UA TYPE STATUS SSID ADDR TYPE
----- aa BASE INV-ALIAS ssss uu BASE
          ALIAS-bbbb NOT-BASE ssss uu ALIAS-bb
          NON-PAV NOT-ALIAS ssss uu NC
----- bb ALIAS UNBOUND ssss uu ALIAS-aa

```

sddd bb ALIAS-bbbb	ssss uu ALIAS-aa
sddd aa BASE-H	ssss uu BASE
sddd bb ALIAS-H	ssss uu ALIAS-H

UNLISTED DEVICES AND REASON CODES X
sddd(rc) sddd(rc) sddd(rc) ...

X Below are the reason codes for unlisted devices: **(01)**DEVICE NOT CONFIGURED, UCB NOT FOUND**(02)**UCB NOT CONNECTED**(03)** DEVICE UNAVAILABLE, SCP ROUTINE IN CONTROL**(04)**SUBCHANNEL ERROR **(05)** DEVICE BOXED **(06)** UCB NOT A DASD**(07)**DEVICE I/O ERROR**(08)**DEVICE IS NOT A DASD**(09)** DSE-1 CCW BUILD FAILED**(0A)**DEVICE IS AN UNBOUND PAV-ALIAS**(0B)**DEVICE IS A SECONDARY OF A PPRC PAIR**(0C)**SUBCHANNEL SET VALUE SPECIFIED IS NOT VALID**(0D)**UCB NOT FOUND IN SPECIFIED SUBCHANNEL SET**(0E)**DEVICE IS A HYPERPAV ALIAS**(0F)**DEVICE IS NOT A HYPERPAV BASE OR ALIAS

The following are examples of DEVSERV QPAVS output.

[Figure 2](#) shows the PAV status for the device at the starting address D123 and the next 2 addresses.

Figure 2. DS QP,D123,3 Output

Host Configuration				Subsystem Configuration			IEEE459I 08.20.32 DEVSERV QPAVS 591
UNIT	NUM.	UA	TYPE	STATUS	SSID	UNIT	UA
						ADDR.	TYPE
0D123	23	NON-PAV			0101	23	BASE
0D124	24	NON-PAV			0101	24	BASE
0D125	25	NON-PAV			0101	25	BASE

3 DEVICE(S) MET THE SELECTION CRITERIA							

[Figure 3](#) shows the PAV status for the alias device at the address D2FF, its base, and other associated alias devices of the logical volume.

Figure 3. DS QP,D2FF,VOLUME Output

Host Configuration				Subsystem Configuration			IEEE459I 08.20.32 DEVSERV QPAVS 591
UNIT	NUM.	UA	TYPE	STATUS	SSID	UNIT	UA
						ADDR.	TYPE
0D222	22	BASE			0102	22	BASE
0D2FE	FE	ALIAS-D222			0102	FE	ALIAS-22
0D2FF	FF	ALIAS-D222			0102	FF	ALIAS-22

3 DEVICE(S) MET THE SELECTION CRITERIA							

[Figure 4](#) shows the PAV status for the base device at the address D222 and its alias volumes without the subchannel set support installed.

Figure 4. DS QP,D222,VOLUME Output

Host Configuration				Subsystem Configuration			IEEE459I 08.20.32 DEVSERV QPAVS 591
UNIT	NUM.	UA	TYPE	STATUS	SSID	UNIT	UA
						ADDR.	TYPE
0D222	22	BASE			0102	22	BASE
0D2FE	FE	ALIAS-D222			0102	FE	ALIAS-22
0D2FF	FF	ALIAS-D222			0102	FF	ALIAS-22

3 DEVICE(S) MET THE SELECTION CRITERIA							

[Figure 5](#) shows that you can "unbox" a boxed alias device at the address D6FF.

Figure 5. DS QP,D6FF,UNBOX Output

```
THE DEVSERV QPAV UNBOX COMMAND HAS BEEN SUCCESSFULLY UNBOXED 1
IEEE459I 08.20.32 DEVSERV QPAVS 591
THE DEVSERV QPAV UNBOX COMMAND HAS BEEN EXECUTED WITH
RETURN CODE = xx, REASON CODE = yy 2
IEEE459I 10.04.23 DEVSERV QPAVS 699
D6FF IS NOT AN UNBOUND PAV-ALIAS DEVICE. 3
THE DEVSERV QPAV UNBOX COMMAND IS NOT EXECUTED
```

1 You receive this message if the return code from DEVSERV QPAVS UNBOX is zero.

2 You receive this message if the return code from DEVSERV QPAVS UNBOX is

nonzero.

3 You may also receive this message with the explanation: "nnnn-IS NOT IN BOX STATE."

Figure 6 describes the status of an unbound alias and the display of its associated UCB control blocks.

Figure 6. DS QP,D5EF,UCB Output

Host Configuration			Subsystem Configuration				
UNIT	NUM.	UA	TYPE	STATUS	SSID	UNIT	UA
---	--	--	--	--	--	---	--
UCB AT V01ED2AA8						3010200E00ED2A81	0000000000000000
0088FF04D5EF0000	0000000008E4C3C2	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000	0000000000000000
UCB PREFIX AT V01FF4208						14F4FFFFFFFFFF	
000C804000000000	000000000001004E	289C1253C00080C0	14F4FFFFFFFFFF				
0148000000000001							
UCB COMMON EXTENSION AT V01ED2A80							
0000009182A00000	01FF420800000000	000000000000FCD3D8	01ECA90000000000				
**** UNLISTED DEVICE(S) AND REASON CODES:							
OD5EF(0A)							
**** (0A) - DEVICE IS AN UNBOUND PAV-ALIAS							

Figure 7 shows the PAV status of the devices that have the same SSID value.

Figure 7. DS QP,SSID=1401 Output

Host Configuration			Subsystem Configuration				
UNIT	NUM.	UA	TYPE	STATUS	SSID	UNIT	UA
---	--	--	--	--	--	---	--
0D400	00	NON-PAV			1401	00	BASE
...							
0D422	22	BASE			1401	22	BASE
...							
0D4FF	FF	ALIAS-D422			1401	FF	ALIAS-22
***	256	DEVICE(S)	MET THE SELECTION CRITERIA				

Figure 8 shows the status of NOT-BASE when the address D345 is defined as a PAV-base in the HCD, but not in the ESS logical subsystem.

Figure 8. DS QP,D345 Output

Host Configuration			Subsystem Configuration					
UNIT	NUM.	UA	TYPE	STATUS	SSID	UNIT	UA	
---	--	--	--	--	--	---	--	
0D345	45	BASE			NOT-BASE	0103	45	ALIAS-00
***	1	DEVICE(S)	MET THE SELECTION CRITERIA					

Figure 9 shows the status of NOT-ALIAS when the address D621 is defined as a PAV-alias in the HCD, but not in the ESS logical subsystem.

Figure 9. DS QP,D621 Output

Host Configuration			Subsystem Configuration					
UNIT	NUM.	UA	TYPE	STATUS	SSID	UNIT	UA	
---	--	--	--	--	--	---	--	
0D621	21	ALIAS-D600			NOT-ALIAS	0106	21	BASE
***	1	DEVICE(S)	MET THE SELECTION CRITERIA					

Figure 10 shows the status of INV-ALIAS when the alias address D6F4 for the base volume in the HCD does not match its base address in the ESS logical subsystem.

Figure 10. DS QP,D6F4 Output

Host Configuration			Subsystem Configuration					
UNIT	NUM.	UA	TYPE	STATUS	SSID	UNIT	UA	
---	--	--	--	--	--	---	--	
0D6F4	F4	ALIAS-D600			INV-ALIAS	0106	F4	ALIAS-06
***	1	DEVICE(S)	MET THE SELECTION CRITERIA					

Figure 11 shows the status of NOT-NPAV for the device at the address F60. The device is defined as a NON-PAV device in the HCD, but is given an alias in the ESS logical subsystem.

ESS logical subsystem.

Figure 11. DS QP,F60 Output

Host Configuration			Subsystem Configuration		
UNIT	UNIT	UA	UNIT	ADDR.	UA
NUM.	TYPE		SSID		TYPE
00F60	F0	NON-PAV	NOT-NPAV	0101	F0
					ALIAS-03

Figure 12 shows the display content when you issue the DEVSERV QPAVS command with the HPAV parameter, with the target of the command being a HyperPAV alias device number.

Figure 12. DS QP,E27F,HPAV Output

HOST CONFIGURATION			SUBSYSTEM CONFIGURATION		
UNIT	UNIT	UA	UNIT	ADDR.	UA
NUM.	TYPE		SSID		TYPE
0E27E	ALIAS-H				
0E27F	ALIAS-H				
****	2 DEVICE(S) IN HYPERPAV ALIAS POOL				

Figure 13 shows the display content when you issue the DEVSERV QPAVS command with the HPAV parameter, with the target of the command being a HYPERPAV base device number.

Figure 13. DS QP,E200,HPAV Output

HOST CONFIGURATION			SUBSYSTEM CONFIGURATION		
UNIT	UNIT	UA	UNIT	ADDR.	UA
NUM.	TYPE		SSID		TYPE
0E200	00	BASE-H	3205	00	BASE
0E27E	ALIAS-H				
0E27F	ALIAS-H				
****	3 DEVICE(S) IN HYPERPAV ALIAS POOL				

Figure 14 shows the display content when you issue the DEVSERV QPAVS command with the VOLUME parameter, with the target of the command being a HYPERPAV alias device number.

Figure 14. DS QP,E27F,VOLUME Output

HOST CONFIGURATION			SUBSYSTEM CONFIGURATION		
UNIT	UNIT	UA	UNIT	ADDR.	UA
NUM.	TYPE		SSID		TYPE
****	UNLISTED DEVICE(S) AND REASON CODES :				
0E27F(0E)					
**** (0E)	- DEVICE IS A HYPERPAV ALIAS				

Parent topic:Displaying the device status



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DEVSERV PATHS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the DEVSERV PATHS command to display the number of configured cylinders for a device or range of devices under CYL header. If a device is part of a DUAL COPY pair (devices behind a 3990-3 or older), then the ALT dddd is displayed under the CYL header. The first character of the UNIT number represents the subchannel set number.

Figure 1 shows the output of the DEVSErv P,F4A command. Figure 1. DEVSErv P,F4A Output

```
18.09.58 SYSTEM1          ds p,0f4a,4
18.09.58 SYSTEM1          IEE459I 18.09.58 DEVSErv PATHS 600
UNIT DTYPe M CNT VOLSER CHPID=PATH STATUS
RTYPE SSID CFW TC DFW PIN DC-STATE CCA DDC      CYL CU-TYPE
00F4A,33909 ,0,000,EV9LIA,00=+ 04=+
2105   201F Y YY. YY. N SIMPLEX 3A           10017 2105
***** SYMBOL DEFINITIONS *****
O = ONLINE          + = PATH AVAILABLE
```

Path Attributes Definitions: **NS (Not Specified)** Accessibility not specified **NP (Non Preferred)** The control unit image that was selected by this command is accessible over the interface associated with this Interface ID, but the interface is a non-preferred path for this control unit image. **PF (Preferred)** The control unit image that was selected by this command is accessible over the interface associated with this Interface ID, and the interface is a preferred path for this control unit image.

Parent topic:[Displaying the device status](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DEVSERV QDASD

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the DEVSERV QDASD command to display the number of configured cylinders for a device or range of devices, including device model 3390 model A. The first character of the UNIT number represents the subchannel set number.

[Figure 1](#) shows the output of the DS QD,F41,RDC,DCE command. [Figure 1. DS QD,F41,RDC,DCE Output](#)

```
09.36.34 SYSTEM1      IEE459I 09.36.34 DEVSERV QDASD 625
UNIT VOLSER SCUTYPE DEVTYPE CYL SSID SCU-SERIAL DEV-SERIAL EFC
00F41 IN7996 2107921 2107900 87927 2606 0113-03261 0113-03261 *OK
  READ DEVICE CHARACTERISTIC
2107E833900E5E8C FFFF72032FFFE000F E000E5A205940222 1309067400000000
0000000000000000 3E3E1F02DFEE0001 0677080F007F4A00 003C000000015777
  DCE AT V00E8A578
4878807100000000 000000002051820 D800FFFEFFFFE3E3E 1FF70A00002A0000
00FD7B9C9400F0FE 161F3C1E000A8C00 1500000000015777 0001577600000000
0000000000000000
****      1 DEVICE(S) MET THE SELECTION CRITERIA
****      0 DEVICE(S) FAILED EXTENDED FUNCTION CHECKING
```

When you add the ATTR parameter, DEVSERV QDASD displays device attributes, including if the device is a solid state drive, for example:

```
UNIT VOLSER SCUTYPE DEVTYPE CYL SSID SCU-SERIAL DEV-SERIAL EFC
ATTRIBUTE/FEATURE YES/NO ATTRIBUTE/FEATURE YES/NO
0D300 TK9085 2107921 2107900 65520 2401 0175-02411 0175-02411 *OK
  SOLID STATE DRIVE Y      ENCRYPTION N
```

Parent topic:[Displaying the device status](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can list SMS classes, aggregate groups, storage groups, optical libraries, optical drives, saved lists, DASD volumes, optical volumes, tape libraries, tape volumes, and data sets on application panels. You can also generate listings by selecting the LIST option on application panels or by typing the LIST line operator. [Quick reference to ISMF commands and line operators](#) summarizes all the ISMF commands and line operators, describes them, and lists the applications from which you can issue them.

ISMF supports the following screen sizes for various lists: 24 x 8027 x 13232 x 8043 x 8031 x 160. The examples shown in this document are for displays with a screen size of 24 x 80. Note: 31 x 160 is half of a 3290 screen. Only half of the 3290 screen is used.

[Listing with view and sort](#)

[Listing data classes](#)

[Listing storage groups, management classes, storage classes, aggregate groups, libraries and drives](#)

Parent topic:[Maintaining the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Listing with view and sort

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you specify the list option on the Data Class, Management Class, Storage Class, Storage Group, Aggregate Group, Copy Pool, Library, or Drive Application Selection panels, the values you specify in the Respecify View Criteria field and the Respecify Sort Criteria field determine characteristics of the list that ISMF displays. If you specify yes, Y, in the Respecify View Criteria field of the Application Selection panel, you see the View Entry panel for that application. You can specify which data columns appear in the list and the order in which they appear, and you can save your specifications to use again. If you specify no, N, in the Respecify View Criteria field, ISMF uses the last used values for data column selection and order. The initial value is all data columns in their default order.

If you specify yes, Y, in the Respecify Sort Criteria field of the Application Selection panel, you see the Sort Entry panel for that application. You can specify by which data columns entries in the list are sorted. You can only specify data columns that are already specified as view criteria. If you specify no, N, in the Respecify Sort Criteria field, ISMF uses the default sorting order.

[Listing data classes](#) gives an example of how the View and Sort panels work together. See z/OS DFSMS Using the Interactive Storage Management Facility for detailed information about View and Sort.

Parent topic:[Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Listing data classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use View and Sort to customize the way you see a list. For example, if you want to see a list of data classes with their Data Set Name Type, Logical Record Length, and Record Organization data columns displayed, and you want the entries to appear sorted by data set name type in ascending order, you can perform the following steps.

Select option 4, Data Class, from the ISMF Primary Option Menu for Storage Administrators. This displays the Display the Data Class Application Selection panel.

Supply values on the Data Class Application Selection panel: **Option** Select option 1, List.**Respecify View Criteria** Select the viewing order by typing / for this field.**Respecify Sort Criteria** Select the sorting order by typing / for this field. This displays the Data Class View Entry panel.

Supply values on the Data Class View Entry panel. This panel shows the columns that you can select. Use DOWN to see the remaining pages for this panel. At any time, you can choose option 2, Save, to save your specifications for later use.

To select attributes, type the corresponding "tags," separated by blanks, in the Specify Tags in Sequence Desired field. The tags are the numbers enclosed in parentheses for each column, for example, (26) Data Set Name Type. For example, to select Data Set Name Type, Logical Record Length, and Record Organization, you would type:

Specify tags in Sequence Desired:
====> 26 5 3

Your selections are displayed in the order that you typed them.

Line Operator and Data Class Name always appear as the first and second data columns in a list but, because you cannot select them from this panel, they do not have corresponding tags. You cannot specify them on the this panel as selected tags.

Press Enter to see the Data Class Sort Entry panel.

Specify values for sorting on the Data Class Sort Entry panel. Specify a field by its numerical tag. Tags for Line Operator and Data Class Name always appear on this panel. Notice that the only other tags displayed are the ones you specified in the Data Class View Entry panel. Press Enter to see a list of data classes that uses your view and sort criteria.

On the list panel, you can enter line operators such as COPY, DISPLAY, and HIDE against individual entries. Some line operators, such as ALTER and DELETE, cannot be used with the ACTIVE control data set.

You can scroll up, down, left, and right. However, ISMF always displays the LINE OPERATOR and DATACLAS NAME fields on the left side of the panel. You can also change the View and Sort criteria by issuing the View or Sort command on the command line.

Parent topic:[Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Listing storage groups, management classes, storage classes, aggregate groups, libraries and drives

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The viewing, sorting and saving procedures for lists of storage groups, storage classes, and libraries correspond to the example of the procedures for data classes.

[Processing management classes](#)

Parent topic:[Listing SMS classes, aggregate groups, storage groups, and libraries using ISMF](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Processing management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

With the introduction of ABARS II the number of columns on the Management Class list was modified to include columns 31 through 38. Even though the number of attributes has increased, the Display, Define and Alter functions are simplified with the reorganization of the current and new management class attributes into four separate attribute groups: Space, Backup, Aggregate Backup, and Class Transition. When displaying, defining, or altering a management class, you can specify which set of attributes to process first.

Parent topic:[Listing storage groups, management classes, storage classes, aggregate groups, libraries and drives](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Altering data set associations

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can change the management class or storage class associated with a data set. The management class or storage class you choose for the data set must be in the ACDS. Enter the ALTER line operator against a data set entry on a Data Set List (option 1 in the ISMF Primary Option Menu).

Parent topic:[Maintaining the Storage Management Subsystem](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Altering SMS components

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can change the definitions of SMS components, libraries, volumes and drives by selecting the alter option from the Application Selection panels. You can also change the definitions by entering the ALTER line operator against a list of components. This method is more efficient if you want to make multiple changes in one SCDS.

You can also use the MVS SETSMS command to alter some components. This topic outlines how to alter SMS components. For information on how to alter libraries and drives, see the following: Altering optical libraries: z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support Altering tape libraries and drives: z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries

This topic outlines how to alter each SMS component using the component's Application Selection panel. Note: You must be in storage administrator mode to alter SMS classes. If you are in user mode, you are not authorized to alter SMS classes and libraries. [Storage administrator authorization](#) describes storage administrator mode.

[Altering storage groups](#)

[Redefining volumes among storage groups](#)

[Altering management classes](#)

[Altering storage classes](#)

[Altering data classes](#)

[Altering copy pools](#)

[Altering aggregate groups](#)

[Altering the SCDS on different DFSMS releases](#)

Parent topic:[Maintaining the Storage Management Subsystem](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Altering storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

As the needs of your installation change, you might need to modify storage groups. To modify a storage group, go to the Storage Group Application Selection panel, and specify the name of the SCDS containing the storage group that you want to modify. Specify the storage group name and select option 3, ALTER. ISMF displays the Storage Group Alter panel.

You can also use the SETSMS command to change the SMS status of storage groups.

Restriction: Any SMS status that is changed using the SETSMS command is overridden by the MVS status if a new configuration is activated. See z/OS MVS System Commands for more information on the SETSMS command.

Parent topic:[Altering SMS components](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Redefining volumes among storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps if you defined DASD volumes to belong to one storage group but you now want them to belong to a different storage group.
Display the Storage Group Application Selection panel (select option 6 from the ISMF Primary Option Menu for storage administrators).

Supply values on the Storage Group Application Selection panel: **CDS Name**s the name of the SCDS. **Storage Group Name**s the name of the storage group. **Option**Select option 4, Volume. This displays the Storage Group Volume Selection panel.

Supply values on the Storage Group Volume Selection panel: **Specify a Single Volume (in Prefix), or Range of Volumes**Type the serial numbers of the volumes you want to redefine. **Option**Select option 4, Delete. Press Enter to delete the volumes from their current storage group.

Return to the Storage Group Application Selection panel.

Supply values on the Storage Group Application Selection panel: **Storage Group Name**s the name of the new storage group to contain the volumes. **Option**Select option 4, Volume. This returns you to the Storage Group Volume Selection panel.

Supply values on the Storage Group Volume Selection panel: Specify the volume serial numbers. Select option 2 to define them to the storage group. Press Enter to process the values.

Activate the updated configuration.Result: The changes take effect when you activate this updated configuration.

You are responsible for maintaining consistent physical connections. You must also be aware of multivolume data sets. In general, all of a multivolume data set must be in a single storage group (it is possible, when the primary storage group is stressed, for a multivolume data set to extend to more than one storage group). You must also maintain storage group, management class, and storage class consistency. If a data set has a management class with certain migration and backup attributes, and you redefine the volume containing that data set from one storage group to another, the new storage group might not be eligible for the same migration and backup processing.

You can also use the MVS SETSMS command to change the SMS status of volumes. Any SMS status that is changed by a SETSMS command is effective only until a new configuration is activated. The MVS status is not affected by either the SETSMS command or the activation of a new configuration. Only the SMS status is affected.

Parent topic:[Altering SMS components](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Altering management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

As the needs of your installation change, you might need to modify management classes.

Perform the following steps to modify management classes.

Display the Management Class Application Selection panel (select option 3 from the ISMF Primary Option Menu for storage administrators).

Supply values on the Management Class Application Selection panel: **CDS Names** the name of the SCDS that contains the management class you want to change.

Management Class Names the name of the management class. **Option** Select option 4, Alter. This displays the Management Class Alter panel. This panel contains the same attributes as the Management Class Define panel. The fields are primed with the information that you entered when you defined the management class.

Type over the old information with your new information on the Management Class Alter panel. Use the DOWN command to see the next pages. See [Values on the management class alter panel](#) for more information.

Use the END command to save the changes and return to the Management Class Application Selection panel. Result: The changes take effect when you activate this updated configuration.

[Values on the management class alter panel](#)

Parent topic: [Altering SMS components](#)



©Copyright IBM Corporation 1990, 2014

Values on the management class alter panel

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following list describes the fields that are specified on the Management Class Alter panel:

- # Versions**Specifies the number of versions of data sets that are assigned this management class, that are to be maintained.**Retain Only/Extra**
- Version(s)**Specify the time period for which the copy is to be retained. The unit of measure for the time period is specified in the accompanying Unit field. Restriction: This field cannot be left blank if the Unit field is specified.
- Copy Serialization**Specifies whether aggregate backup should continue even if the data set is allocated to a job.
- A backup Copy Technique**Specifies whether concurrent copy should be used during aggregate backup processing. Valid values are:
 - P**indicates that concurrent copy is preferred and should be used for backup. A data set is backed up on a nonconcurrent copy volume if it does not reside on a volume supported by concurrent copy, or if the volume on which it resides is unavailable for concurrent copy.
 - R**indicates that concurrent copy must be used for backup. Backup fails for data sets that don't reside on volumes that are supported by concurrent copy, or that are unavailable for concurrent copy.
 - S**indicates standard allocation, in which data sets are backed up without using concurrent copy.

Parent topic:[Altering management classes](#)



©Copyright IBM Corporation 1990, 2014

Altering storage classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

As the needs of your complex change and as new hardware becomes available, you might need to change the storage classes.

Perform the following steps to modify a storage class.

Display the Storage Class Application Selection panel (select option 5 from the ISMF Primary Option Menu for storage administrators).

Supply values on the Storage Class Application Selection panel: **CDS Name**Is the name of the SCDS that contains the storage class you want to change.**Storage Class Name**Is the name of the storage class.**Option**Select option 4, Alter. This displays the Storage Class Alter panel. This panel contains the same attributes as the Storage Class Define panel. The fields are primed with the information that you entered when you defined the storage class.

Type over the old information with your new information on the Storage Class Alter panel. For more information, see [Defining storage class attributes](#). Use the DOWN command to see the next page.

Use the END command to save the changes and return to the Storage Class Application Selection panel.Result: The changes take effect when you activate this updated configuration. The changes that you make take effect on existing data sets or objects at the next scheduled or requested processing time for those data sets or objects (for example OSMC cycle for objects).

Parent topic:[Altering SMS components](#)



Altering data classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Perform the following steps to modify a data class.

Display the Data Class Application Selection panel (select option 4 from the ISMF Primary Option Menu for storage administrators).

Supply values on the Data Class Application Selection panel: **CDS Name** is the name of the SCDS that contains the data class you want to change. **Data Class Name** is the name of the data class. **Option** Select option 4, Alter. This displays the Data Class Alter panel. This panel contains the same attributes as the Data Class Define panel. The fields are primed with the information that you entered when you defined the data class.

Type over the old information with your new information on the Data Class Alter panel. For more information, see [Defining data class attributes](#). Use the DOWN command to see the next page.

Use the END command to save the changes and return to the Data Class Application Selection panel. Result: The changes take effect when you activate this updated configuration. The changes you make apply only to new data sets. The following features of the existing data sets are affected: additional volume amount, extent constraint relief, and dynamic volume count.

In addition, page 1 of this panel allows storage administrators to specify a value for the Additional Volume Amount attribute. Additional Volume Amount is applicable only to extended format VSAM multivolume data sets. Additional Volume Amount is specified when Data Set Name Type = EXT and Volume Count > 1. The valid values for this attribute are 'P' (primary), 'S' (secondary), and blank (not specified; the system default is primary).

For descriptions of the attributes contained in the Data Class Alter and the Data Class Define panels, see [Defining VSAM attributes and specifying media types for data class](#).

Parent topic: [Altering SMS components](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Altering copy pools

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the Copy Pool Alter panel to add or remove storage groups from the copy pool. You can also use this panel to alter the number of backup versions. Perform the following steps to modify a copy pool.

Display the Copy Pool Application Selection panel (select option P from the ISMF Primary Option Menu for storage administrators).

Supply values on the Copy Pool Application Selection panel: **CDS Name** is the name of the SCDS that contains the copy pool. **Copy Pool Name** is the name of the copy pool. **Option** Select option 4, Alter. This displays the Copy Pool Alter panel. The Copy Pool Alter panel contains the same attributes as the Copy Pool Define panel. The fields are primed with the information you entered when you defined the copy pool.

Type over the old information with your new information on the Copy Pool Alter panel. For more information, see [Defining a copy pool](#). Use the DOWN command to see the next page.

Use the END command to save the changes and return to the Data Class Application Selection panel.

Parent topic: [Altering SMS components](#)



©Copyright IBM Corporation 1990, 2014

Altering aggregate groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

As the needs of your installation change, you might need to change aggregate groups. To modify an Aggregate Group, go to the Aggregate Group Application Selection panel and specify the name of the SCDS containing the aggregate group you want to alter. Specify the aggregate group name and select option 4, Alter. Perform the following steps to modify an aggregate group.

Display the Aggregate Group Application Selection panel (select option 9 from the ISMF Primary Option Menu for storage administrators).

Supply values on the Aggregate Group Application Selection panel: **CDS Name**s the name of the SCDS containing the aggregate group you want to alter. **Aggregate Group Name**s the name of the aggregate group. **Option**Select option 4, Alter. Press Enter to see the Aggregate Group Alter panel. It contain the same attributes as the Aggregate Group Define panels discussed in [Defining aggregate group attributes](#). The fields are primed with the information you entered when defining the aggregate group.

Type over the old information with your new information on the Aggregate Group Alter panel. For more information, see [Defining aggregate group attributes](#). Use the DOWN command to see the next page.

Use the END command to save the changes and return to the Data Class Application Selection panel. The changes take effect when you activate this updated configuration.

[Editing Aggregate group attributes](#)

Parent topic:[Altering SMS components](#)



Editing Aggregate group attributes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The first page of the Aggregate Group Alter panel contains aggregate group alter attributes. The SCDS Name and Aggregate Group Name fields are output fields that contain the SCDS and aggregate group names you specified on the Aggregate Group Application Selection panel. The Description field is an optional field of 120 characters in which you can describe the aggregate group.

You can specify the following required attributes on the first page of the Aggregate Group Alter panel: **Number of Copies**Specifies the number of aggregate backup output files to be altered. The valid values range from 1 to 15.

Management Class NameSpecifies the management class name from which the aggregate backup attributes are obtained. The valid values are 1 to 8 alphanumeric characters (first character can not be a digit) or a blank.

The second page of the Aggregate Group Alter panel contains the selection data set names for the aggregate group. SCDS Name and Aggregate Group Name are output fields that contain the SCDS and aggregate group names you specified on the Aggregate Group Application Selection panel.

You can specify the following required attributes on the second page of the Aggregate Group Alter panel: **To Edit a Data Set, Specify Number**Select the number of a selection or instruction data set that you want to edit. When you select a data set number, you can alter the data set by invoking PDF Edit. The PDF edit screen is shown in [Figure 1](#). See z/OS ISPF User's Guide Vol I for more information on PDF Edit commands.

Selection Data SetsThe name of the data set containing the lists of data sets to be included in the application backup. You can specify up to five selection data set names. One data set name is required. If you want to enter a fully qualified data set name, enclose the name in single quotation marks. If you do not enclose the name in single quotation marks, the TSO/E prefix is added to the name as the first high level qualifier.

Member NameThe name of the data set member containing the lists of data sets to be included in the application backup if the selection data set is partitioned. This name must be a valid TSO/E data set member name. If the data set specified in the Selection Data Set Name field is a partitioned data set, you must specify a valid member name. There is no default.

Instruction Data Set NameThe name of the data set containing instruction, commands, and such, that are copied into the control file volume after the backup control file. This data set can only be a sequential data set. You must use a valid TSO/E data set name. The data set name, including the TSO/E prefix, can be no more than 44 characters long. This is an optional field and has no default.

Parent topic:[Altering aggregate groups](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Altering the SCDS on different DFSMS releases

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The SCDS can be updated on other DFSMS releases. Before updating the SCDS, you must first apply coexistence PTFs on your system to ensure the integrity of the SCDS. Because new functional enhancements are not normally retrofitted, you need to update the SCDS to the most current level of the system. Back up the SCDS and ACDS as you would for any other production data set for the purpose of recovery. See *z/OS Migration* for more information about the PTFs required and steps to take before you can update the SCDS.

Parent topic:[Altering SMS components](#)

|

©Copyright IBM Corporation 1990, 2014

Copying SMS components

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To simplify the creation of new SMS classes, storage groups, aggregate groups, and SCDSs, you can copy existing ones and modify them.

When you copy pool storage groups, you can optionally copy the volume list associated with the storage group. For other types, the volume list is not copied.

Before you begin: Generate a list from an SCDS by selecting the LIST option on an application panel.

Perform the following steps to copy an SMS component.

In the LINE OPERATOR column of any List panel, type the COPY line operator and press Enter. You see the Copy Entry panel.

Supply values for the fields on the Copy Entry panel: **from Data Set Name**

Identifies the source you are copying. It is primed with the value you specified on the Application Selection panel. **from Construct Name** Identifies the name of the SMS class or storage group you are copying. It is primed with the value from the Name field of the List panel. **from Construct Type** Identifies whether you are copying a data class, storage class, management class, aggregate group or storage group. It is primed with the type of List panel from which you issued the COPY command. You should generate your list from an SCDS instead of the ACDS prior to copying SMS classes, storage groups, and aggregate groups. The copy will have the same type as the original.

to Data Set Name Identifies the target of the copy. It must be the name of an SCDS. It is primed with the value of the from Data Set Name if the from Data Set Name contains an SCDS name. **to Construct Name** Identifies the name of the SMS class or storage group copy. It is primed with blanks. **Replace like-named**

Construct If the SMS class, storage group, or aggregate group that you identify in the to fields already exists, you use this field to specify whether you want to replace it. If you select this option, the replacement occurs. By default the option is not selected, and the replacement does not occur. **Perform Alter** Indicates if you want to change some of the attributes of the copy you are creating. If you select this option, the system displays the Management Class Application Selection panel.

This panel lists four groups of management class attributes. You can then choose an attribute group for altering ahead of other attributes. By default, the option is not selected. You remain on the Copy Entry panel, where you can perform another copy or return to the original List panel. **Copy Storage Group Volumes (Pool SG Only)** Indicates if you want the volumes that are defined in the storage group to be copied. This option is valid only for pool storage groups. If you select this option, the volumes for the pool storage group are copied. They are defined to the new storage group as new volumes; no residual data is copied. By default, the option is not selected.

When you have specified the values, press Enter to perform the copy.

Parent topic: [Maintaining the Storage Management Subsystem](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Deleting DASD volumes from the system

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You might want to delete DASD volumes from the system. You can do this without performing an initial program load (IPL) by performing the following steps:
Set the DASD volume status to DISNEW. This disables any new allocations to the volume.

Move data sets that you want to keep to other volumes using DFSMSdss.

Delete the DASD volume from the storage group that contains it. See [Deleting DASD volumes from storage groups](#) for information on deleting DASD volumes.

Vary the DASD volume offline. See z/OS DFSShsm Storage Administration if you need specific information about how to vary a volume offline.

Issue the DFSShsm DELVOL command from all DFSShsm systems that are aware of the volume. This prevents DFSShsm from attempting to allocate to the volume during subsequent interval migrations. See z/OS DFSShsm Storage Administration if you need specific information on the DFSShsm DELVOL command.

Parent topic:[Maintaining the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

Deleting DASD volumes from storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you delete a DASD volume from a storage group, first set the volume status to DISNEW. This prevents any new allocations to the volume while allowing existing data sets to still be accessed with DISP=SHR, DISP=MOD, or DISP=OLD. Move any data that you do not want converted to non-SMS from the volume. Next, if the DASD volume is to be reused as a non-SMS-managed volume, run a DFSMSdss NONSMS CONVERTV to convert it out of SMS. Finally, remove the volume from the storage group.

To remove DASD volumes from a storage group, select the Storage Group Application Selection panel. Specify the CDS name and the storage group name that contains the volumes you want to delete. Press Enter to get the Storage Group Volume Selection panel. From the Storage Group Volume Selection panel, indicate the volumes you want to delete in the SPECIFY A SINGLE VOLUME (in PREFIX), OR RANGE OF VOLUMES field, select option 4, and press Enter. Be careful when moving or removing a volume from a storage group because the volume could contain part of a multivolume data set. The changes take effect when you activate this updated configuration.

To prevent DFMSHsm from attempting to allocate to volumes which have been deleted from the storage group, issue the DFMSHsm DELVOL command from all DFMSHsm systems which are aware of the deleted volume.

After deletion, the DASD volume is only eligible for non-SMS allocations. However, if you are reassigning the volume to another storage group, you need to define the volume to the storage group and then activate the updated configuration.

Parent topic:[Maintaining the Storage Management Subsystem](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Deleting storage groups

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You might want to delete a pool type storage group so that you can define all of its volumes to other storage groups or so that you can use the volumes for non-SMS allocations. The following sections describe the procedure for deleting storage groups and some considerations for preventing job failures.

[Deleting or moving system-managed data sets from the storage group](#)

[Preventing job failures](#)

Parent topic:[Maintaining the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Deleting or moving system-managed data sets from the storage group

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you plan to use the volumes for non-SMS allocations, you need to delete or move all system-managed data sets. The following procedure shows how to do this and how to remove the storage group from the SMS configuration:

Set the storage group status to DISNEW to prevent any further new allocations to volumes defined to the storage group. As time passes, DFSMShsm space management processing clears most of the data sets off the volumes in the storage group you are deleting, provided AUTO MIGRATE is 'Y' and most of the data sets have an associated management class.

Use DFSMSdss to move any remaining data sets that have lingered on the storage group volumes when the volumes are sufficiently empty.

Build an SCDS to contain the new configuration. Remove the storage group definition from the SCDS and change the storage group ACS routine so that it does not select the deleted storage group.

Activate the new configuration. To declare volumes non-SMS-managed, you need to INIT them through ICKDSF to non-SMS or CONVERT them through DFSMSdss to non-SMS.

When you delete a VIO or dummy type storage group, you do not have to worry about redefining volumes to other storage groups, but you might need to modify the storage group ACS routine.

Parent topic:[Deleting storage groups](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Preventing job failures

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You should be aware that if you delete a storage group while a job that uses this group is running, the job might fail with message IGD17201I. To prevent this, consider performing the following:

Modify your storage group ACS routine so that it does not assign the storage group to be deleted to any new allocations. Validating your configuration causes message IGD06023I to be received. This message is only a warning. As long as there are no error messages, your configuration is valid.

Activate the configuration with the modified storage group ACS routine and allow it to remain active until jobs that used the old ACS routine have completed. This allows executing jobs to complete using the existing storage group definition.

Delete the storage group from the configuration. Because the storage group no longer exists in the configuration, you do not receive MSGIGD06023I.

Activate the configuration from which the storage group has been deleted.

Parent topic:[Deleting storage groups](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Deleting SMS classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You need to exercise caution when you delete storage, management and data classes, because a user or system might attempt to use a data set that references them.

For example, assume a data set has some specified migration attributes, and at some point it meets the migration criteria. After the data set has been migrated, you delete its associated management class. If the data set is recalled in the future, it must go through the management class ACS routine. If the ACS routine fails to override the undefined or deleted management class, the recall fails because a management class that does not exist is associated with the data set. If this occurs, you must either rewrite the management class ACS routine or specify that the data set should be recalled as a non-SMS data set. If you rewrite the ACS routine, you must check for the deleted management class and assign a valid management class to the data set. If you decide to bypass ACS processing, be aware that the data set is recalled to non-SMS storage.

Another consequence of deleting management classes involves DFSMShsm automatic processing. When DFSMShsm runs through its automatic cycle and processes data sets on the basis of their management class attributes, it attempts to retrieve nonexistent management class definitions. Consequently, DFSMShsm skips the processing of these data sets.

Instead of deleting storage classes and management classes, you should prevent any new allocation from using them by rewriting the corresponding ACS routine to override the deleted storage class or management class. It is safer and provides the possibility that all data sets that reference the management class or storage class might eventually be overridden. Also, you can identify all data sets with storage class or management class and then use ALTER to change to a new storage class or management class. If you decide to delete a storage class or management class, make certain that you inform all users well in advance.

Because data classes are used only at data set allocation, they do not have these problems. The original data class name is never referenced or reused.

If you decide to delete an SMS class, enter the following information on the pertinent Class Application Selection panel: SCDS Name:

'SMS.SCDS1.SCDS'xxxx CLASS NAME: * Select option 1. After you press Enter, ISMF displays the Class List panel. Next to the SMS class that you want to delete, enter DELETE in the LINE OPERATOR column. When you press Enter, you see the Confirm Delete Request panel. Confirm that the displayed SMS class is the one that you want to delete. If it is, enter Y for yes and press Enter. The SMS Class List should appear with '*DELETE' in the LINE OPERATOR column next to the deleted SMS class.

After deleting SMS classes, activate the modified SMS configuration to make your changes part of the active configuration.

Parent topic:[Maintaining the Storage Management Subsystem](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Converting volumes to SMS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can convert empty volumes through ISMF using the INIT line operator from the Volume application. You can convert previously-used, non-SMS-managed volumes using the CONVERTV command of DFSMSdss. You can also unconvert SMS volumes using the CONVERTV command of DFSMSdss.

See z/OS DFSMS Implementing System-Managed Storage for information on how to convert and unconvert volumes.

Parent topic:[Maintaining the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Deleting DASD volume residual data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Each member in a SMS SYSPLEX maintains its own DASD Volume SMS and MVS status in the configuration. If you remove or delete a SMS member from the SMS SYSPLEX, its DASD volume's SMS and MVS data is not deleted from the shared ACDS or COMMDS. To clear this residual data from the configuration, you must allocate a new ACDS and COMMDS and then replace the current ones by activating the updated SCDS, SETSMS SCDS(scds_name).

Parent topic:[Maintaining the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering Storage Management Subsystem information

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This topic describes how you can recover SMS information, including control data sets, source control data sets (SCDS), active control data sets (ACDS), communications data sets (COMMDS), and the SMS address space.

[Recovering control data sets](#)

[Recovering from a systems failure in the SMS complex](#)

[Recovering the SMS address space](#)

[Canceling the SMS address space](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering control data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Allocate a spare ACDS and COMMDS when you allocate your originals. Having spare copies eases the recovery process if SMS cannot access the originals because of I/O errors. Place the spares on a device that every system in your complex can access. Make certain that they reside on a different device from your originals.

In order to support 32 names, an SMS control data set is converted from 8-name mode to 32-name mode. You must confirm this conversion, using the operator console or ISMF.

Restrictions and Recommendations: If an SMS system temporarily goes down or has not yet been started with SMS, then it is not part of the system-managed storage environment. When you activate SMS on the system, it uses the ACDS and COMMDS that are specified in its IGDSMSxx member. If the COMMDS and ACDS are different from the ones used by the remainder of the SMS complex, then the system runs as a separate SMS complex. Consequently, when you change the current ACDS or COMMDS, update the IGDSMSxx member of SYS1.PARMLIB for each system in the SMS complex.

Make copies of your control data sets before the system mode is converted for the following reasons: The conversion is permanent. You might want to maintain them for coexistence with down-level systems. There are special considerations to keep in mind when you move control data sets and their catalog entries. See *z/OS DFSMS Managing Catalogs* for more information on special considerations, including the use of REPRO MERGECAT.

[Recovering an SCDS](#)

[Recovering an ACDS](#)

[Recovering a COMMDS](#)

Parent topic: [Recovering Storage Management Subsystem information](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering an SCDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For purposes of recovery, treat an SCDS the same as any other source VSAM linear data set. You can use DFMSHsm to manage its availability, which relieves you from having to allocate any spares.

If the SCDS and its backups are lost, you can save the ACDS as an SCDS using the SETSMS SAVESCDS command. For more information about how to use this command, see [Parameters of the SETSMS operator command](#).

Parent topic:[Recovering control data sets](#)



©Copyright IBM Corporation 1990, 2014

Recovering an ACDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can recover from errors that prevent access to the ACDS if you have allocated a spare. All permanent errors that make the ACDS unreadable or unwritable require intervention. For permanent I/O errors to the ACDS, the messages IGD0411 and IGD040D appear on the operator console. You have the option of retrying the failing operation or using the spare ACDS.

If you decide to use the spare ACDS, you need to perform three steps.
Reply 'C' on the operator console.

Copy the data from the SMS address space to the spare ACDS by issuing the following command from a system in the SMS complex: `SETSMS SAVEACDS(spare.acds)`

Tell the system to use the spare ACDS by issuing the following command: `SETSMS ACDS(spare.acds)`

You need only to issue the SETSMS ACDS(spare.acds) command on one system. The COMMDS is updated to reflect this change. As the other systems in the SMS complex access the COMMDS (based on the INTERVAL value in their respective IGDSMSxx members), they automatically switch to the new ACDS. Note: In addition to the method previously described, you can also use the SETSMS COPYSCDS command to create an ACDS from any valid SCDS. This command may be especially useful in cases where the created ACDS is to be used on another system, such as in disaster recovery situations. See *z/OS MVS System Commands* for more information on the SETSMS COPYSCDS command.

If your ACDS is corrupted, you should recover it from a good backup copy of the SCDS. Do not use the SETSMS SAVEACDS command after the error that corrupted the ACDS. To replace the ACDS:

Define a new linear VSAM data set with SHROPTNS(3,3), REUSE and more space for the new ACDS.

Issue the command `SETSMS ACDS(newacdsdsn),SCDS(scdsdsn)` to switch the sysplex over to the new ACDS.

Ensure PARMLIB member IGDSMSxx is updated to reflect the new ACDS so that it will be used upon the IPL of any system in the sysplex.

Note: If replacing the corrupted ACDS with a new ACDS does not solve the problem, try reallocating and activating both a new ACDS and a new COMMDS, using the procedure described above.

Parent topic:[Recovering control data sets](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering a COMMDS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you cannot access the COMMDS, you can recover from the error if you allocated a spare data set. All permanent errors that make the COMMDS unreadable require intervention. For permanent I/O errors to the COMMDS, the messages IGD041I and IGD070D appear on an operator console. Reply 'S' on the operator console and issue the following command from a system in the SMS complex:

`SETSMS COMMDS(spare.commds)`

One of three situations results.

If the spare COMMDS is empty it gets formatted automatically, and SMS writes the in-storage copy of the current COMMDS into the spare.commds. You then need to issue the following command on each of the remaining systems in the SMS complex:

`SETSMS COMMDS(spare.commds)`

If the spare COMMDS is not empty but describes an ACDS that is not currently active in the SMS complex, then SMS issues the message IGD076D. This message asks if you want to use the contents of the COMMDS and the ACDS to which it points. Reply 'C' to cause SMS to replace the contents of the spare.commds with the in-storage copy of the current COMMDS. You then need to issue the following command on each of the remaining systems in the SMS complex:

`SETSMS COMMDS(spare.commds)`

If the spare COMMDS is not empty but describes the ACDS that is currently active in the SMS complex, you need to issue the following command on each of the remaining systems in the SMS complex:

`SETSMS COMMDS(spare.commds)`

A response of 'S' to IGD070D is recommended when recovering from the current COMMDS because a response of 'C' might result in an unrecoverable error when trying to reaccess the current COMMDS. When access to the current COMMDS is suspended, SMS is able to access the new COMMDS without accessing the current COMMDS and resulting in further errors.

Without a usable COMMDS, the systems in the SMS complex have no means of communication. Other systems in the SMS complex are aware of the error, but they are unaware of the switch to a new COMMDS until you inform them.

If you can access the current COMMDS but you want to use an alternate one, you only need to issue the SETSMS command from one system. The other systems in the SMS complex detect the change from the old COMMDS to the new COMMDS and they automatically switch to the new one.

Parent topic:[Recovering control data sets](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering from a systems failure in the SMS complex

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To provide for recovery procedures, you should leave one system or system group name slots empty in the corresponding function panels. In case a problem occurs requiring that a system that is defined within the configuration as part of a system group be specified by its specific name (such as, hardware failure), you can add it without disrupting the rest of the system group of SMS complex.

You can initiate recovery with the following steps: Use the ISMF CDS Application to add the system name to the SMS configuration. Activate the configuration, either from ISMF or from the operator console. Use the VARY SMS operator command to make any necessary changes to the configuration. Remove the system name from the configuration and reactivate the configuration once the problem has been corrected.

Parent topic:[Recovering Storage Management Subsystem information](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering the SMS address space

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If the SMS address space fails, SMS automatically attempts to restart up to six times. If SMS fails to recover its address space after the sixth restart, you have two options:

You can end the SMS address space and then restart SMS using the T SMS=xx command. If you have cross memory interactions, users' address spaces wait for the SMS address space to restart. After the sixth automatic restart, SMS displays a message requesting the user to select the next action; retry another restart or terminate the SMS address space. This relates to the SMS address space and *not* the user's address space. You can allow SMS to attempt another restart by replying "YES" to the system-generated message: IGD032D

Parent topic:[Recovering Storage Management Subsystem information](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Canceling the SMS address space

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You cannot cancel SMS once it is activated. You must re-IPL to cancel the SMS address space and deactivate SMS. If the system is set up with automatic activation of SMS at IPL time and you want to deactivate SMS, you need to modify the appropriate SYS1.PARMLIB members to prevent automatic activation of SMS.

Parent topic:[Recovering Storage Management Subsystem information](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Protecting the Storage Management Subsystem

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To ensure that SMS operates correctly, you need to prevent unauthorized end users from modifying information in certain control data sets, such as SMS class, aggregate group, and storage group definitions. Some ISMF commands and functions also require protection from use by unauthorized users. This topic explains how to use Resource Access Control Facility (RACF®), a component of the Security Server for z/OS®, to establish authorization levels for protecting these data sets, commands, and functions.

Data set password protection does not apply to SMS-managed data sets or catalogs by SMS. SMS assumes the presence of RACF or an equivalent security product. Password protection is bypassed for all system-managed data sets.

[Identifying the resource owner and extracting the default classes](#)

[Protecting ISMF functions](#)

[Locating module names for ISMF applications](#)

[Locating module names for ISMF functions](#)

[Locating module names for line operators and commands](#)

[Protecting modules](#)

[Storage administration \(STGADMIN\) profiles in the FACILITY class](#)

[FIELD resource class](#)

[How authorization and protection of classes differ](#)



©Copyright IBM Corporation 1990, 2014

Identifying the resource owner and extracting the default classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The *resource owner* is the actual owner of the data set covered by the RACF® DATASET profile. RACF extracts the resource owner based on data set name. If the resource owner is not identified, the high-level qualifier is used.

If you specify ACSDEFAULTS(YES) in the IGDSMSxx member, RACF uses the resource owner to extract the default SMS classes and application identifier. If the resource owner is a user and no default SMS information is available from the user profile, the default information from the group profile is used. If the resource owner is a group, then the defaults for the group profile is used. You can protect the ability to update the resource owner field, RESOWNER, in the data set RACF profile. Revoked USERID should not be used as a resource owner, or it causes RACF to fail. See also [FIELD resource class](#).

If you specify USE_RESOWNER(NO) in the IGDSMSxx member, RACF uses the execution user ID instead of the resource owner to check authorization. This allows users who do not use a naming convention, user ID or group ID as the high level qualifier of data set names to check authorization to use storage and management classes. If you specify USE_RESOWNER(YES), there is no change to current processing. This is the default.

After ACS routines have been run, RACF is invoked to verify the user's authority to allocate the data set (CREATE/ALTER) and the resource owner's authority to use the STORCLAS and MGMTCLAS (READ). You can protect the ability of a resource owner to use management class and storage class through STORCLAS and MGMTCLAS resource classes.

Parent topic:[Protecting the Storage Management Subsystem](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Protecting ISMF functions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use RACF® authorization to limit access to the following categories of ISMF functions:

The entire ISMF component
The individual ISMF applications: ProfileData SetDASD
VolumeMountable Optical VolumeManagement ClassData ClassStorage
ClassStorage GroupAutomatic Class SelectionControl Data SetAggregate
GroupLibrary ConfigurationDrive ConfigurationData CollectionCopy

PoolListMountable Tape VolumeTape Library
The ISMF line operators
The ISMF commands
ISMF relies on the *RACF program control feature* to protect many of its applications. The RACF program control feature prevents unauthorized end users from running selected ISMF programs. To use the feature, you must activate the RACF Program Class and define your selected ISMF programs to RACF.

With RACF program control you can set up authorization levels for each of these categories, varying the level within a particular category to suit the needs of your installation. Individual end users can execute an ISMF function if one of the following conditions is true: They are authorized to execute the corresponding load module. Their RACF profile contains the OPERATIONS attribute. Their group is authorized to execute the load module. RACF is disabled or the program control feature is turned off. The universal access authority (UACC) for the load module is READ or greater, making the load module available to anyone who can access ISMF.

Recommendation: Protect these functions with RACF program control to make sure that only particular users can use the storage administrator applications and functions. Because a TSO/E user can change his user mode level, as this information is contained in the user's ISPF profile, protect the functions at a different level than user mode level.

The RACF program resource class allows the security administrator to protect various ISMF applications and functions with program control. This is achieved by controlling the access to load modules which are invoked by: ISMF Applications
ISMF Line Operators
ISMF Commands

The load modules reside in the following libraries: SYS1.DGTLIB for DFSMSdfp/ISMFSYS1.DGTLIB for DFSMSdss/ISMFSYS1.DFQLLIB for DFSMShsm

If the installation moves these modules to another load library, the installation-defined load library must be used in the program protection.

To protect a load module, use the RDEFINE RACF command. The syntax of this command is:

```
RDEFINE PROGRAM mod-name OWNER(owner of profile) +
UACC(NONE) ADDMEM('dsn of loadlib'/volser/NOPADCHK)
```

See z/OS Security Server RACF Security Administrator's Guide for a detailed description of how to use the RACF program control features.

Parent topic:[Protecting the Storage Management Subsystem](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Locating module names for ISMF applications

z/OS DFSMSdfp Storage Administration
SC23-6860-01

[Table 1](#) lists the load module names you can use to limit access to certain ISMF applications. The names are included in the panel coding for the ISMF Primary Option Menu for Storage Administrators, which is stored in the panel library with a member name of DGTSMMD2. If you want to limit access to all of ISMF, use module name DGTFMD01. [Table 1. Module Names for ISMF Applications](#)

Parent topic:[Protecting the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Locating module names for ISMF functions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Restricting access to the DGTFPF05 module prevents end users from gaining access to the Primary Option Menu for Storage Administrators. [Table 1](#) lists this and the other load module names you can use to limit access to certain ISMF functions.

Table 1. Module Names for ISMF Functions

Parent topic:[Protecting the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Locating module names for line operators and commands

z/OS DFSMSdfp Storage Administration
SC23-6860-01

[Table 1](#), [Table 2](#), [Table 3](#) and [Table 4](#) list the module names for ISMF line operators and commands. The names are stored in command tables in the DGTLIB load library. The line operators for storage administrators are described in [Quick reference to ISMF commands and line operators](#). Line operators for end users are described in z/OS DFSMS Using the Interactive Storage Management Facility.

Tip: You can invoke CATLIST or VTOCLIST from the Data Set List or outside of ISMF, but they are not ISMF line operators with ISMF load module names.

[Table 1. Module/CLIST Names for ISMF Line Operators, Part 1](#)

[Table 2. Module/CLIST Names for ISMF Line Operators, Part 2](#)

[Table 3. Module/CLIST Names for ISMF Line Operators, Part 3](#)

[Table 4. Module/CLIST Names for ISMF Commands](#)

If you want to look at the command tables, you need the data set name that your installation uses for the load library. The install process for ISMF puts the load modules in SYS1.DGTLIB (for DFSMSdfp/ISMF and DFSMSdss/ISMF), and SYS1.DFQLLIB (for DFSMShsm/ISMF). However, your installation's post-install procedures might involve moving the ISMF modules to other libraries. If the libraries have moved but have kept the same names, you can determine library data sets names by issuing the TSO/E LISTALC command and scanning the low-level qualifiers for DGTLIB and DFQLLIB.

Parent topic:[Protecting the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Protecting modules

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Take the following three program control steps to protect modules:

Use the RDEFINE command or the RACF® ISPF entry panels to identify the modules you want to protect. To define the modules to RACF, supply the name of the load module that you want to protect, the name of the data set that contains the load module, and the volume serial number of the volume that contains the data set. RACF adds each module that you identify to the profile for the PROGRAM general resource class. When you define the modules, you have several options: If you want to define several modules at the same time, you can use asterisk notation. For example, DGT* represents all of the modules beginning with the letters DGT. You can add an access list with user IDs or group names and their associated access authority to the profile. You can define the UACC to give default access to all users or to none. You can use the AUDIT parameter to set up or to bypass RACF logging.

Use the PERMIT command to allow end users to execute an application, line operator, or command associated with a module.

To prevent unauthorized users from copying a program, renaming it to a name that is unknown to the program control, and then executing the renamed program, you should protect the PDS libraries containing RACF-controlled programs with a UACC of NONE. In order for users to execute programs in these libraries, place the libraries in the LNKLIST concatenation. See z/OS Security Server RACF Security Administrator's Guide for more information.

Parent topic:[Protecting the Storage Management Subsystem](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Storage administration (STGADMIN) profiles in the FACILITY class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In order to control the ability to perform functions associated with storage management, define profiles in the FACILITY class whose profile names begin with STGADMIN (storage administration). Although some of these FACILITY profiles are used to protect the Storage Management Subsystem, they are checked whether or not you are using SMS.

There are two sets of FACILITY class profiles. The first are command and keyword related and the second are oriented towards actions and provide authority for storage administrators to do things like backup or copy data sets to which the administrator would usually have no authority.

[Command and keyword related profiles](#)

[Authority to activate a Storage Management Subsystem configuration](#)

Parent topic:[Protecting the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Command and keyword related profiles

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The RACF® FACILITY resource class can control the ability to: Activate a configurationPerform certain catalog functions on data sets using access method servicesPerform certain DFSMSdss functions on data sets

If defined, these profiles are checked before a user is allowed to perform the protected function. The user must have READ access. If these profiles are not defined, other RACF or password checking is still made to verify authority. Also, the user program must be Authorized Program Facility (APF)-authorized.

Exception: Password checking is bypassed if the function is performed on a system-managed data set.

Some FACILITY profiles are not checked if the caller is using the system key or is running in supervisor state. These profiles are:

STGADMIN.IGG.DEFNVSAM.NOBCSSTGADMIN.IGG.DEFNVSAM.NONVRSTGADMINDMIN.IGG.DELETE.NOSCRTCHSTGADMIN.IGG.DELGDG.FORCESTGADMIN.IGG.DELNVR.NOBCSCHKSTGADMIN.IGG.DIRCAT

In addition to the individual profiles, we recommend that the STGADMIN.* profile be defined with UACC NONE. Some STGADMIN profiles allow you to perform a specific function or use a specific keyword, but some functions or keywords can be used unless there is a STGADMIN profile preventing the usage of that keyword. By defining an STGADMIN.* profile with UACC NONE, these sensitive keywords can be protected and system exposures eliminated.

You need to define the following RACF profiles: **STGADMIN.DMO.CONFIG**This profile is used by BUILDIX command for Rapid Index

Rebuild.**STGADMIN.DPDSRN.olddsname**Controls the ability to rename a non-SMS-managed data set whose name is in use in another address space. You can regard them as having duplicate data set names. olddsname is up to 23 characters of the existing data set name. You can use a generic class name such as STGADMIN.DPDSRN.SYS2.*.

Recommendation: Do not give anyone authority to STGADMIN.DPDSRN.* because it is too broad. This option should be used with extreme caution. Very few people should have RACF authority to STGADMIN.DPDSRN.olddsname. Use this option only if you know the data set is not open on any system. For details of how to use this, see z/OS DFSMSdfp Advanced Services.

STGADMIN.IDC.BINDDATAControls the ability to use the access method services BINDDATA command.**STGADMIN.IDC.DCOLLECT**Controls the ability to use the access method services DCOLLECT

command.**STGADMIN.IDC.DIAGNOSE.CATALOG**Controls the ability to run the access method services DIAGNOSE command against

catalogs.**STGADMIN.IDC.DIAGNOSE.VVDS**Controls the ability to run the access method services DIAGNOSE command against a VVDS when a comparison against the BCS is performed. In this case, the BCS is

protected.**STGADMIN.IDC.EXAMINE.DATASET**Controls the ability to run the access method services EXAMINE command against integrated catalog facility catalog data sets.**STGADMIN.IDC.LISTDATA**Controls the ability to use the access method services LISTDATA

command.**STGADMIN.IDC.LISTDATA.ACCESSCODE**Controls the ability to use the access method services LISTDATA ACCESSCODE command. ACCESSCODE is a specialized LISTDATA command that requires an extra level of protection. Both levels (LISTDATA and LISTDATA.ACCESSCODE) are required.

STGADMIN.IDC.SETCACHEControls the ability to use the access method services SETCACHE command. This RACF profile does not include the following four profiles. A user must have SETCACHE access in order to have specific setcache command

authorization.**STGADMIN.IDC.SETCACHE.DISCARDPINNED**Controls the ability to use the access method services SETCACHE DISCARDPINNED

command. **STGADMIN.IDC.SETCACHE.PENDINGOFF** Controls the ability to use the access method services SETCACHE PENDINGOFF command. **STGADMIN.IDC.SETCACHE.REINITIALIZE** Controls the ability to use the access method services SETCACHE REINITIALIZE command. **STGADMIN.IDC.SETCACHE.SUBSYSTEM** Controls the ability to use the access method services SETCACHE SUBSYSTEM command. **STGADMIN.IFG.READVTOC.volser** Controls the ability to obtain READ access to the VTOC or VTOC index. **STGADMIN.IGD.ACTIVATE.CONFIGURATION** Controls the ability to activate an SMS configuration. **STGADMIN.IGG.ALTER.SMS** Controls the ability to alter the storage class and management class of a data set. If this profile is not created, the user must have RACF authority to the storage class and the management class to alter it. To use this profile, the administrator must have ALTER access to the data set whose storage or management class is to be changed. **STGADMIN.IGG.ALTER.UNCONVRT** Controls the ability to alter a system-managed VSAM data set to an unmanaged VSAM data set. **STGADMIN.IGG.DEFDEL.UALIAS** Controls the ability to define or delete an alias related to a user catalog without any other security authority. You can still define or delete an alias if you have alter authority to the catalog, even if you do not have read authority to this FACILITY class. **STGADMIN.IGG.DEFNVSAM.NOBCS** Controls the ability to define or alter a NVR for a data set without affecting the BCS entry if one exists. This profile is only checked by authorized services using the LOCATE macro, not by utilities like IDCAMS. **STGADMIN.IGG.DEFNVSAM.NONVR** Controls the ability to define or alter a BCS entry for a data set without affecting the VVDS entry if one exists. This profile is only checked by authorized services using the LOCATE macro, not by utilities like IDCAMS. **STGADMIN.IGG.DELETE.NOSCRTCH** Controls the ability to delete the BCS entry for a system-managed data set without deleting the data set itself (for example, using DELETE NOSCRATCH). This controls functions which uncatalog data sets. ➤ **STGADMIN.IGG.DELETE.RENAME** ➤ controls the ability to delete data set entries flagged as "rename in process". Attempts without the facility class for data sets flagged in this manner receive message IDC3009I with a return code of 90 and a reason code of 54. The "rename in progress" flag is ignored for users having RACF READ authority to the facility class and issuing a DELETE, and the entry is deleted. This facility class is intended for maintenance purposes. ➤ **STGADMIN.IGG.DELGDG.FORCE** Controls the ability to use DELETE FORCE on a generation data group which contains a system-managed generation data set. **STGADMIN.IGG.DELNVR.NOBCSCHK** Controls the ability to delete the VVDS entry (the NVR) for an system-managed non-VSAM data set without checking the BCS entry and catalog name for the data set. If there is a BCS entry or if the catalog name contained in the NVR does not match the catalog provided in the request, the function is denied unless the user has authority to this profile. **STGADMIN.IGG.DIRCAT** Controls the ability to direct a catalog request to a specific catalog, bypassing the normal catalog search. A directed catalog request is one in which the catalog name is explicitly passed to the catalog in the CATALOG parameter of access method services commands. In an SMS environment, all the catalog requests against system-managed data sets should be satisfied by the normal catalog search order. Directing the catalog request to a specific catalog requires authority to this profile with the exception of LISTCAT and EXPORT DISCONNECT requests. **STGADMIN.IGG.DLVVRNVR.NOCAT** Controls the ability to delete a VVR or NVR without an associated catalog. Users having RACF READ authority to the FACILITY class need no other RACF authority to the master catalog to perform the DELETE VVR or DELETE NVR functions. Attention: Restrict access to this

FACILITY class to users who understand the risk involved in deleting a VVR or NVR entry from a VVDS.

When a catalog is deleted for recovery purposes, or under certain failure conditions, an uncataloged VSAM data set or SMS nonVSAM data set can be left on the volume. The user can issue the DELETE VVR or DELETE NVR command to clean up the volume. In order to do this, the user needs RACF ALTER authority to the master catalog, and the user catalog must exist so that the catalog can be searched to verify that a BCS entry does not exist for the VVR or NVR. This is the usual situation when RACF ALTER authority to the catalog is needed. If the user catalog does not exist, the user must define an empty user catalog so that it can be searched.

The STGADMIN.IGG.DLVVRNVR.NOCAT FACILITY class allows the use of DELETE VVR or DELETE NVR without an associated user catalog. It does not require RACF authority to the master catalog for these commands.

STGADMIN.IGG.LIBRARYControls the ability to DEFINE, DELETE, or ALTER library and volume entries in a tape library.

STGADMIN.IGWSHCDS.REPAIRControls the ability to use the AMS SHCDS command functions, which you can use to list outstanding SMSVSAM recovery and control that recovery.

See z/OS DFSMSdss Storage Administration for more information on DFSMSdss functions.

Parent topic:[Storage administration \(STGADMIN\) profiles in the FACILITY class](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Authority to activate a Storage Management Subsystem configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The FACILITY resource class profile

STGADMIN.IGD.ACTIVATE.CONFIGURATION protects the ability to activate an SMS configuration. If you issue the ACTIVATE command from the Control Data Set Application Selection panel of ISMF, and either the Facility Resource Class is inactive or the named profile does not exist, the operator is queried to decide whether the ACTIVATE action should be allowed. You remain on the panel without the ability to provide additional input until action is taken from the operator console. If you have authority to activate an SMS configuration from ISMF, activation proceeds without confirmation from the operator console, and you continue with normal operations.

For more information on .DFSMSdss functions, see [z/OS DFSMSdss Storage Administration](#).

For listing of the Storage Administrator FACILITY Class Profiles for DFSMSdss, see [z/OS DFSMSdss Storage Administration](#).

Parent topic:[Storage administration \(STGADMIN\) profiles in the FACILITY class](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

FIELD resource class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The RACF® FIELD resource class controls the ability of users to specify or update the following RACF profile fields: Resource owner, RESOWNER, for data set profilesDefault SMS DATACLAS, STORCLAS, and MGMTCLAS values for user/group profilesApplication identifier, DATAAPPL for user/group profiles

Parent topic:[Protecting the Storage Management Subsystem](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

How authorization and protection of classes differ

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The *authorization* of SMS classes differs from the *protection* of SMS classes. For example, if you withhold authorization for a particular storage class, end users cannot specify that storage class in their JCL stream. End users specifying the storage class receive a JCL error if the ACS routine doesn't override the error. Protection involves more than using an SMS class, it involves changing one. All end users might have READ access to the data set containing a particular storage class, but only a select few have UPDATE authority. Also, only people with UPDATE authority to the SCDS can update the SMS classes.

[Authorizing the storage and management classes](#)

[Protecting the control data sets](#)

[Authorizing for TSO/E and ISPF](#)

[Storage administrator authorization](#)

Parent topic:[Protecting the Storage Management Subsystem](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Authorizing the storage and management classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can authorize the use of storage classes and management classes. Data classes do not require authorization.

Two of RACF's resource classes are: STORCLAS and MGMTCLAS. You authorize storage classes and management classes by defining them as RACF® profiles to the general resource classes. The two resource classes, storage class and management class, are distinct from SMS storage class and management class.

Parent topic:[How authorization and protection of classes differ](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Protecting the control data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You need to protect three types of SMS data sets: Control data sets SCDSACDSCOMMDSACS Routine Source Data Sets Partitioned data sets whose members are source ACS routines Sequential data sets that contain one source ACS routine ACS Routine test library Partitioned data sets whose members are ACS test cases.

You can issue SECURITY from the command line of the ISMF Data Set Application to protect these data sets. You can give universal read access to ACDS, COMMDS, and SCDS to anyone. Anyone who is authorized to update an SCDS or an ACS routine source data set is also authorized to change any items within these data sets.

Parent topic:[How authorization and protection of classes differ](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Authorizing for TSO/E and ISPF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To define an SMS data set under TSO/E you do not need APF authorization, but it is recommended because APF authorization reduces path length.

For the DEFINE and IMPORT commands, you can establish this authorization by adding the DEFINE and IMPORT command processor names to the installation access list APFCTABL in CSECT IKJEFTE2, or the parameter AUTHCMD NAMES in SYS1.PARMLIB member IKJTSO00. For a call of access method services, you can establish this authorization by adding IDCAMS to the installation access list APFPPTABL in CSECT IKJEFTE8 or the parameter AUTHPGM NAMES in SYS1.PARMLIB member IKJTSO00. For more information about specifying authorized commands and programs under TSO/E, see *z/OS TSO/E Customization*.

To define or import an SMS data set while under option 6 of ISPF/PDF, you must include the DEFINE and IMPORT processor names in the command table in ISPTCM. See *z/OS ISPF Planning and Customizing* for more information.

Parent topic:[How authorization and protection of classes differ](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Storage administrator authorization

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Storage administrators should receive the following authority from the security administrator: CLAUTH to STORCLAS and MGMTCLAS resource classesAuthority to activate an SMS configurationAuthority to perform exceptional catalog and DFSMSdss operations on system-managed data setsAuthority to all ISMF applications, dialogs, line operators, and commandsUpdate authority to the RESOWNER field in the DATASET profilesUpdate authority to the DATAAPPL, MGMTCLAS, STORCLAS, and DATACLAS, fields in the USER/GROUP profilesAuthority to alter SMS constructs. It might be preferable to have a group profile with authorities mentioned above for all storage administrators.

Parent topic:[How authorization and protection of classes differ](#)



©Copyright IBM Corporation 1990, 2014

Administering VSAM record-level sharing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

VSAM record-level sharing (RLS) is a data set access mode that allows multiple address spaces, CICS® application owning regions (AORs) on multiple MVS systems, and jobs to access data at the same time. With VSAM RLS, multiple CICS systems can directly access a shared VSAM data set, eliminating the need for function shipping between application owning regions (AORs) and file owning regions (FORs). CICS provides the logging, commit, and rollback functions for VSAM recoverable files; VSAM provides record-level serialization and cross-system caching. CICS, not VSAM, provides the recoverable files function. Note: VSAM RLS requires that the data sets be System Managed Storage (SMS) data sets. To be eligible for RLS, a data set that is not already SMS-managed must be converted to SMS.

This topic assumes you are familiar with the concepts of VSAM RLS and how it uses the coupling facility (CF) cache and lock structures. It describes the following tasks to enable and maintain VSAM RLS:

[Preparing for VSAM record-level sharing](#)
[Defining CF cache structures in the SMS base configuration](#)
[Defining storage classes for VSAM RLS](#)
[Activating VSAM RLS](#)
[Monitoring the coupling facility for VSAM RLS](#)
[Recovering VSAM RLS processing](#)
[Falling back from VSAM RLS processing](#)
[See z/OS DFSMS Using Data Sets](#) for more information on VSAM RLS. For information on the online Parallel Sysplex® Configuration Assistant, go to : <http://www.ibm.com/s390/psos/psotool>

[Preparing for VSAM record-level sharing](#)

[Defining CF cache structures in the SMS base configuration](#)

[Defining storage classes for VSAM RLS](#)

[Defining VSAM RLS attributes in data classes](#)

[Activating VSAM RLS](#)

[Monitoring the coupling facility for VSAM RLS](#)

[Recovering VSAM RLS processing](#)

[Falling back from VSAM RLS processing](#)

[Terminating the SMSVSAM address space](#)

[Quiescing or enabling a secondary lock structure](#)

[Deleting a VSAM RLS lock structure](#)

[Displaying information about a secondary lock structure](#)



Preparing for VSAM record-level sharing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Planning for and installing VSAM RLS requires coordination with system hardware and software groups. This section describes the tasks that the storage administrator must do with the MVS systems programmer and the CICS® database administrator to prepare to use VSAM RLS:

[Determining hardware requirements](#)
[Understanding the product environment for VSAM RLS](#)
[Determining applications that can use VSAM RLS](#)
[Ensuring same systems connectivity](#)
[Planning for availability](#)
[Defining sharing control data sets](#)
[Defining CF cache structures](#)
[Defining the primary CF lock structure](#)
[Modifying the SYS1.PARMLIB IGDSMSxx member](#)
[Establishing authorization for VSAM RLS](#)
[Using dssTimeOut](#)

Parent topic: Administering VSAM record-level sharing



©Copyright IBM Corporation 1990, 2014

Determining hardware requirements

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Recommendation: Use multiple CFs with global connectivity. This ensures maximum availability, simplifies systems management and allows for nondisruptive lock transfer in the event of a CF outage.

You must have at least one CF connected to all systems capable of VSAM RLS within the Parallel Sysplex® (for lock structure duplexing, you must have at least two CFs). For multiple CFs, select one facility with global connectivity to contain the master lock structure. For maximum availability, use a second CF with global connectivity. In the event a CF becomes unavailable, the SMSVSAM address space can transfer its in-storage duplexed copy of the locks to the other available CF without causing any application disruption.

You can attach CFs which do not contain the master lock structure to a subset of the systems. A CF cache structure that is referenced by a storage class cache set must have at least the same connectivity as the storage groups to which the storage class maps.

The following hardware requirements are necessary to utilize system-managed lock structure duplexing: CFLEVEL = 10 CF architecture supportCFCC licensed internal code implementationCF-to-CF peer linksA duplexed pair of primary and secondary VSAM RLS lock structure instancesMessage architecture LIC support for each VSAM RLS system image that has a connector to a duplexed lock structure Depending upon your existing configuration, you might also need additional CF processor, storage, and link capacities to utilize system-managed duplexing.

Parent topic:[Preparing for VSAM record-level sharing](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Understanding the product environment for VSAM RLS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

VSAM RLS processing involves support from multiple products, such as CICS®, CICSVR, and DFSMS. See z/OS Migration for information on coexistence of different releases supporting VSAM RLS.

Parent topic:[Preparing for VSAM record-level sharing](#)

 | 

©Copyright IBM Corporation 1990, 2014

Determining applications that can use VSAM RLS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Applications using VSAM RLS benefit from increased data availability inherent in a shared environment that has read/write integrity and record-level locking (as opposed to control interval (CI) locking).

Your applications should fall into one of the following categories:

- VSAM RLS-tolerant application** The application runs correctly in a multi-update environment when RLS is specified in the JCL or when MACRF=RLS is specified on the ACB. The RLS JCL parameter allows batch read programs to use RLS without requiring a recompile of the program. For batch update programs running against nonrecoverable VSAM RLS spheres, it might be possible to modify the allocation from DISP=OLD to DISP=SHR.
- VSAM RLS-exploiting application** The application recognizes when a VSAM data set can be shared at the record level and uses VSAM RLS functions to access the data. CICS® applications are exploiting applications.
- VSAM RLS-intolerant application** The application uses facilities not supported by VSAM RLS, accesses VSAM internal data structures, or is incompatible with the functions of VSAM RLS. For example, the application might perform CI access against the data set.

However, because VSAM RLS can be used by CICS and non-CICS applications, the CICS recoverable files function provides transactional recovery for applications, VSAM RLS is expected to be used primarily by CICS applications. Transactional recovery isolates the changes made by each sharing application. CICS creates a backout log record for each change made to a recoverable file, and VSAM RLS obtains and holds a lock on each changed record. The lock remains held until the transaction ends. If a transaction fails, CICS backs out all changes made by the application to recoverable files, thus isolating the other sharing applications from the failure.

For recoverable data set, in addition to the data sharing across CICS applications, VSAM RLS enables read-with-integrity sharing by batch jobs. Batch jobs can share recoverable files while they are being modified by CICS applications. This is possible because VSAM provides the record locking and buffer coherency functions across CICS and batch. Because VSAM RLS does not provide the transactional recovery function for batch jobs, it does not allow a batch job to open a recoverable data set for output.

VSAM RLS permits read and write sharing of nonrecoverable data sets across CICS and batch jobs. Transactional recovery does not apply to nonrecoverable data sets. While VSAM RLS permits sharing, the jobs must be carefully designed to achieve correct results in a read/write data sharing environment, because they do not have the isolation provided by transactional recovery.

Parent topic:[Preparing for VSAM record-level sharing](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Ensuring same systems connectivity

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You must ensure same systems connectivity for CF cache structures, lock structure, and storage groups. This is to ensure that jobs running in the Parallel Sysplex® have access to data in both the CFs and storage groups. The lock structure must have global connectivity to all systems in the Parallel Sysplex.

[Figure 1](#) shows how connectivity among systems and storage groups matches connectivity among systems and coupling facilities.

Figure 1. Example of Global Connectivity in a Parallel Sysplex

[Figure 2](#) illustrates the problems that can occur if you do not have global connectivity across the Parallel Sysplex: [Figure 2. Example of Insufficient Connectivity in a Parallel Sysplex](#)

In [Figure 2](#), System 3 is connected to both Storage Group 2 and to Coupling Facility 2. This meets the minimum connectivity requirements, but can still cause problems. Because Storage Group 2 data can be placed in Coupling Facility 1, jobs in System 3 can fail if they attempt to access data sets in Storage Group 2 that have been assigned to Coupling Facility 1. You can avoid this by ensuring that, for all systems in the Parallel Sysplex, connectivity among systems and storage groups matches connectivity among systems and coupling facilities.

Parent topic:[Preparing for VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Planning for availability

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In order for VSAM RLS processing to take place, you must ensure the following:
Run all systems performing RLS as a Parallel Sysplex®Define and activate at least two sharing control data sets (SHCDS), and one spare SHCDS for recovery purposesDefine CF cache and lock structures to MVS, using the CF resource manager (CFRM) policy. Define CF cache structures in the SMS base configuration. Associate CF cache set names with storage class definitions, and write ACS routines to associate storage class definitions that map to CF cache structures with data sets. Change the attributes for a data set to specify whether the data set is to be recoverable or nonrecoverable. Specify LOG(NONE) if the data set is nonrecoverable; specify LOG(UNDO) or LOG(ALL) if the data set is recoverable. For more information about using the CFRM policy to define CF structures, see z/OS MVS Setting Up a Sysplex. For more information about specifying recoverable and nonrecoverable data set attributes, see z/OS DFSMS Using Data Sets.

Parent topic:[Preparing for VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining sharing control data sets

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Sharing control is a key element in maintaining data integrity in a shared environment. Because persistent record locks are maintained in the CF, several new classes of failure might occur, such as a Parallel Sysplex®, system, or SMSVSAM address space restart, or a CF lock structure failure. The sharing control data set (SHCDS) is designed to contain the information required for DFSMS to continue processing with a minimum of unavailable data and no corruption of data when failures occur. The SCDS data can be either SMS or non-SMS managed.

The SHCDS acts as a log for sharing support. It is a logically-partitioned linear data set, with CISIZE of 4096, that must be defined with secondary extents, though all extents for each data set must be on the same volume. An SHCDS contains the following information: The name of the CF lock structure in useA list of subsystems and their statusA list of open data sets using the CFA list of data sets with unbound locksA list of data sets in permit non-RLS state

Sharing control is critical to maintaining data integrity in the event of the failures such as Parallel Sysplex, system, or SMSVSAM address space restart, or of the CF lock structure. You should always have at least two active and one spare SHCDSS. Place these data sets for maximum availability. If necessary, you can have up to five active SHCDSS and five spare SHCDSS.

Recommendation: Because the contents of these data sets are highly dynamic, do not use backup and restore functions, because these might cause the loss of VSAM RLS data set recovery information.

Consider the following as you allocate and maintain your SHCDSS: The SHCDS must not be shared outside of the Parallel Sysplex. Refer to the restrictions in [Running SMS in a parallel sysplex environment](#). Allocate SHCDSS so that the number of active and spare data sets ensures the data is always duplexed. At a minimum, define and activate two SHCDSS and at least one spare SHCDS for recovery purposes. You should ensure that there are enough spare SHCDSS, because these are used when I/O errors occur on the active SHCDSS. Place the SHCDSS on volumes with global connectivity. VSAM RLS processing is only available on those systems that currently have access to the active SHCDS. The share options for SHCDSS must be set to (3,3) so that each system in the Parallel Sysplex can properly share the data sets. Place your SHCDSS in such a way as to maximize availability in the event of the loss of a volume. Use storage classes defined with the guaranteed space attribute. Avoid placing SHCDSS on volumes for which there might be extensive volume reserve activity. SMSVSAM should be authorized to update SYS1.DFPSCDS.* data sets. If you protect SYS1.* data sets, be sure SMSVSAM is able to access SYS1.DFPSCDS.* for update. Ensure that the space allocation for active and spare SHCDSS is the same. Ensure data sets have enough space for growth. Ensure that the SHCDS has secondary extents defined. Use the VARY SMS,SHCDS command to activate and maintain your SHCDSS.

You must use the following naming convention when defining your SHCDSS:

SYS1.DFPSCDS.qualifier.volser

where: **qualifier** is a 1 to 8 character qualifier. **volser** is the volume serial number. The **V** prefix allows you to specify numeric volume serial numbers.

Define the SHCDS with the following characteristics: CISIZE equals 4096 Share options are (3,3) Contains secondary extents Resides on a single volume Use the following formula to calculate the size of the primary extent of your SHCDS:
space = (number_of_recoverable_data_sets * number_of_recoverable_subsystems * 2 * 272) bytes
or 13MB, whichever is larger

where: **number_of_recoverable_data_sets** is the average number of recoverable data sets open across the sysplex. **number_of_recoverable_subsystems** is the average number of recoverable subsystems registered with RLS across the sysplex.
For example, if you have an average of 10 recoverable subsystems registered with

RLS with an average of 1000 recoverable data sets open, the amount of space required for the primary extent is (10 * 1000 * 2 * 272) or 5440000 bytes (5313 KB). You can create an SHCDS using access method services. When you use the access method services DEFINE command to create an SHCDS, specify SHAREOPTIONS(3,3) to ensure that the SHCDS can be written to and read from any system. Specify CISIZE as 4096. Select a volume that is a member of a storage class with the guaranteed space attribute.

Figure 1 shows how to create an SHCDS using IDCAMS:

Figure 1. Example for Creating Sharing Control Data Sets Using IDCAMS

```
/* ALLOCATE SHCDS ON XP0301 - SXPXXS04 IS GUARANTEED SPACE
/*
//ALLOCLD1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DEFINE CLUSTER (NAME(SYS1.DFPSHCDS.ACTIVE.VXP0301)
    LINEAR
    CISZ(4096)
    STORCLAS(SXPXXS04)
    SHAREOPTIONS(3 3)
    CYL(10 5)
    VOLUME(XP0301) )
/*

```

Once the SHCDS data sets have been created, you make them available for use by using the VARY SMS, SHCDS (qualifier.Vvolser), NEW command. The (qualifier.Vvolser) is used, not the fully qualified SHCDS name. For example, VARY SMS, SHCDS (ACTIVE,XP0301). This command only needs to be entered on one system in the Parallel Sysplex. The SMSVSAM address space on the system where the command is entered communicates the name of the data set to the other SMSVSAM address spaces in the Parallel Sysplex. Each SMSVSAM address space recatalogs the data set, if it has not already been cataloged, so that you do not need to manually catalog the data set in order for it to be used. Those data sets added for use are saved, and can be accessed when an SMSVSAM address space initializes them or is restarted, or though subsequent IPLs.

The names of the SHCDS data sets are stored in the Sysplex COUPLE data sets. In the event of starting up with a new set of Sysplex couple data sets, the SHCDS data sets names are lost and you must use the VARY SMS SHCDS command to make them available.

To delete and redefine a SHCDS data set, you must first delete the SHCDS data set from SMSVSAM by entering the V SMS.SHCDS(shcdsname).DELETE command, and then you can use IDCAMS DELETE and DEFINE to modify the new SHCDS data set. You must do that even if SMSVSAM is not active. Note, that you always must have at least two active SHCDS data sets and one spare SHCDS data set. To delete a SHCDS data set when there are only the minimum number of SHCDS data sets defined, you must first define additional SHCDS data sets before you can delete the existing SHCDS data sets.

The following example shows how to delete and redefine SHCDS data sets when only the minimum number of SHCDS data sets are defined:

If the following SHCDS data sets are already defined:

Name	Size	%UTIL	Status	Type
TOOSMAIL.VXP0301	7200Kb	5%	GOOD	ACTIVE
TOOSMAIL.VXP0302	7200Kb	5%	GOOD	ACTIVE
WRONGVOL.VXP0201	7200Kb	5%	GOOD	SPARE

Enter these commands to define three new data sets:

```
V SMS,SHCDS(JUSTRITE.VXP0301),NEW
V SMS,SHCDS(RIGHTVOL.VXP0202),NEWSpare
```

These are the resulting SHCDS data sets:

Name	Size	%UTIL	Status	Type
TOOSMAIL.VXP0301	7200Kb	5%	GOOD	ACTIVE
TOOSMAIL.VXP0302	7200Kb	5%	GOOD	ACTIVE
JUSTRITE.VXP0301	14400Kb	2%	GOOD	ACTIVE

JUSTRITE.VXP0302	14400Kb	2%	GOOD	ACTIVE
WRONGVOL.VXP0201	7200Kb	5%	GOOD	SPARE
RIGHTVOL.VXP0202	7200Kb	5%	GOOD	SPARE

Enter these commands to delete the old data sets:

```
V SMS,SHCDS(TOOSMAIL.VXP0301),DELETE
V SMS,SHCDS(WRONGVOL.VXP0201),DELETE
```

The resulting SHCDS data sets are as follows:

```
IGW612I 16:45:49 DISPLAY SMS,SHCDS SYSTEM1
d sms,shcds
Name          Size    %UTIL Status Type
JUSTRITE.VXP0301 14400Kb  2%   GOOD  ACTIVE
JUSTRITE.VXP0302 14400Kb  2%   GOOD  ACTIVE
RIGHTVOL.VXP0202 7200Kb   5%   GOOD  SPARE
```

Note: If you swap in a new set of SHCDS data sets or make changes to an existing SHCDS data set, you must communicate the changes to SMSVSAM through the VARY SMS SHCDS command. Do not move a SHCDS data set from one volume to another. The SHCDS data set naming convention depends on the volume in which the SHCDS data set resides.

For more information see [Changing the state of coupling facility cache structures and volumes](#).

Parent topic: [Preparing for VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining CF cache structures

z/OS DFSMSdfp Storage Administration
SC23-6860-01

CF cache structures must be defined to MVS and also in the SMS base configuration. CF cache structures provide a level of storage hierarchy between local memory and DASD cache. They are also used as a system buffer pool for VSAM RLS data when that data is modified on other systems. Each CF cache structure is contained in a single CF. You might have multiple CFs and multiple CF cache structures.

Several factors determine the number and size of your CF cache structures:

Number of available CFs
Amount of space available in each CF
Amount of data to be accessed through each CF
Continuous availability requirements for CF reconfiguration
Performance requirements for various applications

You use CFRM policy definitions to specify an initial and maximum size for each CF cache structure. DFSMS uses the initial structure size you specify in the policy each time it connects to a CF cache structure. If additional space is required, RLS manages the altering of the cache, up to the maximum size specified. Do not specify ALLOWAUTOALT(YES) for RLS cache structures, which would allow system-initiated alters, thus preventing RLS from being given control to manage the cache structure.

You can assign one or more CF cache structures to each cache set associated with a storage class. Having multiple cache sets allows you to provide different performance attributes for data sets with differing performance requirements.

When more than one CF cache structure is assigned to a cache set, data sets are assigned to each CF cache structure in an effort to balance the load.

[Sharing CF structures](#)

[Determining CF cache structure size](#)

[Using the RLS_MAX_POOL_SIZE parameter to limit local buffer pool size](#)

Parent topic:[Preparing for VSAM record-level sharing](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sharing CF structures

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can share CF structures among applications. However, this results in all data being treated equally, which may not be desirable. If you want to treat all your RLS data equally, sharing CF structures might be an appropriate and relatively simple approach. However, if you want to give some data sets preference over other data sets, you might want to use several CF structures of different sizes. For example, you might place test data sets in a small structure and production data sets in a large structure. Or, to give CICS® data preference over DFMSHsm data, place the CICS data in a larger structure and DFMSHsm data in a smaller structure.

Parent topic:[Defining CF cache structures](#)



©Copyright IBM Corporation 1990, 2014

Determining CF cache structure size

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A CF cache structure must be at least large enough to hold all of the MVS information required to describe a structure of maximum size. To help you achieve the best possible performance with VSAM RLS buffering, the sum total of all the CF cache structure sizes you define (the CF cache) should ideally be the sum total of the local VSAM local shared resources (LSR) buffer pool sizes. The size of the local VSAM LSR buffer pool is the sum of LSR pool size and, if used, the corresponding Hiperspace™ pool size. You can run VSAM RLS with less CF cache storage than this, but the CF cache *must* be large enough for the CF cache directories to contain an entry for each of the VSAM RLS local buffers across all instances of the SMSVSAM server. If the CF cache cannot contain the directory entries describing the local buffers, then the VSAM RLS local buffers are falsely invalidated and must be refreshed. To minimize this, the minimum CF cache structure size should never be less than 1/10 the size of the local buffer pool.

For example, the following CICS® FOR configuration shows the sum total of the local VSAM RLS buffer pool size prior to migrating to VSAM RLS.

When migrating this configuration to VSAM RLS, the CF cache you define should ideally be at least 170MB. In this way, cross-invalidated local RLS buffers can be refreshed from the CF cache structures.

Performance should improve when the CF cache is larger than the sum of the local VSAM LRS buffer pool sizes. When the CF cache is smaller, performance depends upon the dynamics of the data references among the systems involved. In some cases, you might want to consider increasing the size of very small CF caches (2 MB - 10 MB).

In those situations where you can determine that data previously treated as a nonshared resource (NSR) is no longer to be treated as such, you should also include NSR buffer sizes in the total local buffer pool size.

Parent topic:[Defining CF cache structures](#)



Using the RLS_MAX_POOL_SIZE parameter to limit local buffer pool size

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the RLS_MAX_POOL_SIZE parameter of the IGDSMSxx parmlib member to limit the maximum size of the local buffer pools. The value you specify can be either larger or smaller than the default maximum pool size of 100 MB, but it must be supported with the available real and expanded storage. Although VSAM RLS can in some cases ascertain how much buffer space is supported before paging begins to occur, specifying an RLS_MAX_POOL_SIZE value ensures that the local buffer pool does not grow beyond the value you specify.

If RLS_Max_Pool_Size is greater than 1500M, SMS will pass a value of 9999. The value will then be reset to 1728M. A default upper limit is set and this internal upper limit can be changed.

Tip: In some instances, the local buffer pool might *temporarily* grow larger than the RLS_MAX_POOL_SIZE value. Setting a RLS_MAX_POOL_SIZE value that is too low might result in unnecessarily degrading the local hit rate.

Recommendation: Initially, set the RLS_MAX_POOL_SIZE value to 50% more than the sum of the local buffers on a single system. The following table illustrates this, assuming that each CICS® FOR is on a separate system:

If you have multiple applications using RLS, the RLS_MAX_POOL_SIZE value needs to take into account the requirements for all of the applications.

You can use the information from the SMF type 42 record with subtype 19 to evaluate the local buffer hit rates for each of the individual systems. Use this information together with the local system paging rates to help you make additional tuning adjustments to the RLS_MAX_POOL_SIZE parameter.

Parent topic:[Defining CF cache structures](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining the primary CF lock structure

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Requirements: To use VSAM RLS, you must define a single, master CF lock structure. If, however, you use the DUPLEX attribute, you only define one CF lock structure. That is because the structure gets created automatically into the secondary CF.

For maximum availability, define nonvolatile lock structures. CF lock structures enforce the protocol restrictions for VSAM RLS data sets and maintain the record-level locks and other DFSMSdfp serialization. Ensure that the CF lock structures have universal connectivity, so that they are accessible from all systems in the Parallel Sysplex® that support VSAM RLS. For system-managed duplexing, define the lock structure as duplexed (DUPLEX (ENABLED) or DUPLEX (ALLOWED) in the CFRM policy).

For more information about system-managed lock structure duplexing, see [Recovering the CF lock structure](#).

The Primary CF lock structure is named IGWLOCK00. Use the XCF coupling definition process to define it. To estimate its size requirements in megabytes, use the following formula (a megabyte is 1048576 bytes in this case):

$$\text{lock_entry_size} = 10M * \text{number_of_systems} *$$

where: **number_of_systems** is the number of systems in the Parallel Sysplex **lock_entry_size** is the size of each lock entry. This value depends on the MAXSYSTEM value that is specified to the IXCL1DSU Couple Data Set format utility. Use the following table to determine the actual lock entry size for the different MAXSYSTEM setting values: Table 1. Effect of MAXSYSTEM Value on Lock Table Entry Size

[Table 2](#) shows some sample lock allocation estimates:

Table 2. CF Lock Structure Sizing Examples

These lock size estimates include the memory requirements for both the lock table and the record-lock memory. Use these estimates as rough initial values to help you attain a locking structure with a desired false contention target of approximately one-half of 1% or less. Contact your marketing representative for help in arriving at an initial estimate that more closely matches your specific configuration. Note: XES rounds up the number of lock entries to the next highest power of two. For example, if you base your calculations on $2 * 10M = 20M$ locks, XES allocates the structure with 32M locks.

The primary DFSMS CF lock structure, IGWLOCK00, has persistent connections and is a persistent structure. To delete this structure you must use the operator command:

`V SMS ,SMSVSAM ,FORCEDELETELOCKSTRUCTURE`

[Considerations for retained locks and record table full conditions](#)

[Avoiding false contention](#)

[Adjusting the lock structure size](#)

[A lock structure sizing example](#)

Parent topic: Preparing for VSAM record-level sharing



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Considerations for retained locks and record table full conditions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The CF lock structure includes two parts: a lock table, used to determine whether there is R/W interest among systems on a particular resource record table space to keep track of information for retained locks and spheres which have been processed by VSAM RLS

If a commit protocol application such as CICS® fails, the locks protecting updates against recoverable spheres are remembered in the record table space until CICS performs the required backouts. Also, update locks associated with indoubt transactions are remembered until the indoubts are resolved.

When used record table space reaches 80% or greater, informational message IGW326I is issued. A shortage of record table space can occur for the following reasons: The size of the lock structure is too small for normal system operation. A CICS system has failed and cannot be successfully restarted to run its backouts. As a consequence, record table space cannot be freed up. Outstanding indoubt transactions exist. CICS provides commands to display indoubt transactions and resolve them. Transactions are in backout failed state. This means that the backouts could not complete and free record table space.

Because it might be difficult to remedy these situations quickly, you should respond to a record table shortage by modifying the CFRM policy to increase the size of the lock structure, in order to provide additional record table space and increase the space available for retained locks. Then, activate that policy and rebuild the lock structure by operator command. The operator command can be used to change the size of the lock structure.

You can increase record table space by carefully selecting the total lock memory allocation quantity. Record table space is guaranteed to never be less than one-half of the total lock space allocated to RLS, but it can be more if the total size is not a power of two. A lock space allocation of 32 MB causes 16 MB to be allocated to the lock table and 16 MB to be allocated to the record lock table. An allocation of 63 MB, on the other hand, causes 16MB to be allocated to the lock table, and all of the remaining memory (for example, 47 MB) is then allocated to the record table space. Thus, selecting a total lock size that is not a power of two value is a means of causing the record space to grow without necessarily increasing the table size. When you attempt to change the size of the lock structure using either the rebuild or the alter function and the record table size is greater than 50%, VSAM RLS will determine whether the new table has enough space for all existing record table entries, plus enough empty space for future locking processes (120%). If the answer is no, message IGW322I is displayed. A rebuild request is rejected; an alter request proceeds.

If the record table becomes full, VSAM requests which require that a lock be recorded receive RPL feedback, and CICS backs out the transaction. A record table full condition means that transactions cannot complete successfully, and the size of the lock structure *must* be increased.

Parent topic:[Defining the primary CF lock structure](#)



Avoiding false contention

z/OS DFSMSdfp Storage Administration
SC23-6860-01

VSAM RLS assigns locked resources to an entry value in the lock table, and uses this entry value to quickly check whether a resource is already locked. If the lock structure (and thus the lock table) is too small, many locks can be represented by a single value, making "false" lock contention possible. False lock contention occurs when two different locks on different resources attempt to use the same lock entry. The second lock requester is suspended until VSAM RLS determines that there is no real lock contention on the resource. False contention can be a problem for workloads with heavy R/W interests among systems.

To avoid false contention, you need to consider the size of the lock table. The lock table size is determined by the total size of the lock structure. When you define the size of the total lock structure, you should specify a value that is a power of two in order to maximize the lock table size: the lock table comprises 50% of the total space, and the record lock space the remaining 50%. Any memory in excess of a power of two value is allocated to the record lock space exclusively until the next power of two value is reached. At that time, the lock table space is doubled, and the two allocations are once again of equivalent sizes.

VSAM RLS uses the MAXSYSTEM value from the Couple Data Sets format utility to determine the size of each lock entry. Because smaller lock entry sizes imply a larger number of locks for the same memory allocation, it is very important to select an appropriate MAXSYSTEM value. The MAXSYSTEM value represents the maximum number of systems that can be connected into the Parallel Sysplex®. There is a lock table memory penalty when the MAXSYSTEM setting exceeds 7, and another penalty when it exceeds 23. Consequently, from a false contention point of view you want to select a MAXSYSTEM value that does not exceed 7 or 23.

To increase a MAXSYSTEM value that has already been specified, you must first format larger CDSs and switch them into use dynamically. Then, either manually rebuild the CF lock structure using the SETXCF START command or start system-managed duplexing rebuild using the SETXCF START,REBUILD,DUPLEX command.

To decrease a MAXSYSTEM value that has already been specified, you must shut down all of the VSAM RLS address spaces, and manually delete both the persistent connections and the lock structure. Then you must restart the VSAM RLS address spaces. If you decrease the MAXSYSTEM value without first shutting down, the decrease has no effect; the Parallel Sysplex continues to run using the old MAXSYSTEM value.

[Monitoring for false contention](#)

[How much contention is acceptable](#)

[Reducing false contention](#)

Parent topic:Defining the primary CF lock structure

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Monitoring for false contention

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can determine the amount of false contention by using either the resource measurement facility (RMF) or the DISPLAY SMS,CFLS command.

Parent topic:[Avoiding false contention](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

How much contention is acceptable

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For the best performance, you want to achieve the least possible amount of global lock contention, both real and false. The amount of real lock contention is application-dependent; it depends on record access patterns. False lock contention is almost entirely determined by the size of the lock table, with a larger lock table having less false lock contention than a smaller one. A good goal is to have total (real and false) global lock contention of less than one percent. The false contention component of the total global lock contention should be less than one-half of one percent, and ideally, should be substantially less than this.

Parent topic:[Avoiding false contention](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Reducing false contention

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If false contention becomes a problem, try the following: Reduce the amount of real lock contention in your applications, if possibleSpecify a larger size for the lock structure and manually rebuild itEnsure the MAXSYSTEM parameter of the Couple Data Set utility is not too large for the number of members in your Parallel Sysplex® A MAXSYSTEM value of 7 or less allows you twice as many lock entries as a MAXSYSTEM value of 8, as shown in [Table 1](#).

Parent topic:[Avoiding false contention](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Adjusting the lock structure size

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Once you select an initial lock structure size and the Parallel Sysplex® has been running with that size for some time, you should monitor the percentage of false contentions. You can then use this percentage to help you select an even more appropriate lock structure size.

Use the following formula to determine the estimated lock structure size based on a false contention percentage:

$$\text{Minimum Lock Structure Size} = F * M / T$$

where: **F** is the measured false contention percentage, expressed as a percentage (for example, a rate of 0.02 = 2%, so $F = 2$)
M is the current lock structure allocation size
T is the target false contention target percentage, expressed as a percentage (for example, a rate of 0.005 = 0.5%, so $T = 0.5$)

Parent topic:[Defining the primary CF lock structure](#)



©Copyright IBM Corporation 1990, 2014

A lock structure sizing example

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Suppose a Parallel Sysplex® has a MAXSYSTEM setting of 3, with two systems currently connected to the CF. You could use the initial sizing formula to estimate its initial size, as shown below:

Initial Lock Structure Size = $10M * 2 * 2$

where: number_of_systems = 2
lock_entry_size = 2 bytes This yields an initial lock structure size of 40MB. However, to maximize the lock table space itself, you should size it with a number that is a power of two. In this case, the initial total lock size could be set at either 32 or 64 MB.

For the purpose of this example, we select 32 MB as the initial lock structure size. We then run for a while before determining the rate of false contentions. Assuming that this false contention rate is 1.5%, and that we have a target false contention rate of 0.5%, we can then use the following formula to modify the lock structure size:

Minimum Lock Structure Size = $1.5 * 32MB / 0.5$

where: 1.5 is the measured false contention rate of 1.5%
32 MB is the specified lock structure size
0.5 is the target false contention rate of 0.5% The adjusted size should now be 96 MB. However, because this needs to be expressed as a power of two, unless we actually select a value of at least 128 MB, it is unlikely that we will find that the false contention rate meets or exceeds our target of 0.5%.

In this example, the Parallel Sysplex would run better with a larger lock structure size than initially allocated. On the other hand, had the monitored false contention rate been less than an initial target value of 0.5%, it would not mean that we were wasting CF storage by having it allocated to the lock structure. In fact, although a false contention rate of 0.1% is not feasible in many cases, it is still ideal, assuming that it can be achieved with a reasonable amount of CF memory.

Parent topic:[Defining the primary CF lock structure](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Modifying the SYS1.PARMLIB IGDSMSxx member

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The IGDSMSxx parmlib member includes the following parameters to support the CF and VSAM RLS processing: CF_TIME, which aligns creation of all the CF-related SMF type 42 records with subtypes 15, 16, 17, 18 and 19 DEADLOCK_DETECTION, which specifies the interval for detecting deadlocks between systemsRLSINIT, if YES, specifies whether the SMSVSAM address space is started as part of system initialization or by the V SMS,SMSVSAM,ACTIVE command. If you specify NO, to start SMSVSAM later, you must do the following: change RLSINIT to YES, issue the SET SMS=xx command, then IPL or issue the V SMS,SMSVSAM,ACTIVE command. The default is NO.RLS_MaxCfFeatureLevel, which specifies the method VSAM RLS caching uses to determine the size of the data that is placed in the CF cache structure. RLS_MAX_POOL_SIZE, which specifies the maximum size of the SMSVSAM local buffer pool RlsAboveTheBarMaxPoolSize, which specifies the maximum amount of virtual storage above the 2-gigabyte bar that can be used for VSAM RLS buffering.RlsFixedPoolSize, which specifies the amount of real storage (both above and below the 2-gigabyte bar) to be dedicated to VSAM RLS buffering.SMF_TIME, which aligns the SMF type 42 records for DFSMS (with subtypes 2, 15, 16, 17, 18 and 19) to the SMF_TIME interval USEEAV, which specifies at the system level, whether SMS is to select extended address volumes during volume selection processing. This check applies to new allocations and when extending data sets to a new volume. You can enter an operator command to change the value of this keyword. **NOT**This is the default. SMS does not select extended address volumes during volume selection. Setting USEEAV to NO has the following implications: Data sets might still exist on extended address volumes in either the track-managed or cylinder-managed space of the volume. In addition, when you specify or default to USEEAV=NO, the system skips EAV volumes for the following functions to avoid the failure of DFMSHsm requests: RecallMigration to an ML1 or ML2 DASD volumeBackupRecoverARECOVERDaily backup volume / backup volume spill processWhen a system where USEEAV is set to NO backs up a data set from an EAV, the data set will be recovered to a non-EAV volume and the vendor attributes will be lost. The system issues message ARC0784I in this case.

YESSMS is to use extended address volumes to allocate new data sets or to extend existing data sets to new volumes.**Note:** When SMS is not active, USEEAV is not available and the installation must use alternate means to control the usage of EAVs.

BreakPointValue, which specifies the number of cylinders (0-65520) that SMS uses to select volumes for VSAM data sets. The default is 10. When SMS is not active, the system assigns the default value. You can modify these parameters at any time during VSAM RLS processing. For more information, refer to [Changing IGDSMSxx parameters to support the coupling facility](#). For more information about the IGDSMSxx member, refer to the topic about [IGDSMSxx](#) in z/OS MVS Initialization and Tuning Reference.

Parent topic:[Preparing for VSAM record-level sharing](#)



Establishing authorization for VSAM RLS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To establish authorization to access VSAM RLS resources, assign a RACF attribute of PRIVILEGED or TRUSTED to the VSAM RLS server address space, SMSVSAM. With PRIVILEGED, most RACROUTE REQUEST=AUTH macro instructions done for SMSVSAM are considered successful, without any checking being performed. The checking done for the CHKAUTH operand on the RACROUTE REQUEST=DEFINE macro instruction is also bypassed. All other RACF processing occurs as usual. RACF does not:Call any exit routinesGenerate any SMF recordsUpdate any statistics.

TRUSTED is similar to PRIVILEGED. Most RACROUTE REQUEST=AUTH macro instructions that are done for SMSVSAM are considered successful, without any checking being performed. RACF does not:Call any exit routinesUpdate any statistics. RACF does generate SMF records that are based on the audit options specified in SETROPTS LOGOPTIONS and the UAUDIT setting in the USER ID profile.

If the VSAM RLS server address space is neither PRIVILEGED nor TRUSTED, grant the SMSVSAM server the appropriate access authorization:

Add SMSVSAM with the STARTED attribute if you are using a started task group. Authorize SMSVSAM for update access to SYS1.DFPSCDS.* data sets. If you protect SYS1.* data sets be sure SMSVSAM is able to access SYS1.DFPSCDS.* for update. If you protect volumes that contain RLS-accessed data then authorize SMSVSAM for update access to the volume profiles. To use the access method services SHCDS command, you must be authorized to the STGADMIN.IGWSHCDS.REPAIR resource in the FACILITY class. The SHCDS command lists SMSVSAM recovery associated with subsystems and spheres, and controls that recovery. You should also ensure that only those users who need the capability, such as CICS subsystems, have access to register a subsystem name to SMSVSAM. Use the RACF subsystem name class to restrict this access. For more information, refer to CICS Transaction Server for z/OS Release Guide, at [CICS Transaction Server for z/OS Information Center](#).

Parent topic:[Preparing for VSAM record-level sharing](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using dssTimeOut

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Specifies the number of seconds that DFMSMSdss will wait during backup processing for quiesce data set requests to complete. Specify a value from zero to 65536 seconds (which is more than 18 hours). If you specify a value between 1 and 299 seconds, the system uses a value of 300 seconds (which equals 5 minutes). The value specified in the DSSTIMEOUT parameter value is activated when the first instance of the SMSVSAM address becomes active in the sysplex. All subsequent SMSVSAM instances will use the same value.

You can alter the DSSTIMEOUT value dynamically in the following ways:
Using the SETSMS DSSTIMEOUT(nnnn) commandBy adding or updating the DSSTIMEOUT parameter in IGDSMSxx parmlib member and then activating it with the SET SMS=xx commandThe default is 0.

Parent topic:[Preparing for VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

Defining CF cache structures in the SMS base configuration

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In order for DFSMSdfp to use the CF for VSAM RLS, after you define one or more CF cache structures to MVS, you must also add them in the SMS base configuration.

To add CF cache structures to the base configuration, you associate them with a cache set name. This cache set name is also specified in one or more storage class definitions. When a storage class associated with a data set contains a cache set name, the data set becomes eligible for VSAM record-level sharing and can be placed in one of the CF cache structures associated with the cache set.

The system selects the best cache structure in which to place the data set.

Refer to [Defining CF cache structures](#) and [Defining the primary CF lock structure](#) for more information on using CFRM policies to define CF cache and lock structures.

Perform the following steps to define CF cache structures to DFSMS.

Select option 8, Control Data Set, from the ISMF Primary Option Menu for Storage Administrators, to display the Control Data Set (CDS) Application Selection panel.

Supply values on the CDS Application Selection panel: **CDS Name**Specify the name of the SCDS that is to contain the base configuration for VSAM RLS. The CDS Name field is primed with the last used SCDS name. This name might differ from the SCDS name of your base configuration for VSAM RLS.**Option**Select option 7, Cache Update. This displays the CF Cache Set Update panel.

Define your CF cache sets on the CF Cache Set Update panel. The panel allows you to define up to 256 CF cache sets. Each CF cache set can have up to eight CF cache structure names assigned to it. Use the cache structure names you defined in the MVS CFRM policies. [Figure 1](#) shows how CF cache structures can be defined in multiple cache sets, and also shows how data sets associated with a storage class definition containing a cache set name can be placed in multiple cache structures.

Figure 1. Example of CF Cache Structure Definition in Base Configuration

Parent topic:[Administering VSAM record-level sharing](#)



Defining storage classes for VSAM RLS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This section describes how to assign the cache set names defined in the base configuration to a storage class, so that data sets associated with that storage class can be eligible for VSAM RLS and use CF cache structures. It also describes how to indicate the relative importance of the data associated with the storage class.

Requirement for JES3 users: In a JES3 environment, be careful to define cache set names only in those SMS storage classes that are used by data sets opened for VSAM RLS processing. When you define a cache set name in a storage class, any job accessing a data set associated with that storage class is scheduled on a VSAM RLS-capable system. If all storage classes have cache set names defined for them, then all jobs accessing SMS-managed data sets are scheduled to VSAM-RLS-capable systems. This could cause a workload imbalance between those systems and down-level systems.

To assign the CF cache sets defined in the base configuration to storage classes, perform these steps.

Select option 5, Storage Class, from the ISMF Primary Option Menu for Storage Administrators. This displays the Storage Class Application Selection panel.

Supply values on the Storage Class Application Selection panel: **CDS Name** is the name of the SCDS you defined for VSAM RLS in the CDS Name field.

Option Select option 3, Define. This displays the Storage Class Define panel.

Press the DOWN key to view the second page of the Storage Class Define panel.

Supply values on the Storage Class Define panel: **CF Cache Set Name** is the name of the CF cache set you defined in the base configuration. **CF Direct Weight or CF**

Sequential Weight Specify a weight attribute for the data in one of the fields, to indicate the data's relative importance. The default is a weight value of 6. Note: DFSMS only supports the default value. Regardless of what you specify, all data eligible for VSAM record-level sharing is assigned a weight value of 6.

Parent topic: [Administering VSAM record-level sharing](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Defining VSAM RLS attributes in data classes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can define data classes specifically for data sets eligible for VSAM RLS. When you have data classes with VSAM RLS parameters, you do not need to change current AMS DEFINE statements and job streams.

Recommendation: To avoid having too many data classes, you should specify the JCL LGSTREAM ID keyword for SMS VSAM data sets defined by JCL instead of the Logstream ID attribute in the data class. See [Defining data class attributes](#) for more information.

Parent topic:[Administering VSAM record-level sharing](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Activating VSAM RLS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The SMSVSAM address space always starts at IPL time, providing RLSINIT(YES) is specified in the IGDSMSxx parmlib member. VSAM RLS processing is available once certain minimum requirements are met. This section describes those requirements.

[Enabling VSAM RLS processing](#)

[Enabling a data set for VSAM RLS processing](#)

Parent topic:[Administering VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

Enabling VSAM RLS processing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

VSAM RLS processing is available once the following requirements are met: All systems are running as a Parallel Sysplex®. At least two SHCDSs, and one spare SHCDS have been activated. At least one existing CF cache structure is defined to MVS and to the SMS base configuration. If a subset of the cache structures is not available, some data sets might not be accessible.

The CF lock structure IGWLOCK00 is available. The SMS address space is started. If a null configuration is active, only existing data sets already assigned to a CF cache structure can be processed.

If VSAM RLS processing is not enabled, all attempts to open a data set where VSAM RLS is specified on the ACB or the JCL fail.

SYSVSAM must not be in the RNLDEF exclude list for any data sets accessed by RLS.

Under the following conditions, VSAM RLS access from a system is *not* available:
No SHCDSs are available. The CF lock structure, IGWLOCK00, is not available. The SMSVSAM address space has failed and a response to message IGW418D is pending. The SMSVSAM address space has failed, and manual restart is in effect (the response to message IGW418D was C). None of the CF cache structures defined in the SMS configuration are available. If a subset of CF cache structures are not available, data sets bound to those structures might not be accessible from this system.

Parent topic:[Activating VSAM RLS](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Enabling a data set for VSAM RLS processing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For a data set to be opened for VSAM RLS processing, VSAM RLS processing must be available, and the LOG parameter must be specified on the DEFINE CLUSTER or the ALTER CLUSTER command for the data set. Additionally, if the data set is currently assigned to a CF cache structure, that cache structure must be available. If the data set is not currently assigned to a CF cache structure, a cache set must be specified on the storage class, and at least one of the associated CF cache structures must be available.

Parent topic:[Activating VSAM RLS](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Monitoring the coupling facility for VSAM RLS

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This section describes:

- Displaying information about the CF
- Altering the size of CF cache structures
- Altering the size of the CF lock structure
- Changing the IGDMSxx parameters that support the CF
- Altering the Status of CF Cache Structures and Volumes
- Selecting Data Sets for CF Statistical Monitoring

[Displaying CF information](#)

[Changing the size of coupling facility cache structures](#)

[Changing the size of coupling facility lock structures](#)

[Changing IGDMSxx parameters to support the coupling facility](#)

[Changing the state of coupling facility cache structures and volumes](#)

[Selecting data sets for coupling facility statistical monitoring](#)

Parent topic:[Administering VSAM record-level sharing](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying CF information

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use MVS operator commands as well as ISMF to request CF information. Use the MVS DISPLAY XCF command to display information about CFs, connections to the CF, and CFRM policies. Use the DISPLAY SMS command or ISMF applications to request information about DFSMS CF structures and SHCDSs.

The following sections describe using the DISPLAY SMS command and ISMF applications in greater detail.

[Using the DISPLAY SMS command](#)

[Using ISMF](#)

Parent topic:[Monitoring the coupling facility for VSAM RLS](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using the DISPLAY SMS command

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The syntax for the DISPLAY SMS command is shown in [Figure 1](#). [Figure 1.](#)

```
DISPLAY SMS,SMSVSAM[,ALL]
      ,CFCACHE(CF_cache_structure_name|*)
      ,CFLS
      ,CFVOL(volume_serial_number)
      ,MONDS(specification_mask|*)
      ,QUIESCE
      ,SEP
      ,SHCDS
```

The following list describes the various forms of the DISPLAY SMS command and what information is displayed: DISPLAY SMS,SMSVSAMdisplays the status of lock structures connected to the system from which the command is entered.

DISPLAY SMS,SMSVSAM,ALLdisplays the status of all lock structures in the sysplex. If the lock structure is duplexed, composite state information is also displayed.

Specify ALL to see the status of all SMSVSAM servers.

DISPLAY SMS,CFCACHE(CF_cache_structure_name|*)displays information about CF cache structures. Specify CFCACHE(*) to request information for all CF cache structures. Specify a specific cache structure name to display information about only that cache structure.

DISPLAY SMS,CFLSdisplays information about the CF lock structure. This information includes the lock rate, lock contention rate, false contention rate, average number of requests waiting for locks, the lock structure size, and primary structure information. If the lock structure is in duplex mode, secondary structure information will also be displayed.

DISPLAY SMS,CFVOL(volser)displays a list of CF cache structures containing data for the volume specified. Also displays the CF_VOLUME status.

DISPLAY SMS,MONDS(specification_mask|*)Specify MONDS(*) to view all the data set specifications eligible for CF statistics monitoring. Use a specification mask to view only a subset of those specifications. You can specify a full or partial data set name, and you must specify at least one high level qualifier. A wild card in the data set name cannot be followed by additional qualifiers.

DISPLAY SMS,SMSVSAM, QUIESCEdisplays active QUIESCEs on the system on which it is issued.

DISPLAY SMS,SEPdisplays the name of the data set separation profile, if one is specified.

DISPLAY SMS,SHCDSdisplays information about SHCDSS. This information includes SHCDS names, sizes and the amount of free space for all the active SHCDSS and their status. It also includes the names of all the spare SHCDSS.

Parent topic:[Displaying CF information](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using ISMF

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the ISMF Data Set, Volume and Storage Class applications to display CF information. You can also use ISMF to view the cache sets defined for a specific CDS and the cache structure names associated with each cache set.

[Data set application](#)

[Volume application](#)

[Storage class application](#)

[Control data set application](#)

Parent topic:[Displaying CF information](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data set application

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the Data Set Selection Entry panel to specify certain attributes and request a list of corresponding data sets and their characteristics. The data sets must be open for VSAM RLS access in order to appear on the list. If you specify CF attributes, ISMF displays the following CF information for each data set matching your criteria:

- CF status indicator Indicates whether the data set is in use by RLS processing, whether a forward recovery of the data set is in progress, and whether the sphere has been quiesced for RLS processing.

- CF monitor status Indicates whether CF cache structure statistical monitoring is on or off.

- CF cache structure name Specifies the name of the CF cache structure in which the data set is stored.

- CF cache set name Specifies the name of the CF cache set with which the data set is associated.

You can also use these values as attributes on the Data Set Filter, View, and Sort panels to customize the data set list created by ISMF.

Parent topic:[Using ISMF](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Volume application

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the Volume Selection Entry panel to specify certain attributes and request a list of corresponding volumes and their characteristics. You can request that ISMF display CF volume status information, indicating one of the following conditions:

- The volume is enabled for VSAM RLS processing and can be associated with a CF cache structure
- VSAM RLS processing is finishing and no new data can be placed in CF cache structures
- No VSAM RLS data for the volume exists in any of the CF cache structures and no CF cache structures can be assigned to the volume

You can also use the CF volume status attribute on the volume filter, view, and sort panels to further customize the volume list created by ISMF.

Parent topic:[Using ISMF](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Storage class application

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the Storage Class Application Selection panel to request that ISMF display storage class names and CF cache set names associated with a specific CF cache structure name. You can also request to see the CF direct or sequential weights assigned to a specific storage class definition.

In addition, you can use the CF cache set name, CF direct weight, and CF sequential weight attribute on the storage class filter, view, and sort panels to further customize the storage class list created by ISMF.

Parent topic:[Using ISMF](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Control data set application

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the CDS Class Application Selection panel to request that ISMF display CF cache structure names for all CF cache sets defined for a specific SCDS.

Parent topic:[Using ISMF](#)



©Copyright IBM Corporation 1990, 2014

Changing the size of coupling facility cache structures

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DFSMS uses the initial structure size specified in the CFRM policy each time it connects to a CF cache structure. In each case, the ALTER(YES) keyword is specified, indicating to MVS that this structure can be dynamically reconfigured. To alter the size of a CF cache structure, issue the SETXCF START command, using the following format:

SETXCF START,ALTER,STRNAME=CF_cache_structure_name,SIZE=newsize

where: **CF_cache_structure_name** is the name of the cache structure being altered. **newsize** is the new structure size in megabytes. This new size can be larger or smaller than the size of the current CF cache structure, but it cannot be larger than the maximum size specified in the CFRM policy.

MVS automatically starts the alter process in place, without disruptions to the application using the CF cache structure. The alter function does not cross CF boundaries and does not take the place of the rebuild function. If you require a larger structure size than that specified in the CFRM policy, you must activate a new CFRM policy and rebuild the structure. The alter function changes the cache structure in place; you can use the rebuild function to move a cache structure to another CF. See [Defining CF cache structures](#) for information on estimating the size of CF cache structures.

Parent topic:[Monitoring the coupling facility for VSAM RLS](#)



Changing the size of coupling facility lock structures

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the rebuild function to change the size of a lock structure by rebuilding it using a new size. If the new size is too small, then the rebuild is stopped and message IGW322I is displayed. You can then start the rebuild process again using the size recommendations displayed in message IGW322I.

You can also issue the SETXCF START command to change the size of the lock structure. Each time it connects to a CF lock structure, DFSMS uses the initial structure size specified in the CFRM policy. In each case, either the ALTER(YES) or AUTOALTER(YES) keyword is specified, indicating to MVS that this structure can be dynamically reconfigured.

To alter the size of a CF lock structure, issue the SETXCF START command, using the following format:

`SETXCF START,ALTER,STRNAME=CF_lock_structure_name,SIZE=newsize`

where: **CF_lock_structure_name** is the name of the lock structure being altered. **newsize** is the new structure size in megabytes. This new size can be larger or smaller than the size of the current CF lock structure, but it cannot be larger than the maximum size specified in the CFRM policy, and less than its minimum size set by the coupling facility.

MVS automatically starts the alter process in place, without disruptions to the application using the CF lock structure.

If VSAM RLS determines that the lock table space or record table space that results from the alter function will be too small for future locking activity, VSAM RLS issues message IGW322I and continues with the alter function. In this case, you must issue a new alter using the size recommendations displayed in message IGW322I.

The alter function does not cross CF boundaries and does not take the place of the rebuild function. If you require a larger structure size than that specified in the CFRM policy, you must activate a new CFRM policy and rebuild the structure. The alter function changes the lock structure in place; you can use the rebuild function to move a lock structure to another CF.

Parent topic:[Monitoring the coupling facility for VSAM RLS](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Changing IGDSMSxx parameters to support the coupling facility

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The SYS1.PARMLIB IGDSMSxx member includes several parameters that support the coupling facility. With the exception of RLSINIT, these parameters apply across all systems in the Parallel Sysplex®. The parameter values specified for the first system that was activated in the sysplex are used by all other systems in the sysplex.

The following SYS1.PARMLIB IGDSMSxx member parameters support VSAM RLS:

CF_TIMEDILOCK_DETECTIONRLSINITRLS_MaxCfFeatureLevelRLS_MAX_POOL_SIZE
RLSIsAboveTheBarMaxPoolSizeRLsFixedPoolSizeSMF_TIME

You can change the values for these parameters during VSAM RLS processing only when all the spheres are closed . The new values are then used by all systems in the Parallel Sysplex, except for RLSINIT parameter values which are used only by the system accessing the changed parmlib member when SMSVSAM is next started.

To change the values of these parameters, use one of the following methods:
Issue the SETSMS operator command, specifying the parameter with different values.Issue the T SMS=xx command, where xx identifies an IGDSMSxx member where the parameter values are different than those currently in use.

The following list describes the parameters and their values:

{CF_TIME(nn|3600)} Indicates the number of seconds between recording SMF type 42 records with subtypes 15, 16, 17, 18, and 19 for the CF (both cache and lock structures). You can specify a value from 1 to 86399 (23 hours, 59 minutes, 59 seconds). The default is 3600 (one hour). This keyword sets the interval time for the following SMF 42 subtypes: **SUBTYPE 15**CF storage class average response time**SUBTYPE 16**CF data set average response time**SUBTYPE 17**CF lock structure activity**SUBTYPE 18**CF cache partition summary**SUBTYPE 19**SMSVSAM least recently used statistics summary

{DEADLOCK_DETECTION(iii|15,ffff|4)} Specifies the deadlock detection intervals used by the Storage Management Locking Services. **iii**specifies the length in seconds of the local deadlock detection interval, as a one to four digit numeric value in the range 1-9999. The default is 15 seconds.**ffff**specifies the number of local deadlock cycles that must expire before global deadlock detection is run, as a one to four digit numeric value in the range 1-9999. The default is 4 cycles.**[RLSINIT({NO|YES})]**Specify YES if you want the SMSVSAM address space started as part of system initialization or the V SMS,SMSVSAM,ACTIVE command. This value applies only to the system accessed by the parmlib member and is acted upon when SMSVSAM is next started. The default is NO.

[RLS_MaxCfFeatureLevel({A|AMx|Z})]Specifies the method that VSAM RLS uses to determine the size of the data that is placed in the CF cache structure. If you specify A, caching proceeds using the RLSCFCACHE keyword characteristics that are specified in the SMS data class that is defined for the VSAM sphere.

With feature level A the user can also specify Mx values in the form of AMx to customize IGW500I messages for open spheres as follows: **M0**IGW500I messages for all open spheres will be suppressed.**M1**IGW500I messages for all open spheres will be displayed.**M2**Only for the first sphere opened in the sysplex will an IGW500I message be displayed.**M3**IGW500I messages for all open spheres in the sysplex will be sent to hardcopy.**M4**Only for the first sphere opened in the sysplex will an IGW500I message be sent to hardcopy.

If you do not specify a value, or if you specify Z, then only VSAM RLS data that have a Control Interval (CI) value of 4K or less are placed in the CF cache structure. The default is Z.

Restrictions: The restrictions are described below. If A is specified for the RLS_MaxCfFeatureLevel parameter, systems lower than z/OS® V1R3 will not be able to connect to the CF cache structure.If a lower-level system is the first system

activated in the sysplex, RLS_MaxCfFeatureLevel defaults to Z, and all systems will be able to connect to the CF cache structure. If the SETSMS command is used to change the RLS_MaxCfFeatureLevel value to A on a mixed-level system, the command is rejected and message IGW500I is issued. The Mx modifier can only be combined with feature level A in the form of AMx, otherwise an IGW467I will be issued to indicate that the Parmlib value specified is not valid. If the SETSMS command is used to change the RLS_MaxCfFeatureLevel value to Z when there are VSAM spheres assigned to cache structures, the command is rejected and message IGW500I is issued.

[RLS_MAX_POOL_SIZE({nnnn|100})] Specifies the maximum size in megabytes of the SMSVSAM local buffer pool. SMSVSAM attempts to not exceed the buffer pool size you specify, although more storage might be temporarily used. Because SMSVSAM manages buffer pool space dynamically, this value does not set a static size for the buffer pool. Use SMF 42, subtype 19 records to help you determine the maximum size of the SMSVSAM local buffer pool.

You can specify a two to four-digit numeric value, with 10 as the minimum value. If you specify a value less than 10, the field is set to 10. If you specify a value greater than 1500, SMSVSAM assumes there is no maximum limit. We recommend that you limit the size of the local buffer pool.

The default is **100 MB**.

[RIsAboveTheBarMaxPoolSize{(sysname, maxrls; ...)|(ALL,maxrls)}] Specifies the maximum amount of virtual storage above the 2-gigabyte bar that can be used for VSAM RLS buffering in a system in the sysplex, where: sysname identifies a system and maxrls is specified in megabytes for that system. Multiple sets of sysname, maxrls pairs may be specified, separated by semicolons. (ALL,maxrls) specifies that each system in the sysplex will have the same specified value. The default for maxrls is 0, and the specifiable values are 500 to 2000000 (that is, 500MB to 2 terabytes). **[RIsFixedPoolSize{(sysname, maxrls; ...)|(ALL,maxrls)}**]

Specifies the total amount of real storage (both above and below the 2-gigabyte bar) to be dedicated to VSAM RLS buffering in a system in the sysplex, where: sysname identifies a system and maxrls is specified in megabytes for that system. Multiple sets of sysname, maxrls pairs may be specified, separated by semicolons. (ALL,maxrls) specifies that each system in the sysplex will have the same specified value. The default for maxrls is 0, and the specifiable values are from 0 to 80% of the total unpinned available real storage. When 80% is reached, the RLS address-space SMSVSAM Initialization will issue a message to warn of the limit.

[SMF_TIME({YES|NO})] Specifies that the following SMF type 42 records are created at the SMF interval time, and that all of the indicated records are synchronized with SMF and RMF data intervals. **SUBTYPE 2** Cache control unit statistics (IBM® 3990-3, 3990-6, IBM RVA, and IBM ESS)**SUBTYPE**

15Coupling facility storage class average response time**SUBTYPE 16**Coupling facility data set average response time**SUBTYPE 17**Coupling facility lock structure activity**SUBTYPE 18**Coupling facility cache partition summary**SUBTYPE**

19SMSVSAM least recently used statistics summary This allows the customer to merge these SMF records for a specified time period and obtain both the 'system' view and the 'user' view of activity in the interval. YES is the default. DFSMS creates the specified SMF record when the interval period expires and SMF sends the event notification signal.

If you specify YES, SMF_TIME overrides the following IGDSMSxx parameters: CACHETIME, CF_TIME.

Parent topic:Monitoring the coupling facility for VSAM RLS

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Changing the state of coupling facility cache structures and volumes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the VARY SMS command to control processing for volumes, data sets, or systems.

To alter the state of the specified CF cache structure, issue the VARY SMS command, using the following format:

`VARY SMS,CFCACHE(cf_cache_structure_name),ENABLE|QUIESCE`

When a CF cache structure is enabled, VSAM RLS data can be stored in the cache structure. This is the normal state of operations, and is the state the CF cache structure is in after the Parallel Sysplex® has been IPLed. When a CF cache structure is quiesced, no VSAM RLS data can be stored in it.

To alter the state of the specified volume as it relates to all CF cache structures, issue the VARY SMS command using the following format:

`VARY`

`SMS,CFVOL(volser),ENABLE|QUIESCE`

When a volume is CF-enabled, data contained on this volume can be stored in a CF cache structure. in the cache structure. This is the normal state of operations. When a volume is CF-quiesced, no data contained on it can be stored in a CF cache structure.

Use the VARY SMS CFVOL command if it is necessary to modify a volume without using the VSAM PUT/ERASE macros or DFSMSdss. This ensures that when the modified volume is again made available for VSAM RLS processing, CF cache structures do not contain downlevel data.

Rule: Setting a volume to the CF-quiesced state does not stop SMS from selecting this volume during data set allocation. To stop SMS from selecting this volume, issue the VARY SMS command, using the following format:

`VARY`

`SMS,VOLUME(volser),DISABLE`

Parent topic:[Monitoring the coupling facility for VSAM RLS](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Selecting data sets for coupling facility statistical monitoring

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the VARY SMS command to specify which data sets are eligible for CF statistical monitoring. If statistical monitoring is on, SMF TYPE 42, subtype 16 records are produced. Use the following command format:

VARY
SMS,MONDS (dsname{dsname,...}),{ON|OFF}

Select OFF to indicate that the specified data sets are no longer eligible for statistical monitoring. You can specify a full or partial data set name, with at least one high-level qualifier. An asterisk cannot be followed by other qualifiers. You can specify up to 16 data set specifications with each command. This command affects activity for the specified data sets across all systems in the Parallel Sysplex®.

Parent topic:[Monitoring the coupling facility for VSAM RLS](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering VSAM RLS processing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This section describes: Recovering the CF lock structure
Recovering a CF cache structure
Recovering the SMSVSAM address space
Recovering an SHCDS

[Recovering the CF lock structure](#)

[Recovering a CF cache structure](#)

[Recovering the SMSVSAM server address space](#)

[Recovering a shared control data set](#)

Parent topic:[Administering VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering the CF lock structure

z/OS DFSMSdfp Storage Administration
SC23-6860-01

At VSAM open time, the SMSVSAM address space checks to ensure that the CF lock structure is available. Record-level sharing cannot occur if the CF lock structure is unavailable or has failed.

[User-managed lock structure rebuild](#)

[System-managed lock structure duplexing rebuild](#)

Parent topic:[Recovering VSAM RLS processing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

User-managed lock structure rebuild

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A CF lock structure might fail and need to be rebuilt if the CF lock structure named IGWLOCK00 does not exist, or if it is not connected to the system attempting to open a VSAM data set for record-level sharing. In either case, DFSMS internally initiates a rebuild queueing all applications using VSAM RLS during the rebuild process. The rebuild is transparent to these applications.

If both the CF lock structure and the system fail, all recoverable data sets which were open for VSAM RLS processing at the time of failure are converted to "lost locks". The data sets become unavailable to any processing besides recovery processing (such as backouts), and no new sharing is allowed until the recovery processing is complete. A new CF lock structure must be available to perform the recovery processing.

In this case, you can redefine or replace the CF lock structure, correct the problem causing its unavailability, or move the work requiring VSAM RLS to another system which has connectivity to an available CF lock structure.

Parent topic:[Recovering the CF lock structure](#)



©Copyright IBM Corporation 1990, 2014

System-managed lock structure duplexing rebuild

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Using the system-managed duplexing rebuild function, you can create and maintain a duplex (secondary) copy of the lock structure in case of failure. When the structure is in duplex mode and a failure occurs, VSAM RLS switches from the failed lock structure to the active lock structure, and record-level sharing can proceed. The lock structure reverts to simplex mode, and continues to operate as it would in user-managed mode (see [User-managed lock structure rebuild](#).)

When the lock structure is in simplex mode and DUPLEX(ENABLED) is specified for the lock structure in the CFRM active policy, the system attempts to start a system-managed duplexing rebuild if the environment allows. When DUPLEX(ALLOW) is specified, you must use SETXCF START,REBUILD,DUPLEX to manually initiate a system-managed duplexing rebuild.

You can stop system-managed duplexing manually using SETXCF STOP,REBUILD,DUPLEX or by specifying DUPLEX(DISABLED) in the CFRM active policy.

Restriction: When the lock structure is in duplex mode, user command rebuild requests are rejected. You must use the alter function to change the lock structure. For more detailed information about the system-managed duplexing rebuild function, refer to z/OS MVS Programming: Sysplex Services Guide.

Parent topic:[Recovering the CF lock structure](#)



Recovering a CF cache structure

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In the event a CF cache structure fails, DFSMS attempts to rebuild it so that it remains available to the data with which it is associated. A CF cache structure is also rebuilt if DFSMS detects a loss of connectivity or an undersized cache structure.

If the rebuilding process is successful and connectivity resumes as before, all opens that were tied to the CF cache structure that failed are automatically reestablished. If the rebuild process fails, but another CF cache structure is defined in the cache set associated with the storage class and connectivity exists to that CF cache structure, then all opens are automatically reassigned to this alternate CF cache structure.

If the rebuild process fails and no alternate CF cache structure is available, any opens currently using the CF cache structure that failed are marked as broken. DFSMS fails the rebuilding process if the new CF cache structure is smaller than the failed structure, or if it does not have the same connectivity. In those cases where DFSMS cannot rebuild a CF cache structure, the next attempt to open a data set associated with the failed cache structure fails. You might need to redefine the cache structure or correct the connectivity problems.

Parent topic:[Recovering VSAM RLS processing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering the SMSVSAM server address space

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If the SMSVSAM server fails, it is automatically restarted. This is a complete reinitialization of the address space and the data space; all connections to prior instances of the SMSVSAM server are invalidated.

Requirement: For SMSVSAM to start, the RLSINIT parameter of the IGDSMSxx parmlib member must be set to YES.

The SMSVSAM server can be automatically restarted up to six times. If the limit is reached, the system issues message IGW418D. You respond to this message by indicating whether the automatic restart facility is to be re-enabled by the system or whether you want to manually enable it. If the automatic restart mechanism has been disabled, use the VARY SMS,SMSVSAM,ACTIVE command to restart SMSVSAM and re-enable the automatic restart facility.

Parent topic:[Recovering VSAM RLS processing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Recovering a shared control data set

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You should always run with at least two active and one spare SHCDSs. If a permanent I/O error occurs for an active SHCDS, or if an SHCDS becomes inaccessible from one or more systems, it is automatically replaced by one of the spare SHCDSs. When a system is forced to run with only one SHCDS, it issues a message requesting that you add another active SHCDS and at least one spare SHCDS. If any system does not have access to an SHCDS, all opens for VSAM RLS processing are prevented on that system until an SHCDS becomes available. The information in SHCDSs is continuously updated. Therefore, backup/restore procedures for SHCDSs are ineffective.

Use the following command formats for the VARY SMS command to add and delete SHCDSs:

To add a new, active SHCDS: `VARY SMS,SHCDS(SHCDS_name),NEW`

To add a new, spare SHCDS: `VARY SMS,SHCDS(SHCDS_name),NEWSPARE`

To delete either an active or a spare SHCDS: `VARY SMS,SHCDS(SHCDS_name),DELETE`

Parent topic:[Recovering VSAM RLS processing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Falling back from VSAM RLS processing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you fall back from VSAM RLS processing, the following occurs: the SMSVSAM address space shuts down permanently on every system in the Parallel Sysplex®; the SMSVSAM automatic restart capability is disabled; the lock structure IGWLOCK00 is de-allocated (but any secondary lock structures are not deleted), and all knowledge of SHCDSS and pending subsystem recovery is deleted. The following sections describe some rules and considerations to follow when falling back from VSAM RLS processing. They also outline the fallback procedure to follow, providing you are only falling back from RLS processing and that you are staying at current product levels. If you are also falling back to different CICS®, CICSVR, or DFSMS release levels, see [Understanding the product environment for VSAM RLS](#) for a list of rules and considerations to follow.

[Fallback rules and considerations](#)

[Fallback procedure](#)

Parent topic:[Administering VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Fallback rules and considerations

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Consider and plan for the following situations before you decide to fall back from VSAM RLS processing: There might be outstanding recovery for VSAM RLS data sets. The fallback procedure results in the loss of locks protecting back out.RLS indicators in the catalog must be reset. There might be applications which can only function with VSAM RLS, and which cannot return to a VSAM NSR, LSR or global shared resources (GSR) environment.

Parent topic:[Falling back from VSAM RLS processing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Fallback procedure

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Recommendation: Do not use this procedure for normal or abnormal disabling of the SMSVSAM server.

Follow these steps to fall back from VSAM RLS processing:

Ensure that there are no outstanding recovery requirements. Use the access method services SHCDS LISTRECOVERY or LISTSUBSYSDS command to list all current recovery requirements known to SMSVSAM. Any recovery must be completed prior to continuing with fallback processing, or data integrity is compromised. SMSVSAM is not aware of certain subsystem-related recovery, such as indoubt resolution.

See z/OS DFSMS Access Method Services Commands for a complete description of the access method services SHCDS command.

Activate an SMS configuration where all nonblank cache set specifications on storage classes are changed to blank cache set specifications. The SMS configuration should not include any cache sets defined in the base configuration. This ensures that the CF is only used for access to data sets which are already bound to a CF cache structure.

Ensure that you have no running applications which specify VSAM RLS processing, either specified in an ACB or using JCL.

Quiesce all CF cache structures. Use the VARY SMS command, as follows:

SMS, CFCACHE(CF_cache_structure_name), QUIESCE

VARY

where: **CF_cache_structure_name** is the name of the cache structure being quiesced. Enter this command for each CF cache structure in your configuration. Use the D SMS,CFCACHE command to verify that all CF cache structures are quiesced.

Reset RLS indicators in all applicable catalogs, using the SHCDS CFRESET command.

When you delete RLS information in the catalog, revert to CICS® FCT definitions, even if the CICS level remains unchanged. Be sure to complete all the preceding steps before continuing with the procedure to permanently shut down SMSVSAM and delete all knowledge of the lock structure and sharing control.

Change the value for the RLSINIT parameter in parmlib member IGDSMSxx to NO in *all* applicable parmlib members and activate the change.

Enter the following MVS command to disable the SMSVSAM server:

SMS, SMSVSAM, TERMINATESERVER

*ROUTE ALL V

Reply C to any outstanding IGW418D message. If FORCE SMSVSAM,ARM does not disable the SMSVSAM server, use FORCE SMSVSAM.

Enter the following command to complete the VSAM RLS fallback procedure:

SMS, SMSVSAM, FALLBACK

VARY

This command issues message IGW523 to request confirmation. You should first ensure that all SMSVSAM servers are disabled, then respond with:

FALLBACKSMSVSAMYES

Any other response cancels the command. The FALLBACK command forces all lock table connections, and deletes the lock structure and the sharing control group, IGWXSGIS. If any of these steps fail, or if another FALLBACK command is already in process, the command is rejected. Note: This command deletes the IGWLOCK00 lock structure, but does not delete any secondary lock structures. To delete secondary lock structures, use the SETXCF FORCE DELETE command. To completely fall back, you must also remove all secondary lock structures from the CFRM policy.

Fallback is complete when message IGW524I is issued to the console that issued the FALLBACK command.

See [Understanding the product environment for VSAM RLS](#) for a list of rules and considerations to follow if you are also falling back to different CICS, CICSVR, or DFSMS releases.

Parent topic:[Falling back from VSAM RLS processing](#)



[©Copyright IBM Corporation 1990, 2014](#)

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Terminating the SMSVSAM address space

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To terminate the SMSVSAM address space, enter the following command from the MVS console:

V SMS,SMSVSAM,TERMINATESERVER

Note: Use the VSMS,SMSVSAM,TERMINATESERVER command before you partition a system out of the XCF Sysplex. Failure to do so can result in unexpected abends in the SMSVSAM address space.

Parent topic:[Administering VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Quiescing or enabling a secondary lock structure

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To quiesce (stop) or enable a VSAM RLS secondary lock structure, enter the VARY SMS command, as follows:

`V SMS,CFLS(lockstructurename),Quiesce|Enable`

When a VSAM RLS secondary lock structure is enabled, SMSVSAM attempts to connect to the structure during open processing for a VSAM data set using a storage class containing a lock set that references that lock structure. Once the data set is open, the lock structure holds record locks for the sphere.

The quiesce option stops access to the VSAM RLS secondary lock structure. SMSVSAM does not allow any new spheres to access this lock structure. All spheres that are already connected are allowed access until they close. When all of the connected spheres close, the lock structure transitions from Quiescing to Quiesced and SMSVSAM disconnects from the lock structure.

Note: The VARY SMS command is not valid for the primary CF lock structure IGWLOCK00.

Parent topic:[Administering VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Deleting a VSAM RLS lock structure

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can delete secondary lock structures only with the operator command. All DFSMS lock structures are persistent structures that have persistent connections. When the secondary lock structure transitions to Quiesced state, SMSVSAM does not unallocate the lock structure in the coupling facility.

If you want to delete a secondary lock structure, the SMSVSAM address space previously connected must be terminated. You can use the following command:

SMS, SMSVSAM, FORCEDELETELOCKSTRUCTURE(lockstructurename)

To delete the primary lock structure, IGWLOCK00, all SMSVSAM address spaces must be terminated. You can use the following commands:

SMS, SMSVSAM, FORCEDELETELOCKSTRUCTURE(lockstructurename)

Or v SMS, SMSVSAM, FORCEDELETELOCKSTRUCTURE

For more information, see [Terminating the SMSVSAM address space](#).

Parent topic:[Administering VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Displaying information about a secondary lock structure

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To display information about one secondary lock structure or about all secondary lock structures, use the DISPLAY SMS command, as follows:

D SMS,CFLS(ALL | lockstructurename)

You can use the D SMS,SMSVSAM operator command to display the status of multiple SMSVSAM addresses spaces and secondary lock structures.

To display the status of the lock structures connected to the system from which you issued the command, use: D SMS,SMSVSAM

To display the status of all the lock structures in the sysplex, use: D SMS,SMSVSAM,ALL

Parent topic:[Administering VSAM record-level sharing](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Writing ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This topic documents intended Programming Interfaces that allow you to write programs to obtain the services of DFSMS. This topic is intended to help you to write ACS routines.

ACS routines determine SMS classes and storage groups for all new data set allocations and for data set allocations that occur from converting, copying, recalling, restoring or moving data sets. For objects, ACS routines determine: storage group, when storing them; or storage class and management class, when storing or changing them, or during class transitions. You write ACS routines in the ACS language, which is a high-level programming language. You can write your ACS routines, one for each type of SMS class and one for your storage groups. After writing the routines, you must translate them into an object form that SMS understands. A successful translation places the ACS object in a specified SCDS. After you activate the configuration contained in that SCDS, ACS routines govern storage management.

When you enter the ISMF Automatic Class Selection Application and select the EDIT option, you are linked to PDF Edit, where you can create or modify ACS routines. Leaving EDIT returns you to the ISMF Automatic Class Selection Application, where you can translate, validate, or test any ACS routine.

This topic contains four main sections. The first section describes the ACS language constants. The second section describes *read-write variables*. The third section describes *read-only variables*, which the ACS routines use for comparison operations. The fourth section describes the ACS language statements and illustrates the use of the statements in a storage group selection routine.

[Constants](#)

[Read-write variables](#)

[Read-only variables](#)

[Statements](#)



©Copyright IBM Corporation 1990, 2014

Constants

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use four types of constants in ACS routines: **Numeric**A numeric is a string containing up to ten characters, 0 - 9. You can use numerics in comparison operations involving the &NQUAL, &NVOL, and &RETPD read-only variables, which are discussed in [Read-only variables](#).**KB**, **MB**KB and MB are suffixes for numeric constants, such as 200 KB and 10 MB. One KB = 1,024 bytes while one MB = 1,048,576 bytes. Any comparison operation involving the &SIZE and &MAXSIZE read-only variables require that you use KB or MB. They are discussed in [Read-only variables](#). The maximum prefix value for KB is 2147483647. The maximum prefix value for MB is 2097151. Rule: When used for DASD storage, K and M normally mean 1000 and 1000000, not the values used here.

LiteralA literal is a character string, such as 'SYS1.PARMLIB', that is enclosed in single quotation marks. The maximum length of a literal is 255 characters. If you want a literal to contain a single quotation mark, such as PAYROLL'SDATA, then you must specify two single quotation marks: 'PAYROLL"SDATA'.**Mask**A mask is a character string, such as SYS1.*LIB, that is *not* enclosed in single quotation marks. You can use a mask to represent job names, volume serial numbers, or other system values that have a common string of characters, such as all volume serial numbers that begin with IMS™. You can also use a mask to represent data set, object or collection names that have a common string of characters. A mask must begin with an alphabetic character, numeric character (0 - 9), national character (\$, @, #), asterisk (*), or percent sign (%). The three characters “*”, “%”, and “.” have special significance in a mask. In addition, the characters “-” and “+” cannot be used in masks. These characters are reserved for use as continuation characters. The following sections describe the rules for using both the simpler masks and the slightly more involved data set masks. See [FILTLIST statement](#) for an explanation of the use of the mask characters.

[Simple mask rules](#)

[Simple mask examples](#)

[Data set mask rules](#)

[Data set mask examples](#)

Parent topic:[Writing ACS routines](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Simple mask rules

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following rules apply to the special characters in a simple mask containing a single level name: An asterisk, “*”, means that zero or more characters can be present in its place. Two or more adjacent asterisks are not allowed within a simple mask. A “%” represents exactly one nonblank character. “%%%” represents three character positions.

Parent topic:[Constants](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Simple mask examples

z/OS DFSMSdfp Storage Administration
SC23-6860-01

TSO*All names of any length beginning “TSO”***XYZ***All names of any length having three adjacent characters “XYZ”**IMS%%%**All six-character names beginning “IMS”*%**WK%%**All names where the second and third characters of the last five (or only five) are “WK”

Parent topic:[Constants](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data set mask rules

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following rules apply to the special characters in a data set mask: You can separate data set qualifiers with periods, “.”. Each qualifier has a maximum length of eight characters. The maximum length for the entire data set mask is 44 characters. A “%” represents a single character position. “%%%” represents three character positions. A single “*” by itself indicates that at least one qualifier is needed to occupy that position. A “*” within a qualifier means that zero or more characters can be present. A qualifier can be a single “*”. A “***” means that zero or more qualifiers can be present. A “**” cannot appear with any other characters within a qualifier. Three or more adjacent “*” are not allowed within a qualifier. The read-only variables which cannot be used for comparisons are: &ACSENVIR, &DSNTYPE, &DSORG, &DSTYPE, &LABEL, &RECORDG, and &XMODE.

Parent topic:[Constants](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data set mask examples

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SYS1.**All names where the first (or only) qualifier is “SYS1”**.**OUTLIST**All names where the last (or only) qualifier is “OUTLIST”*.*.**PAYROLL.*.SALARY.***All names with six qualifiers where the third qualifier is “PAYROLL” and the fifth qualifier is “SALARY”*.%**TEST.*.DATA**All names with four qualifiers where the second qualifier has six characters ending in “TEST” and the fourth qualifier is “DATA”**.***ABC*.****All names where some (or only) qualifier contains the characters “ABC” (or only “ABC”)**LABMGR.**.DATA**All names where the first qualifier is “LABMGR” and the last qualifier is “DATA”

Parent topic:[Constants](#)



©Copyright IBM Corporation 1990, 2014

Read-write variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You write ACS routines to assign values to read-write variables. You can also use read-write variables as values in comparison operations. These read-write variables *are* case sensitive. The ACS language has four read-write variables:

&DATACLAS&STORCLAS&MGMTCLAS&STORGRP

The &STORGRP read-write variable should only be used in the storage group routine. It is null on input to the routine unless VOL=REF is specified.

If the read-write variables are explicitly specified by the user, they have an initial value that might be overridden by the ACS routine. If the value is not overridden and the initial name is not defined in a currently active configuration, the allocation fails.

Requirement: You must specify a read-write variable on the PROC statement of the corresponding ACS routine. See [PROC statement](#) for details.

Each ACS routine can set only its corresponding read-write variable: The data class routine can set only &DATACLAS. The storage class routine can set only &STORCLAS. The management class routine can set only &MGMTCLAS. The storage group routine can set only &STORGRP. The ACS routines assign values to read-write variables using the SET command, which is explained in [SET statement](#).

In an ACS routine, you can assign an alphanumeric name enclosed in single quotation marks to the read-write variables. Also, you can assign a list of up to fifteen alphanumeric storage group names, each enclosed in single quotation marks, to the &STORGRP read-write variable. If more than one storage group name exists in the list, then each name must be enclosed in single quotation marks and separated by commas (for example, 'SG1', 'SG2').

[Table 1](#) indicates which read-write variables you can set, which ones you can use for comparisons, and which ones are invalid, for each of the ACS routines:

Table 1. Using Read-Write Variables in ACS Routines

[Using read-write variables with volume reference \(VOL=REF\)](#)

[Using read-write variables with data set stacking](#)

Parent topic: [Writing ACS routines](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using read-write variables with volume reference (VOL=REF)

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When volume reference (VOL=REF) is used, the storage group of the referenced data set is passed to the ACS routines in the &STORGRP read-write variable. However, keep in mind the following: The storage group name might not be available if the reference is to a data set on SMS-managed tape. This is because private tapes can be entered into a tape library with a blank storage group name. In this case, the AS routine should use the &LIBNAME read-only variable to determine the storage group for the referenced data set. In this case, the referenced and referencing data sets must reside in the same storage group. If the reference is to a new data set, there can be multiple candidate storage groups for the referenced data set and the actual storage group might not have been selected yet. In this case, only the first candidate storage group is passed as input to the ACS routines, and this might not be the storage group in which the referenced data set is eventually allocated.

Parent topic:[Read-write variables](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using read-write variables with data set stacking

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When a data set stacking inconsistency is detected, the ACS routines are re-invoked. When they are available, the following values are passed as input to the ACS routines:

- The storage group of the primary data set
- The storage class of the primary data set
- The management class selected by the previous invocation of the management class ACS routine for the stacked data set
- The data class selected by the previous invocation of the data class ACS routine for the stacked data set

Values might not be available for the following reasons:

- No storage class and storage group is available from the primary data set if it is directed to non-SMS-managed media.
- No management class is available for the stacked data set if it was initially assigned no storage class.
- If SMS was invoked by JES3, it is unable to access the work areas to obtain the storage class, management class, or data class.
- The storage group name might not be available if the primary data set is on SMS-managed tape. This is because private tapes can be entered into a tape library with a blank storage group name. In this case, the ACS routine should use the &LIBNAME read-only variable to determine the storage group for the stacked data set.
- In this case, the primary data set and the stacked data set must reside in the same storage group.
- If the primary data set is new, there might be multiple candidate storage groups for it and the actual storage group might not have been selected yet. In this case, only the first candidate storage group is passed as input to the ACS routines, and this might not be the storage group in which the primary data set is eventually allocated.

Parent topic:[Read-write variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Most ACS variables are read-only. Read-only variables contain data set and system information, and they reflect what is known at the time of the allocation request. You can use read-only variables in comparison operations, but you cannot change their values.

Attention: In the data class ACS routine, the &DSNTYPE, &DSORG, &MAXSIZE, &NVOL, &RECORC, and &SIZE variables all default to null if no corresponding value is specified in the JCL. Some values of the &DSNTYPE variable are set from values on the DD statement or dynamic allocation.

All of the read-only variables appear in [Table 1](#). The read-only variables *are* case sensitive. The following pages explain the uses of the read-only variables.

Table 1. Read-Only Variables

Name	Description
&ACCT_JOB	The accounting information from the JOB statement. (For a description of the indexing function for accounting information, see Special functions .) Type: Literal Max value: 142 characters
&ACCT_STEP	The accounting information from the EXEC statement. This information is refreshed for each step in the job. (For a description of the indexing function for accounting information, see Special functions .) Type: Literal Max value: 142 characters
&ACSENVIR	The environment in which the ACS routine was invoked, one of: ALLOC for new data set allocations (this is the default) ALLOCTST for a pre-allocation or allocation test environment. CHANGEOSREQ object change environment CONVERT for data set convert in place operations CTRANSOSMC object class transition environment RECALL for data set recall operations RECOVER for data set recover operations RENAME for data set alter rename operations RMMPOOL for DFSMSrmm requests for a storage group name RMMVRS for DFSMSrmm requests for a management class name STOREOSREQ object store environment other installation exit can set its own value before re-invoking ACSType: Literal Max value: 8 characters
&ACSENVIR2	The sub-environment in which the ACS routine was invoked. The valid values are the values for &ACSENVIR as well as FLASHCPY and blank. Type: Literal Max value: 8 characters
&ALLVOL	The volume serial numbers specified for data set allocations when &ACSENVIR is not recall or recover. When the environment is recall or recover, &ALLVOL is either the volume serial number on which the data set resided at the time it was migrated or backed up, or the volume serial number specified as the target volume of the recall or recover. Exception: &ALLVOL is not available to the storage group ACS routine when the environment is recall or recover and when VOLCOUNT(ANY) is specified. See Special functions for usage information. Type: Literal Max value: 6 characters The &ALLVOL ACS read-only variable contains the following values when you use VOL=REF: 'REF=SD' (the volume reference is to an SMS-managed DASD or VIO data set)'REF=ST' (the volume reference is to an SMS-managed tape data set)'REF=NS' (the volume reference is to a non-SMS-managed data set)
&ANYVOL	The volume serial numbers explicitly specified for the volumes if &ACSENVIR is not recall or recover. When the environment is recall or recover, &ANYVOL is either the serial number on which the data set resided at the time it was migrated or backed up, or the volume serial number specified as the target volume of the recall or recover. Attention: &ANYVOL is not available to the storage group ACS routine when the environment is recall or recover and when

VOLCOUNT(ANY) is specified. See [Special functions](#) for usage information.

Type: Literal

Max value: 6 characters

The &ANYVOL ACS read-only variable contains the following values when you use VOL=REF: 'REF=SD' (the volume is reference to an SMS-managed DASD data set)'REF=ST' (the volume is reference to an SMS-managed tape data set)'REF=NS' (the volume is reference to a Non-SMS-managed data set)

&APPLICThe name of the application associated with the resource owner of the data set (which is set only if RACF® is installed and ACSDEFAULTS is YES in IGDSMSxx). Type: Literal

Max value: 8 characters

&BLKSIZEThe numeric value for the block size specified on the DD statement, dynamic allocation, or TSO ALLOCATE, ranging from 0 - 2147483647 (KB). Type: Numeric

Max value: 2147483647

&DDDDNAME in the DD statement of the data set. Type: Literal

Max value: 8 characters

&DEF_DATACLASThe data class name associated with the resource owner of the data set (set only if RACF is installed and ACSDEFAULTS is YES in IGDSMSxx).

Type: Literal

Max value: 8 characters

&DEF_MGMTCLASThe management class name associated with the resource owner of the data set (set only if RACF is installed and ACSDEFAULTS is YES in IGDSMSxx). Type: Literal

Max value: 8 characters

&DEF_STORCLASThe storage class name associated with the resource owner of the data set (set only if RACF is installed and ACSDEFAULTS is YES in IGDSMSxx). Type: Literal

Max value: 8 characters

&DSNThe name of the data set or collection for which ACS processing is taking place. For VSAM data sets, only the cluster name is passed to the ACS routine; the component names are not. If the data set has an absolute or relative generation number, it is stripped from &DSN. The generation number is the low-level qualifier of the data set name. For the data set naming rules, see z/OS MVS JCL Reference.

Type: Literal

Max value: 44 characters

&DSNTYPEThe data set name type, one of: **BASIC**The data set is neither extended format nor large format.**EXC**Extended format data set is preferred. The data set allocation is attempted in nonextended format if the necessary system resources for extended are not available.**EXR**Extended format data set is required. The data set allocation fails if unable to allocate in extended

format.**HFS**Hierarchical file system data set**LARGE**Large format data

LIBRARYPDSE, partitioned data set extended**PDS**Partitioned data set**null**No value specifiedType: Literal

Max value: 7 characters

Note: Some values of the &DSNTYPE variable (BASIC, EXC, EXR, and LARGE) are set from values on the DD statement or dynamic allocation, as well as from the data class or the LIKE parameter.

&DSORGThe data set organization, one of: **PS**Physical sequential**PO**Partitioned**VS**VSAM organization**DA**DAM organization**null**No value specifiedType: Literal

Max value: 2 characters

Restriction: When DIRECTORY blocks are specified on the SPACE parameter in

the JCL, &DSORG is set to PO.

&DSOWNERThe name of the user or group that owns the data set (set only if RACF is installed). Type: Literal

Max value: 8 characters

&DSTYPEThe data set type, one of: **GDS**One generation data set of a generation data group, or any data set allocated with a relative generation number (such as A.B.C(+1)) or an absolute generation number (such as

A.B.C.G0000V00). **PERM**Standard permanent data sets **TEMP**Temporary data sets **null**None of the above. Type: Literal

Max value: 8 characters

&EATTREATTR (extended attributes) value, one of: **OPT**Extended attributes are optional **NO**The data set cannot have extended attributes **blank**Not specified. This is the default. Type: Literal

Max value: 4 characters

&EXPDTThe expiration date in the form of YYYYDDD where YYYY is a year from 1900 - 2155 and DDD is a day in a year from 1 - 366. Exception: Expiration dates of 99365 and 99366 are considered "NEVER-SCRATCH" dates.

Type: Literal

Max value: 7 characters

&FILENUMThe value of the FILENUM ACS read-only variable. This variable corresponds to the data set sequence number on the JCL LABEL parameter. The default is 1. This field is optional. Type: Numeric

Max value: 5 characters

&GROUPThe RACF-defined default group associated with the user, or the group specified in the GROUP keyword on the JCL JOB statement. If the environment is recall or recover, DFSMSHsm sets &GROUP for both authorized and non-authorized users if it is available. Therefore, do not rely on a null value for &GROUP to determine if a user is a DFSMSHsm Authorized User or not. When DFSMSHsm invokes the ACS routines, &GROUP is the group associated with &USER. Type: Literal

Max value: 8 characters

&HLQThe high-level (first) qualifier of the data set or collection name. Type: Literal

Max value: 8 characters

&JOBThe job name, the started task name, or the TSO/E userid from the JOB statement, depending on the execution mode (&XMODE). (See [Determining Distributed FileManager/MVS data set creation requests](#) for Distributed FileManager/MVS usage information.) Type: Literal

Max value: 8 characters

&LABELThe value of the LABEL ACS read-only variable. This variable corresponds to the label field of the JCL LABEL parameter. Allowable values are NL, AL, SL, NSL, SUL, AUL, BLP, LTM or blank. The default is IBM® Standard Label. This field is optional. Type: Literal

Max value: 3 characters

&LIBNAMEThe name for the LIBNAME ACS read-only variable, can contain a 1 to 8 character tape library name. This field is optional. Type: Literal

Max value: 8 characters

&LLQThe low-level (last) qualifier of the data set or collection name. Type: Literal

Max value: 8 characters

&MAXSIZEThe maximum size (in KB or MB) of a new data set. For non-VSAM data sets, the value is &SIZE plus 15 extents. For VSAM data sets, the value is primary + (123 * secondary * volume count) if extent constraint removal is set to No. If extent constraint removal is set to Yes, the value is &SIZE plus 7257 extents. See [Using read-only variables](#) for more information about the values of &MAXSIZE and &SIZE for VSAM data sets. Also see [Constraints when using read-only variables](#).

Type: Suffix numeric

Max value: 2147483647 for KB, 2097151 for MB

&MEMHLQThe high-level (first) qualifier of the object name. Type: Literal

Max value: 8 characters

&MEMLLQThe low-level (last) qualifier of the object name. Type: Literal

Max value: 8 characters

&MEMNThe name of an object. Type: Literal

Max value: 44 characters

&MEMNQUALThe number of qualifiers in the object name. Type: Numeric

Max value: 22

&MSPDESTThe destination, specified in data set name format, for a tape management system-driven tape allocation. This value is specified through the AMS pre-ACS installation exit. The data set name format lets you specify a sequence of destinations to be identified, where each qualifier is a specific destination. For example, a data set vaulted first at location OUTD and then sent to OLTS could have an MSPDEST of 'OUTD.OLTS'. The actual values depend on the support provided by your tape management system. Type: Alphanumeric

Max value: 44 characters

&MSPARMAdditional information related to a tape management system-driven tape allocation. This is a variable length field that can be indexed. The value is specified through an external exit. For a description of the indexing function for virtual tape system parameters, see [Virtual tape system parameter](#). Type: Alphanumeric

Max value: 142 characters

&MSPOLICYThe name of a management policy associated with tape data for a tape management system-driven allocation. You can use the DFSMSrmm EDGUX100 installation exit to set MSPOLICY to a VRS management value name. You can also set the value of this variable using the SMS pre-ACS installation exit or allow your tape management system to set it using the pre-ACS installation exit.

Type: Alphanumeric

Max value: 8 characters

&MSPOOLA tape pool name associated with the data set being allocated. In a system-managed tape environment with scratch pool support, you can use this variable to specify a default storage group, where the tape storage group is equivalent to the tape pool specified in the variable. If you use the DFSMSrmm EDGUX100 installation exit, you can set this variable to the pool name or prefix determined by the DFSMSrmm scratch pool processing. This variable can also be set through the pre-ACS installation exit. Type: Alphanumeric

Max value: 8 characters

&NQUALThe number of qualifiers in the data set or collection name. Type:

Numeric

Max value: 22

&NVOLThe maximum of the volume count, UNIT count, and number of explicit VOL=SER specifications. Type: Numeric

Max value: 2147483647

&PGMThe name of the program the system is running. (See [Determining Distributed FileManager/MVS data set creation requests](#) for Distributed FileManager/MVS usage information.) Type: Literal

Max value: 8 characters

&RECORDThe data set record organization, one of: **K**SVSAM key sequenced (KSDS)**E**VSAM entry sequenced (ESDS)**R**RVSAM relative record (RRDS)**L**SVSAM linear**null**No value specifiedType: Literal

Max value: 2 characters

&RETPDThe retention period of NNNN days where NNNN is from 0 - 9999 (JCL maximum). The &RETPD value is calculated from the specified &EXPDT value. If 99365 or 99366, the two "NEVER EXPIRED" dates, is specified after 01 January 2000, the calculated value for &RETPD is set to 0. &EXPDT, however, contains the true value of 1999365 or 1999366. Type: Numeric

Max value: 2147483647

&SECLABLSpecifies the default security label in the RACF profile of the user or data set if the SECLABEL class is active. Otherwise, the read-only variable will contain a null value. Type: Literal. The first character must be alphabetic, \$, #, or @.

Max value: 1 - 8 alphanumeric or national (\$, #, @) characters.

&SECOND_QTYSpecifies a secondary allocation quantity from JCL or AMS control cards. It has meaning only when the &SPACE_TYPE variable is defined. In conjunction with &SPACE_TYPE, &SECOND_QTY allows the DATACLAS ACS routine to make the appropriate DATACLAS assignment. You must modify your existing ACS routines if you wish to use these two variables. **&SIZE**The primary amount of space (in KB or MB) requested for a new data set or the amount of space actually used in an existing data set on a DASD volume (see [Constants](#)). See [Using read-only variables](#) for more information about the values of &MAXSIZE and &SIZE for VSAM data sets. Type: Suffix numeric

Max value: 2147483647 for KB; 2097151 for MB

&SPACE_TYPESpecifies the allocation unit from JCL or AMS control cards to be used in conjunction with the secondary space allocation amount. In conjunction with the &SECOND_QTY variable, &SPACE_TYPE allows the DATACLAS ACS routine to make the appropriate DATACLAS assignment. You must modify your existing ACS routines if you wish to use these two variables. Valid values: TRK, CYL, K, M, U, BLK, and blank

&SYSNAMESpecifies the system name of the system on which the ACS routine is executing. This field is optional. See [Using read-only variables](#) for usage information. Type: Literal

Max value: 8 characters

&SYSPLEXSpecifies the Parallel Sysplex® name of the system on which the ACS routine is executing. This field is optional. See [Using read-only variables](#) for usage information. Type: Literal

Max value: 8 characters

&UNITIBM-supplied or installation defined generic name for a device type (for example, 3380, SYSDA). For additional possible settings for the &UNIT variable, see [ACS routine environments](#). Type: Literal

Max value: 8 characters

A slash (/) preceding a four digit number represents a unit address (for example, /3380).

When you allocate a tape data set with DISP=MOD and no unit information is specified in the JCL, this variable is blank and SMS might attempt to manage the tape data set as a DASD-resident data set.

&USERThe user ID of the person allocating the data set. When DFSMShsm invokes the ACS routines, &USER is either the requestor of the recall or recover, or the user ID of the DFSMShsm address space. If the environment is recall or recover, DFSMShsm sets &USER for both authorized and non-authorized users if it is available. Therefore, do not rely on a null value for &USER to determine if a user is a DFSMShsm authorized user or not. (Refer to [Determining Distributed FileManager/MVS data set creation requests](#) for Distributed FileManager/MVS usage information.) Type: Literal

Max value: 8 characters

&XMODEThe execution mode in which the data set is being allocated, one of:

BATCHBatch execution mode **TSO** TSO execution mode **TASK**A started address
spaceType: Literal
Max value: 8 characters

[Using read-only variables](#)

[Constraints when using read-only variables](#)

[Comparison operators](#)

[Comparison rules](#)

[Boolean expressions](#)

[Special functions](#)

[Determining Distributed FileManager/MVS data set creation requests](#)

Parent topic:[Writing ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following sections describe various considerations for using read-only variables:

[Initializing read-only variables](#)

[Using default read-only variables](#)

[Ensuring correct values for &SIZE and &MAXSIZE](#)

[Using OAM read-only variables](#)

[Using &DSN for a partitioned data set](#)

[When the value of &USER is null](#)

[Using the &SYSNAME and &SYSPLEX read-only variables](#)

[Read-only variables on a data set rename](#)

Parent topic:[Read-only variables](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Initializing read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SMS derives the values of read-only variables before it invokes the ACS routines. The values are based on what is known at the time of the allocation request (for example, if a unit name has been specified on the allocation request, then the &UNIT variable contains the specification).

Exception: &DSNTYPE is not initialized with the DSNTYPE default specified in the IGDSMSxx member in SYS1.PARMLIB.

Parent topic:[Using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using default read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Read-only variables are defaulted from data class (if assigned) before storage class, management class, and storage group ACS routines are invoked.

Parent topic:[Using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

Ensuring correct values for &SIZE and &MAXSIZE

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you allocate a data set using the TSO/E ALLOCATE command and you do not explicitly specify space requirements, then &SIZE and &MAXSIZE do not contain the correct values. Instead they both contain a value of zero. If your ACS routines rely on the values of &SIZE or &MAXSIZE in this situation, the data set might be assigned to the wrong class or group.

For a VSAM data set definition, the &SIZE and &MAXSIZE read-only variables reflect the space value specified in the CLUSTER component. If one is not specified in the CLUSTER component, then the space value specified in the DATA component is used. If a space value also is specified for the INDEX component *and* it is of the same type of space unit; for example, both are in tracks, cylinders, KB or MB, it is added to what was specified for the DATA component. If the INDEX component space unit is not of the same type as specified for the DATA component, it is ignored and not added to &SIZE or &MAXSIZE. For DFSORT work data sets, &SIZE and &MAXSIZE are zero. In the RECALL environment, &SIZE and &MAXSIZE are zero for empty partitioned data set and PS data sets because the size of the existing data set is not known at the time that DFSMS runs its ACS routines.

As input to data class ACS routine, &SIZE and &MAXSIZE are calculated from space information from JCL or IDCAMS, as follows: For non-VSAM data sets: &SIZE = P + Directory (if any).&MAXSIZE = P + (S*15) + Directory (if any). For VSAM data sets: &SIZE = P.&MAXSIZE = P + (S*254).

After data class is derived by the data class ACS routine, &SIZE and &MAXSIZE are recalculated depending on whether Extent Constraint Removal is specified or not in the selected data class before calling subsequent data class routines, as follows: For non-VSAM data sets, if no space is specified on JCL, then the space values defined in the selected data class are used to recalculate &SIZE and &MAXSIZE as follows: &SIZE = P + Directory (if any).&MAXSIZE = P + (S*15) + Directory (if any). For VSAM data sets, regardless whether space is specified or not, &MAXSIZE is recalculated depending on the specification of Extent Constraint Removal attribute as follows: If Extent Constraint Removal = NO, then &MAXSIZE = P + (S*122)*volcnt. If Extent Constraint Removal = YES, and the ADD' 1 Volume Amount in Data Class = P, then &MAXSIZE = ((P+S*122))*volcnt. If Extent Constraint Removal = YES, and the ADD' 1 Volume Amount in Data Class = S, then &MAXSIZE = ((P+S*122)) + ((S*123)*(volcnt-1)).

Parent topic:[Using read-only variables](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using OAM read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

&MEMN, &MEMHLQ, &MEMLLQ and &MEMNQUAL are used to name an object in a collection.

Restriction: These read-only variables are valid only when &ACSENVIR is equal to one of the three valid OAM environments (STORE, CTRANS, CHANGE), otherwise, the passed value is nullified by ACS before invoking the ACS routines.

Parent topic:[Using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using &DSN for a partitioned data set

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For a partitioned data set, &DSN consists only of the data set name. A member name, if specified, is not be part of the value of the &DSN or &MEMN variables. There is no class selection capability based on member name.

Parent topic:[Using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

When the value of &USER is null

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Not all read-only variable values are significant during the actual operation of any ACS routine. For example, the value of &USER could be null in the storage group selection routine if no user ID had been specified on the JOB statement or determined from the environment.

The user ID (and group ID) is only available if specified using JCL or if RACF® (or any other security product) is active. For example, on a RESTORE, if a user ID is specified, the ACERO passed by DFSMSdss does not contain the user ID, and because RACF is not active, DFSMS does not interrogate the ACEE to fill it in.

Parent topic:[Using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using the &SYSNAME and &SYSPLEX read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Due to connectivity constraints on old DASD and tape, it might be necessary to know the system and Parallel Sysplex® where an ACS routine is being executed in order to direct the allocation to a storage groups that is accessible from the current system. To support this function, SMS ACS processing uses the &SYSNAME and &SYSPLEX read-only variables.

Do not use the &SYSPLEX variable in ACS routines for JES3 systems as Parallel Sysplex name support is not supported in a JES3 environment. Also, ACS routines for JES3 systems should not rely on the &SYSNAME variable, as the system on which the ACS routines are run (during converter/interpreter) might have nothing to do with the system on which the job is eventually executed.

Parent topic:[Using read-only variables](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Read-only variables on a data set rename

z/OS DFSMSdfp Storage Administration
SC23-6860-01

On a data set rename using &ACSENVIR=RENAME, not all read-only variables are passed to the ACS routines.

The following read-only variables *are* passed to the ACS routines:

&ACSENVIR&APPLIC (set by ACS routines)&DEF_DATACLAS (set to null by ACS routines)&DEF_MGMTCLAS (set to null by ACS routines)&DEF_STORCLAS (set to null by ACS routines)&DSOWNER&DSN&DSORG&DSTYPE&EXPDT&GROUP&SYSNAME (set by ACS routines)&SYSPLEX (set by ACS routines)&USERThe following read/write variables are also passed to the ACS routines:

&DATACLAS&MGMTCLAS&STORCLASDuring rename processing, the STORCLAS ACS routine is not called or redriven. However, redriving the MGMTCLAS ACS routine on a data set rename invokes the management class ACS routine when an SMS-managed cluster, generation data set (GDS) or non-VSAM data set is renamed. Catalog management invokes MGMTCLAS ACS routines during rename processing. Refer to [Access Method Services](#) for more information.

Parent topic:[Using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Constraints when using read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following sections summarize some constraints when using read-only variables.

[Read-only variables not allowed in storage group routine](#)

[Read-only variables in different environments](#)

[Read-only variables not available when LIKE is used](#)

[OAM read-only variables](#)

Parent topic:[Read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Read-only variables not allowed in storage group routine

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following read-only variables are **not** allowed in the storage group selection routine: &ACCT_JOB&ACCT_STEP&DD&JOB&MSVGP&PGM&XMODE

Parent topic:[Constraints when using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Read-only variables in different environments

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following read-only variables are *not* passed to any ACS routine unless the environment is specified as &ACSENVIR=ALLOC:
&ACCT_JOB&ACCT_STEP&DD&GROUP&JOB&MSGVP&PGM&USER&XMODE
Exceptions: If the environment is a non-IBM supported environment, the following read-only variables are passed regardless of what is specified in &ACSENVIR:
&JOB&DD&PGM If the environment specified in &ACSENVIR is RECALL, RENAME, or RECOVER, the following read-only variables are passed:
&GROUP&USER

Parent topic:[Constraints when using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Read-only variables not available when LIKE is used

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you use the LIKE parameter on a JCL DD statement or the ALLOCATE command, the following read-only variable values are not available to the ACS routines: &DSNTYPE&DSORG&MAXSIZE&RECORG&SIZE Thus, a data set allocated like a second data set might go into a different storage group than the second data set.

Parent topic:[Constraints when using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

OAM read-only variables

z/OS DFSMSdfp Storage Administration
SC23-6860-01

For restrictions, see [Using OAM read-only variables](#).

Parent topic:[Constraints when using read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Comparison operators

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Comparison operators allow you to determine the relationship between two values. The following comparison operators are allowed: **This:Means this:GT or >Greater thanLT or <Less thanNG or ¬>Not greater thanNL or ¬<Not less thanEQ or =EqualNE or ¬=Not equalGE or >=Greater than or equalLE or <=Less than or equal**

Alphabetic characters are sorted before digits (A-Z come before 0-9). The following comparison is true for all high-level qualifiers alphanumerically greater than "M": IF &HLQ > 'M' THEN . . .

For FILTLIST or mask comparisons, only EQ and NE are valid. See [FILTLIST statement](#) for details.

Parent topic:[Read-only variables](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Comparison rules

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following rules apply to comparisons: For a comparison to be valid, one operand must either be a read-only variable or a read-write variable and the other operand must be a constant (any of the four types), a read-only variable, or a FILTLIST name. Numerics are right justified. Literals are left justified and padded with blanks. Type checking is done to ensure that numeric read-only variables are not being compared to characters (literals) and that character (literal) read-only variables are not being compared to numbers. &NQUAL, &NVOL, &SIZE, &MAXSIZE &MEMNQUAL and &RETPD are the only numeric read-only variables. Limited length checking of read-only variables with their maximum length values is performed to ensure that the maximum lengths are not exceeded. For example, the literal to which &DSN is being compared must be no longer than 44 characters. See [Read-only variables](#) for maximum lengths.

Parent topic:[Read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Boolean expressions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the following Boolean operators in any ACS routine: **This:Means this:AND or &&;AndOR or |Or** Expressions in parentheses are processed first. In the following example, the set of OR expressions are processed first, so that WHEN

```
((CONDITION 1) OR  
  (CONDITION 2) OR  
  (CONDITION 3) AND  
  (CONDITION 4)) SET ...
```

is processed as follows: WHEN

```
(CONDITION 1 AND CONDITION 4) OR  
(CONDITION 2 AND CONDITION 4) OR  
(CONDITION 3 AND CONDITION 4) SET ...
```

If you want the AND expression be processed before OR, the AND expression must be included in a set of parentheses. In the following example, AND is processed first so that WHEN

```
((CONDITION 1) OR  
  (CONDITION 2) OR  
  ((CONDITION 3) AND  
  (CONDITION 4))) SET...
```

is processed as follows: WHEN

```
((CONDITION 1) OR  
  (CONDITION 2) OR  
  (CONDITION 3 AND CONDITION 4)) SET...
```

Parent topic:[Read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Special functions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The ACS language provides the following three indexing functions, which allow you to make class selections based on specific details about each data set.

Table 1. Indexing Functions for Read-Only Variables

Data set qualifier indexing function

&ALLVOL and &ANYVOL functions

Accounting information indexing function

Virtual tape system parameter

Parent topic:[Read-only variables](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data set qualifier indexing function

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The data set qualifier indexing function lets you index the &DSN and &MEMN variables (for accessing particular qualifiers): **&DSN(1)**first qualifier of the data set name**&DSN(3)**third qualifier of the data set name**&DSN(&NQUAL)**last qualifier of the data set name**&MEMN(1)**first qualifier of the object name**&MEMN(2)**second qualifier of the object name**&MEMN(&MEMNQUAL)**last qualifier of the object name The only accepted values for indexes are numbers (1 through 22) and the read-only variables, &NQUAL and &MEMNQUAL.

Parent topic:[Special functions](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

&ALLVOL and &ANYVOL functions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

These functions let you compare the volume serial numbers explicitly specified on input with a comparison variable (for example, a FILTLIST variable). The use of &ALLVOL in a comparison expression returns a true value if ALL of the input volsers satisfy the desired condition. The use of &ANYVOL returns a true value if ANY of the input volsers satisfies the desired condition. For example, let IMS101, IMS102, and TSO191 be the input volsers to the routine in [Figure 1](#):

Figure 1. Example of Constraints when Using the &ALLVOL and &ANYVOL Read-Only Variables

```
PROC STORCLAS
  IF &ALLVOL = IMS* THEN
    (code left out)
  SELECT
    WHEN (&ANYVOL = TSO*)
    (code left out)
  END
```

The IF statement is false because not *a//volumes* match. While IMS101 and IMS102 satisfy the IMS* mask, TSO191 does not.

The WHEN statement is true, because *any*(at least one) volume does match. While IMS101 and IMS102 fail to satisfy the TSO mask, TSO191 does satisfy the mask. For a detailed explanation of the IF statement, refer to [IF statement](#).

Parent topic:[Special functions](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Accounting information indexing function

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Allows you to reference specific fields in the JOB or STEP account information. A field is defined as a unit of data which is separated by commas in the account information. The use of &ACCT_JOB(n) and &ACCT_STEP(n), where n is the field number, indicates that indexing is requested. If either &ACCT_JOB or &ACCT_STEP is specified without an index, then the default action is to access the first field of the accounting information (for example, &ACCT_JOB(1) or &ACCT_STEP(1) result by default)

information . &ACCT_JOB(1) first field of the JOB accounting
&ACCT_JOB first field of the JOB accounting information (default)
&ACCT_JOB(3) third field of the JOB accounting information
&ACCT_STEP first field of the STEP accounting information (default)
&ACCT_STEP(5) fifth field of the STEP accounting information

The only accepted values for indexes are numbers from 1 - 71.

Parent topic:[Special functions](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Virtual tape system parameter

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Allows you to access the subfields in the virtual tape system parameter set by the Pre-ACS Installation Exit. The parameter is defined as a 256-byte character string including one or more sets of 2-byte length field followed by the value. The use of &MSPARM(n), where n is the field number, indicates that indexing is requested. If &MSPARM is specified without indexing, then the default is to access the first field of the parameter (for example, &MSPARM1) result by default).

field of the parameter (default)
&MSPARM(1) first field of the parameter
&MSPARM(4) fourth field of the parameter

&MSPARM first

The only accepted values for indexes are numbers from 1 - 71.

Parent topic:[Special functions](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Determining Distributed FileManager/MVS data set creation requests

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The ACS routines are used to determine the SMS classes for data sets created by Distributed FileManager/MVS. You can use the &JOB, &PGM, and &USER read-only variables to distinguish Distributed FileManager/MVS data set creation requests. The following examples show how each of these variables can be used.

&JOB The value of the &JOB variable is the job specified in the Advanced Program to Program Communications (APPC) transaction program established by the installation systems programmer. [Figure 1](#) shows how each &JOB can be used to determine the SMS storage class for a data set being created using Distributed FileManager/MVS: [Figure 1. Example Showing &JOB Read-Only Variable](#)

PROC &STORCLAS

```
IF &JOB = 'GDEDFFM' AND &STORCLAS = '' THEN
  SET &STORCLAS = 'DFMCLASS'
ELSE
  .
  .
  .
```

&PGM The value of the &PGM variable is specified in the transaction program profile and is always GDEISASB. [Figure 2](#) shows how &PGM can be used to determine the SMS storage class for a data set being created using Distributed FileManager/MVS: [Figure 2. Example Showing &PGM Read-Only Variable](#)

PROC &STORCLAS

```
IF &PGM = 'GDEISASB' AND &STORCLAS = '' THEN
  SET &STORCLAS = 'DFMCLASS'
ELSE
  .
  .
  .
```

&USER You can designate a specific user ID under which all Distributed FileManager/MVS transaction programs are to run. You can then check the user ID to determine if the data set is being created using Distributed FileManager/MVS. The user ID is specified in the job statement in the Distributed FileManager/MVS transaction program profile. [Figure 3](#) shows how &USER can be used to determine the SMS storage class for a data set being created using Distributed FileManager/MVS:

[Figure 3. Example Showing &USER Read-Only Variable](#)

```
IF &USER = 'user_ID' AND &STORCLAS = '' THEN
  SET &STORCLAS = 'DFMCLASS'
ELSE
  .
  .
  .
```

Parent topic:[Read-only variables](#)



©Copyright IBM Corporation 1990, 2014

Statements

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This section describes the function and syntax of the ACS language statements that you can use when writing ACS routines.

The continuation characters “+” and “-” allow you to extend literal constants to the next line. To ignore the leading blanks on the following line, use “+”. If you want to include the leading blanks on the next line as part of a literal, then use a “-”. You cannot continue masks, numbers, KB or MB numerics, or keywords.

The maximum number of nesting levels for any combination of ACS statement types is thirty-two. (For example, a nested IF statement is one that appears within an IF statement.)

Comments begin with a slash-asterisk pair, “/*”, and end with an asterisk-slash pair, “*/”. You can place comments anywhere within an ACS routine where a delimiter might appear. Comments cannot be nested; each comment ends at the first occurrence of an asterisk-slash pair, “*/”. Asterisks within the comment statement are treated as a special character. A maximum of 500 asterisks can be included in a single comment statement (multiple comment lines not ended with “*/”). This could affect the number of lines allowed in comment continuation.

The statement types are defined as follows:

PROC	Start of an ACS routine
FILTLIST	Definition of filter criteria
SET	Assigns a value to a read-write variable
DO	Start of statement group
IF	Provides conditional statement execution
SELECT	Defines a set of conditional execution statements
EXIT	Causes immediate termination of the ACS routine and can be used to force allocation failures
WRITE	Sends a message to the end user
END	End of statement group (DO or SELECT) or ACS routine (PROC).

[PROC statement](#)

[FILTLIST statement](#)

[SET statement](#)

[DO statement](#)

[IF statement](#)

[SELECT statement](#)

[EXIT statement](#)

[WRITE statement](#)

[END statement](#)

[Sample ACS routine](#)

Parent topic:[Writing ACS routines](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

PROC statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

PROC is the first statement of each ACS routine. It identifies the ACS routine and which read-write variable the routine sets. You can precede the PROC statement with blank lines or comments, but not with other statements. For TSO CLIST coexistence, you can place a blank and then a number, such as 0 or 1, after the PROC statement. The number does not affect ACS language processing. To identify an ACS routine and the value it is to determine, you must specify a read-write variable at the end of the PROC statement. You must also place an END statement at the end of each ACS routine.

PROC <n>*read-write variablen* is optional and can contain any numeric value. *read-write variable* is a mandatory value that can be DATACLAS, STORCLAS, MGMTCLAS, or STORGRP. You can optionally precede the variable with an ampersand, &.

```
PROC 1 DATACLAS
PROC 0 &STORCLAS
PROC &MGMTCLAS
PROC STORGRP
```

END

Parent topic:[Statements](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

FILTLIST statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The FILTLIST statement is a definition list that you can use when testing variables in an ACS routine. You define the information that you want to include and exclude in the list using the INCLUDE and EXCLUDE keywords. Then you can compare read-only variables to items in the list using IF-THEN and SELECT-WHEN statements, without having to write elaborate AND and OR combinations.

FILTLIST is a definition statement that simplifies comparison operations. It is not an execution statement, and it does not change the value of any variables.

Because a FILTLIST can contain only literal values, you can only compare it to literal read-only variables. This excludes the numerically valued &NQUAL, &NVOL, &SIZE, &MAXSIZE, &MEMNQUAL, and &RETPD read-only variables from FILTLIST comparisons.

You must define a FILTLIST before you reference it in the body of an ACS routine.

FILTLIST*name* [<INCLUDE(/*list*>) [<EXCLUDE(/*list*>)] *name* is mandatory and can be up to 31 alphanumeric characters in length. You can also use an underscore, _, but it cannot be the first character. In the FILTLIST, you can optionally precede the name with an ampersand, & When referring to the FILTLIST in the body of the routine, you must always precede the FILTLIST name with an ampersand. You must specify INCLUDE, EXCLUDE, or both in the FILTLIST statement. If a list item satisfies both the INCLUDE and EXCLUDE criteria, EXCLUDE takes precedence and prevents the item from being included in the list. /*list* can contain literals, simple masks, and data set masks. You can specify up to 255 entries in the INCLUDE or EXCLUDE lists. [Figure 1](#) shows an example of coding a FILTLIST statement:

Figure 1. Example of a FILTLIST Statement_{PROC STORCLAS}

```
FILTLIST VLIST2 INCLUDE(DBX*, TSO*) EXCLUDE('DBX191', 'TSO256')
IF &ALLVOL = &VLIST2 THEN
  (some action)
END
```

Figure 2. Using the INCLUDE and EXCLUDE

In the environment shown in [Figure 2](#), the value of the IF statement is *true* for any of the following volume serials: TSO191TSO002DBX256DBXRES

The value of the IF statement is *false* for the following volume serials, because they match the EXCLUDE filter criteria: TSO256DBX191

Parent topic:[Statements](#)



©Copyright IBM Corporation 1990, 2014

SET statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The SET statement assigns values to the read-write variables. The values can be the names of constructs or RACF® derived defaults (&DEF_DATACLAS, &DEF_STORCLAS, or &DEF_MGMTCLAS). You can assign one name to &DATACLAS, one name to &STORCLAS, and one name to &MGMTCLAS, but you can assign a list of up to 15 names to &STORGRP.

The names can be from one to eight characters long, and they must be enclosed in single quotation marks. The individual names belonging to a &STORGRP must be enclosed in single quotation marks and separated by commas. A name must begin with either an alphabetic or national (\$, @, #) character, and the remaining characters can be alphabetic, numeric, or national.

In the body of an ACS routine, you can set only the value of the read-write variable identified in the PROC statement. You cannot set a read-write variable equal to another read-write variable, a FILTLIST variable, or a read-only variable (except for the RACF-derived values). For example, the following is not valid:

```
SET &STORCLAS = &PGM
```

You can assign a null value to any of the read-write variables except for &STORGRP. You can assign a null value by specifying two single quotation marks with nothing between them ("").

SET *read-write variable* = *value*

where: *read-write variable* is a mandatory value that can be &DATACLAS, &STORCLAS, &MGMTCLAS, or &STORGRP. You can specify EQ in place of the equals sign, =. *value* can be one name, a null value, a RACF read-only variable name, or a list of names for the storage group ACS routine. [Table 1](#) summarizes the possible assignments of *value*.

Table 1. Read-Write Variable Assignments

```
SET &STORCLAS EQ 'SCNORM'  
SET &STORCLAS = 'SCNORM'  
SET &DATACLAS = &DEF_DATACLAS      /* RACF read-only variable */  
SET &MGMTCLAS = 'SCRATCH5'  
SET &STORGRP = 'SG1','SG2','SG3'    /* list of values */  
SET &DATACLAS = ''                  /* null value assignment */
```

You must have RACF installed and ACSDEFAULT(YES) specified in the IGDSMSxx to assign the &DEF_DATACLAS, &DEF_STORCLAS, or &DEF_MGMTCLAS. Otherwise a null value is assigned to a read-write variable.

Parent topic:Statements



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DO statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can group a collection of ACS language statements using a DO statement paired with an END statement. The DO statement can follow an IF-THEN clause, an ELSE clause, or a SELECT-WHEN group.

DO <group of statements> END

group of statements can consist of zero or more ACS language statements.

[Figure 1](#) shows an example of a DO statement:

[Figure 1. Example of a DO Statement](#)

```
DO
  WRITE 'No Payroll allowed'
  EXIT CODE(1)
END
```

Parent topic:[Statements](#)



©Copyright IBM Corporation 1990, 2014

IF statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the IF statement for conditional statement execution. You must always follow an IF statement with a THEN clause. The THEN clause can be a single statement or a DO-END group of statements. You can optionally follow a THEN clause with an ELSE clause. If you specify the ELSE clause, you must follow it with either a single statement or a DO-END group of statements. If you want to specify the ELSE clause but do not want to follow it with an executable statement, follow it with an empty DO-END pair.

If the result of the IF statement comparison is *true*, the THEN clause is executed. If the result is false, the ELSE clause is executed. If you omit the ELSE, processing continues with the next sequential statement after the THEN clause.

IF relational expression **THEN** clause **ELSE** clause relational expression can be a single comparison or it can be multiple comparisons joined by Boolean operators. **THEN** is a mandatory keyword. **clause** can contain a single statement, a DO-END group of statements, or a SELECT statement. **ELSE** is an optional keyword. An ELSE clause is mandatory if you specify the ELSE keyword.

For information on what constitutes a valid relational expression, refer to [Comparison rules](#).

Figure 1 shows two examples of IF statements: [Figure 1. Examples of IF Statements](#)

Example of a specified ELSE:

```
IF &DSOWNER = 'BILL' THEN
  SET &STORCLAS = 'ONE'
ELSE
  SET &STORCLAS = 'TWO'
```

Example of a null ELSE:

```
IF &DSOWNER = 'BILL' THEN
  SET &STORCLAS = 'ONE'
ELSE
  DO END
  SET &STORCLAS = 'TWO'
```

The statement on the left sets &STORCLAS equal to ONE or TWO, depending on the value of &DSOWNER

The statement on the right sets &STORCLAS equal to TWO, regardless of the value of &DSOWNER. If &DSOWNER equals 'BILL', then &STORCLAS is set to 'ONE'. The ELSE clause is skipped, and execution falls to the next statement, which changes &STORCLAS to 'TWO'. If &DSOWNER does not equal 'BILL', execution falls to the ELSE, which results in no assignment to &STORCLAS. Then execution proceeds to the next sequential statement, which sets &STORCLAS to 'TWO'.

Parent topic:[Statements](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SELECT statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the SELECT statement to write conditional statements in sequential form rather than IF-THEN-ELSE form. A SELECT statement consists of a SELECT keyword, one or more WHEN clauses, an optional OTHERWISE clause, and an END statement. You can specify the SELECT statement in one of two forms. In the first form shown below, you include the variable being tested after the SELECT statement. In the second form shown below, you include the variable being tested after the WHEN keyword.

The first true WHEN condition is executed, and the remaining WHEN conditions are ignored. If none of the WHEN conditions is true and there is an OTHERWISE clause, then the OTHERWISE action is taken.

```
SELECT (variable)
WHEN (value) <action>
.
.
<WHEN (value) <action>>
<WHEN (value) <action>>
<OTHERWISE <action>>
END
```

where: *variable* can be any read-only or read-write variable. At least one WHEN keyword is mandatory. *value* can be a constant or a FILTLIST name. *action* can be an ACS statement, SELECT group, or a DO END group. OTHERWISE is an optional keyword.

```
SELECT (&DSOWNER)
WHEN ('IBMUSER1') SET &STORCLAS = 'PAYROLL'
WHEN ('IBMUSER2') SET &STORCLAS = 'TEST'
WHEN ('IBMUSER3') SET &STORCLAS = 'DEVELOP'
OTHERWISE      SET &STORCLAS = 'NORMAL'
```

END

SELECT WHEN (relational expression) <action>

```
<WHEN (relational expression) <action>>
<WHEN (relational expression) <action>>
```

<OTHERWISE <action>> END At least one WHEN keyword is mandatory. *relational expression* can be a single comparison or it can be multiple comparisons joined by Boolean operators. *action* can be an ACS statement, SELECT group, or a DO END group. OTHERWISE is an optional keyword.

[Figure 1](#) shows an example of coding a SELECT statement:

[Figure 1. Example of a SELECT Statement](#)

```
SELECT
WHEN (&DSOWNER = 'IBMUSER') SET &STORCLAS = 'PAYROLL'
WHEN (&DSOWNER = 'IBMUSER2')

  IF &ACCT_JOB = '1234' THEN
    SET &STORCLAS = 'TEST'
  ELSE
    SET &STORCLAS = 'EVERYONE'

WHEN (&DSTYPE = 'TEMP')   SET &STORCLAS = '
WHEN (&HLO = 'CADAM')     SET &STORCLAS = '
OTHERWISE                 SET &STORCLAS = 'COMMON'
```

END

Parent topic: Statements



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

EXIT statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The EXIT statement immediately terminates the operation of an ACS routine.

EXIT <CODE(*n*)> CODE is an optional keyword.*n* is an exit code. A nonzero value for *n* causes the subsequent ACS routines to be skipped and the allocation to fail with no explicit value assigned to the read-write variable in the ACS routine. If you do not specify a value for *n*, it assumes the default value of zero. [Figure 1](#) shows an example of an EXIT statement:

Figure 1. Example of an EXIT Statement

```
PROC STORCLAS
FILTLIST SECVOL INCLUDE(PAY*, REC*) EXCLUDE('PAYR20', 'REC195')
FILTLIST VALID_UNITS INCLUDE('3380','3390','SYSDA','')
IF &UNIT ~=&VALID_UNITS THEN
  DO
    SET &STORCLAS =
    EXIT
  END
  IF &ALLVOL ~=&SECVOL THEN EXIT CODE(22)
END
```

If the first IF statement is *true* (&UNIT does not match any unit named in the VALID_UNITS filter criteria), then execution of this ACS routine terminates immediately. Allocation proceeds, because the exit code is zero, the default. If the second IF statement is *true* (none of the input volumes match the SECVOL FILTLIST criteria), then execution of this ACS routine terminates immediately and the allocation fails. The value for CODE, 22, is set and displayed as part of the allocation failed error message written to the end user.

Parent topic:[Statements](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

WRITE statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the WRITE statement to issue a message to an end user at execution and allocation time. With the WRITE statement, you might notify end users that you are removing a particular storage class, you might inform end users that they lack sufficient authority to use a particular management class, or you might tell an end user that you have moved a tape data set to DASD.

In a TSO/E ALLOCATION environment, the text of the WRITE message displays only if the allocation fails.

Exception: The WRITE message is never displayed for any product or application that uses Dynamic Allocation (for example: DFSMSdss, DFSORT SORTWKnn data sets).

Under certain conditions related to data set stacking, SMS invokes ACS routines more than once. Consequently, you might want to take special care when using WRITE statements in order to avoid duplicates in the job log.

A WRITE message can contain up to 110 characters of text and variables. The message substitutes the value of a variable for a variable name. With the exception of &ANYVOL and &ALLVOL, you can use any of the read-only variables in write statements. All numerical values are in hexadecimal when displayed in write statements.

You must enclose the message in single quotation marks. If you want a single quotation mark to be part of the message, use two single quotation marks to represent it.

You can use continuation characters (+, -) to continue text or a literal onto a subsequent line. The closing single quotation mark signifies the end of text. A variable can be referenced on a subsequent line after the closing quotation mark without a continuation character.

For example, the following WRITE message: `WRITE 'This line''s short.'` displays as: `This line's short.`

The following WRITE message: `WRITE 'THE DATACLAS IS EQUAL TO'`
`&DATACLAS`

displays as: `THE DATACLAS IS EQUAL TO DEFAULT`

A nine-character system message id and a single blank character precede your message to the end user. At execution and allocation time, an end user can receive a maximum of five messages. If any more messages are generated, a sixth and final message indicates that additional messages have been generated, but the additional messages are not displayed.

WRITE 'message' message is written with the end user's job messages.

Assuming the value of &STORCLAS is 'SC1:', the following WRITE message

'WARNING - &STORCLAS SPECIFIED (' &STORCLAS ') IS NOT ALLOWED'

WRITE

displays as: `IGD01005I WARNING - &STORCLAS SPECIFIED (SC1) IS NOT ALLOWED`

When multiple storage group names are assigned to &STORGRP, only the first name in the list of storage group names is displayed in a WRITE message. For example, if the values for &STORGRP are 'SG1', 'SG2', and 'SG3', the following

WRITE statement: `WRITE ' &STORGRP IS '&STORGRP' '`

displays: `IGD01010I &STORGRP IS SG1`

and the following WRITE statement: `WRITE '&STORGRP SET TO SG1, SG2 AND SG3 '`

displays: `IGD01010I &STORGRP SET TO SG1, SG2 AND SG3`

Parent topic:Statements

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

END statement

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The END statement concludes an ACS routine, a DO group, or a SELECT statement. [Figure 1](#) shows an example of an END statement:

Figure 1. Example of an END Statement

PROC STORCLAS
(source code)

END

Parent topic:[Statements](#)



©Copyright IBM Corporation 1990, 2014

Sample ACS routine

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1 illustrates some techniques for using the ACS routines.
Figure 1. Production ACS Routine for Storage Class_{PROC STORCLAS}

```
*****
/* THIS IS THE PRODUCTION SELECTION SPECIFICATION FOR SETTING STORCLAS */
*****
```

```
FILTLIST DBVOLS INCLUDE(IMS*,DB2*)
    EXCLUDE('IMS053','DB2007')                                /* ALL DATABASE VOLUMES */

FILTLIST DBJOBS INCLUDE(IMS*,PROD*,ACCT*)                   /* ALL DATA BASE JOBS */

FILTLIST VALID_UNITS
INCLUDE('3330','3340','3350','3375','3380','3390','SYSDA','') /* VALID UNITS FOR SMS */

IF &UNIT ^= &VALID_UNITS
```

```
THEN DO
    SET &STORCLAS =
    WRITE 'INVALID UNIT TYPE FOR SMS ALLOCATION'
    EXIT
END
```

```
SELECT
```

```
WHEN (&DSN = SYS1.*)                                     /* SYSTEM DATA */
    SET &STORCLAS = 'SYSTEM'

WHEN ((&ALLVOL = &DBVOLS) && (&JOB = &DBJOBS))          /* DATABASE DATA */
    SET &STORCLAS = 'DBPOOL'

WHEN ((&DSN(3) = 'CLEAR') | (&ANYVOL ^= TSO*))           /* NON-SMS DATA */
    SET &STORCLAS = 'COMMON'

WHEN (&DEF_STORCLAS ^= '')                                /* IF DEFAULTS EXIST */
    SET &STORCLAS = &DEF_STORCLAS;

OTHERWISE SET &STORCLAS = 'COMMON'                         /* ALL OTHER DATA */

END                                                       /* END STORCLAS PROC */
```

The FILTLIST VALID_UNITS INCLUDE statement in [Figure 1](#) does not contain the latest devices. Update the FILTLIST VALID_UNITS INCLUDE statement when new devices are installed at your installation.

Restriction: If the null unit that is illustrated in the FILTLIST VALID_UNITS INCLUDE statement in [Figure 1](#) is not in the valid DASD units FILTLIST, and a null storage class is assigned to allocations that do not have a valid unit (units in the DASD units FILTLIST), then you cannot manage VSAM data sets allocated using IDCAMS.

Parent topic:[Statements](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Quick reference to ISMF commands and line operators

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The abbreviations in the Application column of these tables represent the following:
ACSAutomatic class selection **AG**Aggregate groups **CDS**Control data set **CP**Copy Pool
DCData class **DS**Data set **DV**Optical drive **DVOL**DASD volume **LA**List **LB**Optical library
MCManagement class **OVOL**Mountable optical volume **SC**Storage class
SGStorage group **TL**Tape library **TVOL**Mountable tape volume
Tip: If you specify an equal sign after any DFSMSdss or DFSMShsm line operator, processing occurs in 'last-use mode,' which recalls the last values entered for that particular line operator or command, and DFSMSdfp does not display an entry panel.

[Table 1](#) and [Table 2](#) list the ISMF commands and line operators available to storage administrators. See z/OS DFSMS Using the Interactive Storage Management Facility for the line operators and commands available to the end user.

Table 1. ISMF Commands

Table 2. ISMF Line Operators



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SETCACHE functions and device information

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This topic shows you how to: Maintain the media characteristics
Modify DASD storage control
Provide access to IBM® service support representatives at remote locations

To perform these tasks, you enter line operators or list commands on data set or volume lists and then complete data entry panels. Refer to your online help panels or z/OS DFSMS Using the Interactive Storage Management Facility.

[Maintaining the media characteristics of a volume](#)

[Modifying DASD storage control characteristics](#)

[Providing remote authorization codes](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Maintaining the media characteristics of a volume

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use ISMF to examine and manage the characteristics of your data stored on DASD volumes. Eight line operators are available to perform the following media maintenance tasks: Inspect a subset of a volume for magnetic surface defects. ISMF also provides options to help you remedy the problem. Examine a drive and your data to determine if errors exist. This function helps you distinguish between errors caused by drive problems and errors caused by media problems. Initialize a DASD volume for use in an MVS system. Change the volume label of a DASD volume. You can specify a new volume serial number or a new owner ID. Generate a job stream that is used to change a volume from a nonindexed format VTOC(OSVTOC) to an indexed format VTOC(IXVTOC) or vice versa. Reset a device that has been write inhibited, reset an indefinite status condition, or clear a fence status of a path, a device, or both. Generate a job stream that is used to perform the procedures necessary for installation, head-disk assembly (HDA) replacement, and physical movement of IBM® DASD. Perform the track validation functions of medial initialization with the problem determination and data verification functions of the ANALYZE command, and also the INSPECT functions if required. For information on the ICKDSF functions that ISMF uses to perform these functions see Device Support Facilities (ICKDSF) User's Guide and Reference.

Parent topic:[SETCACHE functions and device information](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Modifying DASD storage control characteristics

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the SETCACHE line operator to control the caching in a storage control unit.

You can use the SETCACHE line operator from the Volume List panel to modify the storage control unit characteristics. The following characteristics can be modified with the SETCACHE line operator: Performance characteristics, which control the caching of read or write requests. You can modify these characteristics to reduce the frequency of access to DASD. Availability characteristics, which control the automatic duplication of the device activity onto a secondary volume. You can also modify these characteristics to move the contents of one volume to another while the data is active. Resources characteristics, which control how data that is stored in the cache is destaged or discarded. You can directly manage data in cache and nonvolatile storage. Enter the SETCACHE line operator against a particular volume from the Volume List panel. You can invoke SETCACHE with a PERFORMANCE, AVAILABILITY, or RESOURCES parameter depending on the characteristic you would like to modify. You can also enter SETCACHE without a parameter. On a 3990 Storage Control unit with cache, ISMF takes you to the SETCACHE Features Entry panel. From this panel you can select the characteristic you would like to modify. ISMF takes you to either the PERFORMANCE, AVAILABILITY, or RESOURCES Entry panel. On the Model 3880-13/23, ISMF takes you to the PERFORMANCE Entry panel where you can do your modifications.

The specific features that you can modify depend on the model of the storage control unit that you are working with. You must also have storage administration authorization to use the SETCACHE line operator.

In order to use the SETCACHE function under ISMF, you must modify the IKJTSOxx member in SYS1.PARMLIB. IDCAMS must be added to AUTHPGM NAMES, and SETCACHE must be added to AUTHCMD NAMES.

[Modifying caching characteristics](#)

[Modifying duplexing characteristics](#)

[Modifying destaging characteristics](#)

[Submitting jobs](#)

Parent topic:[SETCACHE functions and device information](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Modifying caching characteristics

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the Setcache Performance Entry panel to modify the caching characteristics of the storage control units. If you are using a Model 3990 Model 3 Storage Control, you have support for the caching of both read and write requests. If you are using a 3880-13 or 3880-23 Storage Control, you have support for the caching of read requests. In the terminology of the SETCACHE line operator, the caching of read requests is controlled by the READ CACHE or READ and SYSTEM CACHE functions. The caching of write requests is controlled by the DASD FAST WRITE, NON-VOLATILE STORAGE (NVS), and CACHE FAST WRITE functions.

Restriction: The cache/DFW of the IBM® Enterprise Storage Server® (ESS) is on by default, and you are not allowed to modify it. In addition, the ESS does not support the dual-copy function. You receive a message, with a condition code of 12, indicating these changes if you issue any of the cache/DFW or dual-copy commands.

[Read caching/Read and system caching](#)

[Write caching](#)

Parent topic:[Modifying DASD storage control characteristics](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Read caching/Read and system caching

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can establish or terminate the READ CACHE or READ and SYSTEM CACHE functions to control the caching of read requests. Read caching is performed on a track basis. When you need data from a volume, a read request is made to the storage control unit. The storage control unit reads the data from the volume and sends it back to you. At the same time, a copy of all the data on the track that the storage control unit accesses is saved in cache that resides in the storage control unit. Any subsequent requests that you make for data on the track can be satisfied from the cache. Read performance is improved when the data for your read requests come from the cache.

You can specify that the storage control unit perform this type of read caching on a volume level. Caching can be turned on and off for individual volumes. This feature gives you greater flexibility to manage the volumes that are using the caching resources.

Parent topic:[Modifying caching characteristics](#)



©Copyright IBM Corporation 1990, 2014

Write caching

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If your storage control unit supports the caching of write requests, you can use the SETCACHE line operator to establish or terminate the DASD FAST WRITE and CACHE FAST WRITE functions. Read and system caching must be active in order for fast write (both DASD and CACHE) to be in effect.

DASD FAST WRITE provides support for the caching of write requests. NON-VOLATILE STORAGE (NVS) provides backup storage for this function. When you write data out to a volume, a write request is made to the storage control unit. The storage control unit accepts the data and puts it in the storage control unit cache. Later, the storage control unit schedules the data for writing.

When the data is stored in the storage control unit cache, it is also written to NVS. NVS is a buffer backed up by battery power that can maintain data up to 48 hours. NVS provides insurance against the loss of data written to the storage control unit cache and not yet sent to DASD.

The CACHE FAST WRITE feature allows part of the subsystem storage to be set aside as work space. Data written out in CACHE FAST WRITE mode is not staged for writing to DASD unless you make an explicit request or the system needs to reuse some of the cache space. CACHE FAST WRITE is for temporary work files only because data in cache is eventually discarded.

Parent topic:[Modifying caching characteristics](#)



Modifying duplexing characteristics

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the Setcache Availability Entry panel to manage the availability of a duplex pair of devices that control the duplication and recovery of data. The AVAILABILITY characteristics affect only data that is on DASD.

To establish a duplex pair, you must specify: A primary and a secondary device. The type of synchronization between the two devices. You can specify the rate of copy in the synchronization. Once duplexing is established, the storage control unit writes data to the primary device and simultaneously places a copy of it in cache along with control information in NVS. At a later time, the data in cache is written out to the secondary device, and the control information in NVS is updated.

If duplexing is interrupted, the dual copy pair is suspended. The storage control still writes information to NVS, but data is written to only the primary device. When duplexing is resumed, the secondary device is placed back in synchronization by the storage subsystem.

You also can use the Setcache Availability Entry panel to move the contents of one device to another. The panel allows you to establish a duplex-pair between two devices to copy data. The pair is immediately broken when all data has been copied. The original device is then available for maintenance.

Parent topic:[Modifying DASD storage control characteristics](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Modifying destaging characteristics

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the Setcache Resources Entry panel to directly manage the data that has already been written to the cache and NVS. This function of the SETCACHE line operator allows you to: Destage dataDiscard pinned dataSet all the volumes and the subsystem back to their default status

Once data has been written to the cache or NVS, the Resources Entry panel allows you to write the data to DASD. If this destaging process fails, the data is pinned in cache or NVS. The Resources Entry panel allows you to discard this pinned data. All the volumes attached to a storage subsystem along with the subsystem itself can be set back to their default status by reinitializing the storage subsystem. Restriction: reinitializing a subsystem causes all duplex pairs to be lost and any fast write data in cache and NVS to be discarded.

Parent topic:[Modifying DASD storage control characteristics](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Submitting jobs

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When you specify the SETCACHE line operator, ISMF takes you through the entry panels you need to complete given the functions you choose to perform. ISMF eventually asks you to specify whether the job is performed in the foreground or the background. In the foreground, ISMF uses the IDCAMS TSO SETCACHE support. ISMF subsequently redisplays updated versions of the entry panels with their new values after the job has executed. The list panels are not updated unless a REFRESH is done. For some functions, execution in foreground can occupy your TSO terminal for 20 minutes or more. In these cases, execution in the background is recommended.

When you specify that the SETCACHE job be performed in background, ISMF displays the IDCAMS Job Submission Entry panel. From this panel, you can submit the job or specify a data set where ISMF saves the job. The Job Submission Entry panel also optionally allows you to change the JCL that ISMF uses to run the job. At each level of the SETCACHE panels, the ISMF online help provides detailed descriptions for each of the fields.

Once ISMF takes you into the Setcache Entry panel, the online help provides detail descriptions for each of the fields.

Parent topic:[Modifying DASD storage control characteristics](#)



©Copyright IBM Corporation 1990, 2014

Providing remote authorization codes

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The RAUTH line operator provides an easy way to acquire the remote access authorization codes for your IBM® 3990 Storage Control units:

When you enter the RAUTH line operator from the Volume List panel, ISMF returns passwords on the RAUTH display panel.

Each password is valid for one hour. Once the password is displayed, ISMF does not display the password again. If you specify RAUTH again, ISMF returns new passwords.

Using these passwords, an off-site IBM Service Representative can log on to all models of the IBM 3990 Storage Control unit family (Models 1, 2, 3, and 6). The Service Representative can then provide help diagnosing and correcting problems. You must have storage administrator authorization to use the RAUTH line operator. Because the standard support facilities of ISPF are available, for example, printing the screen, it is your responsibility to maintain the security of the passwords. Also, RACF® DASDVOL alter authority is required over the volume the request is made against. ISMF online help provides a description of each of the fields on the RAUTH display panel.

Parent topic:[SETCACHE functions and device information](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Space utilization and capacity planning

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can collect information on active and inactive data set space utilization and capacity planning by using the ISMF Data Collection application to produce a DCOLLECT job. DCOLLECT provides measurement data in a sequential data set (also called a flat file), which can be used as input to the DFSSrmm Report Generator for creating customized reports and to other applications such as billing and report formatting.

You can use it to produce measurement data on: active data sets, which are data sets that have not been backed up or migrated; inactive data sets, which are data sets that *have* been backed up or migrated; capacity planning, which concerns volume capacity and usage; and volumes. Information can be obtained on the following: Active data setsVolumesMigrated data sets (DFSMSHsm)Backup data sets (DFSMSHsm)DASD capacity planning (DFSMSHsm)Tape capacity planning (DFSMSHsm)

DCOLLECT is an access method services function.

[Using the data collection application](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using the data collection application

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use ISMF panels to generate the job control language for the access method services DCOLLECT command. The job is expected to be executed while the DFSMS environment is active.

Perform the following steps to generate the JCL for the access method services DCOLLECT command.

Select option C, Data Collection, on the ISMF Primary Option Menu for storage administrators. This displays the Data Collection Entry panel.

Supply values on the panel. The fields on the entry panel for the Data Collection application are primed to collect data only for active data sets and volumes. You can also collect information on migration data, backup data and capacity planning data. You can specify Y (Yes) in any combination of the fields for Select Data Collection options.

If you are collecting information on active data sets or volumes, use the DOWN command to reach the next page of the Data Collection Entry panel.

Supply values for one or more volumes or storage groups about which DCOLLECT is to collect information..

If you want to specify volumes to exclude, use DOWN command to reach the last page of the Data Collection Entry panel.

Supply values to specify volumes to exclude.

When you have finished entering values on the Data Collection Entry panel, press Enter to display the DCOLLECT Job Submission Entry panel. On this panel, you can choose to submit the job you just specified, or save the job to a data set, which you can then edit. You can edit the job statement on this screen before submitting the job. The JCL produced uses the access method services execute statement that you have already set up in the profile as a default. You can specify that you want to edit the execute statement on the Data Collection Job Submission Entry panel. When you press Enter, you can edit the execute statement.

The following list explains the fields on page 1 of the Data Collection Entry panel.

DATA SET INFORMATION If you want information collected on active data sets selected by storage group or volume, specify Y, yes, in this field. Otherwise, specify N, no. The default is Y.

VOLUME INFORMATION If you want to collect information on active volumes selected by storage group or volume, specify Y, yes, in this field. Otherwise, specify N, no. The default is Y.

MIGRATION DATA If you want to collect information on migration data, specify Y in this field. Otherwise specify N, no. If you specify Y, you must specify a migration data set name in the MIGRATION DATA

SET NAME field. The default is N.

BACKUP DATA If you want to collect information on backup data, specify Y in this field. Otherwise specify N, no. If you specify Y, you must specify a backup data set name in the BACKUP DATA SET NAME field.

The default is N.

CAPACITY PLANNING DATA If you want to collect information on capacity planning, specify Y in this field. Otherwise specify N, no. If you specify Y, you must specify both a migration data set name and a backup data set name. The default is N.

SMS DATA Specify Y, yes, in this field if you want to collect SMS information. The default is N.

DATA SET NAME You must enter the name of an output data set in this field. DCOLLECT puts the information it collects in the data set you specify. The data set can either be an existing data set or a data set that is allocated during the data collection job while DFSMS is active.

OPTIONAL PASSWORD If the data set is not system-managed and it has a password, you must specify the password in the OPTIONAL PASSWORD field. Password

protection is less secure than RACF® protection. The data set must be cataloged. This field is ignored for system-managed data sets. **REPLACE CONTENTS**If you want DCOLLECT to overlay any data which is already contained in the output data set, specify Y, yes, in the REPLACE CONTENTS field. If you want DCOLLECT to append newly collected data to the data already contained in the output data set, specify N, no, in the REPLACE CONTENTS field. N, no, is the default, that is, data already contained in the output data set is not erased when the job is submitted. If the data set has never been written in, it does not matter whether you specify Y or N. **NUMBER OF DATA SETS**If you are specifying an output data set that is to be allocated during the data collection job, you should estimate the total number of data sets which reside on the volumes about which you are collecting data. Specify this number in the NUMBER OF DATA SETS field. This number is used to estimate the size of the output data set. If you save the JCL generated, instead of submitting it directly from this application, you can edit the DD statement for the output data set to change the space calculation. **MIGRATION DATA SET NAME**If you are collecting migration data or information for capacity planning, you must specify a data set name in the MIGRATION DATA SET NAME field. This data set must contain migration information. For DFSMShsm, this data set is the Migration Control Data Set (MCDS) for either the system you are running on or the system about which you want information. This data set must be accessible to the system running the DCOLLECT job. **BACKUP DATA SET NAME**If you are collecting backup data or information for capacity planning, you must specify a data set name in the BACKUP DATA SET NAME field. This data set must contain backup information. For DFSMShsm, this data set is the Backup Control Data Set (BCDS) for either the system you are running on or the system about which you want information. This data set must be accessible to the system running the DCOLLECT job. When you are collecting capacity planning information, you must specify migration and backup data sets that are on the same DFSMShsm system.

On the second page, you specify one or more volumes or storage groups about which DCOLLECT is to collect information. If you specified Y (Yes), in the DATA SET INFORMATION or VOLUME INFORMATION fields on the first page of the DCOLLECT entry panel, you must specify at least one volume or storage group on this panel.

SPECIFY VOLUMES In this field, you can specify the names of the volumes from which information is collected. The information collected can be about the volumes themselves or about the active data sets they contain. You can specify a full volume name (for example, SYSTSO), a partial volume name (for example, SYST*), an asterisk to indicate all online volumes, or six asterisks to indicate the system residence (SYSRES) volume. DCOLLECT ignores repeated entries. If you specify a volume more than once, DCOLLECT collects information on that volume only once. This includes volumes belonging to storage groups that you have specified. If you specify a single asterisk, DCOLLECT ignores all other entries in the SPECIFY VOLUMES field. Although you can specify only 50 volumes on this panel, you can edit the job control language produced to specify up to 255 separate volumes.

SPECIFY POOL STORAGE GROUPS In this field, you can specify the names of pool storage groups from which information is collected. Information from all of the volumes in a specified storage group is collected. Enter only pool storage group names. Only pool storage groups are appropriate for data set or volume information collection. The information collected can be on the volumes themselves or on the active data sets they contain. You must specify the full name of a storage group. You cannot specify a partial name or an asterisk for a storage group name. Although you can specify only 10 storage groups on this panel, you can edit the job control language produced to specify up to 255 storage groups.

If you are collecting information only on active data sets, DCOLLECT collects information on the active data sets on the volumes and, in the storage groups you specify. It does not collect information on other data sets on those volumes and in those storage groups.

If you are collecting information only on active volumes, DCOLLECT collects information on the volumes and in the storage groups you specify. It ignores information about the data sets on those volumes.

Parent topic:[Space utilization and capacity planning](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using data set separation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Data set separation allows you to designate groups of data sets in which all SMS-managed data sets within a group are kept separate, on the physical control unit (PCU) level or the volume level, from all the other data sets in the same group.

Overview

[Syntax for creating a data set separation profile](#)

[Volume selection for data set separation by volume](#)

[Data set requirements for a data set separation profile](#)

[Creating multiple data set separation profiles](#)

[Using data set separation with generation data groups and striping](#)

[Environmental conditions affecting data set separation](#)

[Factors affecting code path length](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Overview

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To use data set separation, you must create a data set separation profile and specify the name of the profile to the base configuration. During allocation, SMS attempts to separate the data sets listed in the profile.

A data set separation profile contains at least one data set separation group. Each data set separation group specifies whether separation is at the PCU or volume level, whether it is required or preferred, and includes a list of data set names to be separated from each other during allocation.

Restriction: You cannot use data set separation when allocating non-SMS-managed data sets or with any process that does not use SMS volume selection, such as a full-volume copy utility (for example PPRC).

Recommendation: Use data set separation only for a small set of mission-critical data for the following reasons: When separation is at the PCU level, an existing data set might be allocated to one or more PCUs from which the current allocation is to be separated. In this case, SMS either rejects all volumes on the PCUs from the SMS volume preference list or places them at a less-preferred position. Volume rejection might drastically reduce the number of eligible volumes or result in allocation failures if SMS rejects all volumes. The use of wildcard characters in the data set names that need to be separated from each other might drastically reduce the number of eligible volumes or result in allocation failure if SMS rejects all volumes. Data set separation can affect system performance. See the performance concerns listed in [Factors affecting code path length](#).

For information about specifying the name of the data set separation profile to the base configuration, see [Specifying the DS separation profile](#).

The following table lists the topics that are contained in this section:

Parent topic:[Using data set separation](#)



Syntax for creating a data set separation profile

z/OS DFSMSdfp Storage Administration
SC23-6860-01

A data set separation profile contains one or more data set separation groups. Use the following syntax to create a data set separation group:

```
SEPARATIONGROUP|SEP (PCU | {VOLUME|VOL})TYPE ({REQUIRED|REQ|R} | {PREFERRED|PREF|P})DSNLIST|DSNS|DSN(data-set-name,data-set-name[,data-set-name,...])
```

where **SEPARATIONGROUP(PCU)**Indicates that separation is on the PCU level.
SEPARATIONGROUP(VOLUME)Indicates that separation is on the volume level.
VOLUME may be abbreviated as VOL.**TYPE(REQUIRED)**Indicates that separation is required. SMS fails the allocation if the specified data set or data sets cannot be separated from other data sets on the specified level (PCU or volume). REQUIRED may be abbreviated as REQ or R.**TYPE(PREFERRED)**Indicates that separation is preferred. SMS allows the allocation if the specified data set or data sets cannot be separated from other data sets on the specified level. SMS allocates the data sets and issues an allocation message that indicates that separation was not honored for a successful allocation. PREFERRED may be abbreviated as PREF or P.**DSNLIST(data-set-name,data-set-name,...)**Specifies the names of the data sets that are to be separated. You must specify at least 2 names. The data set names must follow the naming convention described in z/OS MVS JCL Reference. You can specify the same data set name in multiple data set separation groups. Wildcard characters are supported, beginning with the third qualifier. For example, you could specify BJONES.TEST.* but not BJONES.*. The wildcard characters are as follows:
*Indicates that either a qualifier or one or more characters within a qualifier can occupy that position. An asterisk can precede or follow a set of characters.
**Indicates that zero or more qualifiers can occupy that position. When ** is the last qualifier, one or more qualifiers can occupy that position. A double asterisk cannot precede or follow any characters; it must be preceded or followed by either a period or a blank.
%Indicates that exactly one alphanumeric or national character can occupy that position. You can specify up to eight % characters in each qualifier. If only one data set name is specified with DSNLIST, the data set name must contain at least one wildcard character.

[Deprecated syntax](#)

[Writing comments](#)

[Indicating continuation and termination](#)

[Example](#)

Parent topic:[Using data set separation](#)

©Copyright IBM Corporation 1990, 2014

Deprecated syntax

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following earlier form of the syntax for SEPARATIONGROUP is tolerated by z/OS® V1R11. It supports separation at the PCU level only.

`SEPARATIONGROUP | SEPFAILLEVEL | FAIL({PCU|NONE})DSNLIST | DSNS | DSN(data-set-name[,data-set-name,...])`

You specify either required or preferred data set separation by using one of the values for the FAILLEVEL keyword: **FAILLEVEL(PCU)**Indicates that separation on the PCU level is *required*. SMS fails the allocation if the requested data set cannot be separated on the PCU level from other data sets that are listed in the SEPARATIONGROUP. An allocation failure message is issued that indicates the number of volumes that were rejected by the data set separation profile. For the current syntax, see TYPE(REQUIRED).

FAILLEVEL(NONE)Indicates that separation on the PCU level is *preferred*. SMS allows the allocation if the requested data set cannot be separated on the PCU level from other data sets that are listed in the SEPARATIONGROUP. SMS allocates the data sets on the same PCU and issues an allocation message that indicates that separation was not honored for a successful allocation. During volume selection, SMS treats all volumes on the same PCU level as less-preferred. For the current syntax, see TYPE(PREFERRED).

When this syntax is used, data set names listed in a data set separation group must not contain quotation marks or wildcards, and must follow the naming convention described in z/OS MVS JCL Reference. You can specify the same data set name in multiple data set separation groups.

[Migrating to the new syntax](#)

Parent topic:[Syntax for creating a data set separation profile](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Migrating to the new syntax

z/OS DFSMSdfp Storage Administration
SC23-6860-01

If you have the deprecated syntax in your data set separation profile, consider updating the profile to use the current syntax. Toleration PTFs are available that provide support for the z/OS® V1R11 syntax for PCU-level separation on lower level systems, with the exception of wildcard characters in data set names. With the PTFs, separation groups defined with syntax that is not supported, such as SEP(VOL) or wildcard characters in the data set name list, are ignored.

[Example](#)

[Parent topic:Deprecated syntax](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Example

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following examples show definitions that are equivalent with the deprecated and current syntax: Table 1. PCU-Level Syntax Examples

Parent topic:[Migrating to the new syntax](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Writing comments

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use one of the following two methods for entering comments into a data set separation group:

!SMS ignores all text on a line following the comment character (!). To continue the comment on the next line, you must also precede the next line with the comment character. For example:

! This illustrates a comment which continues
on the next line.

/* */SMS ignores all text within the comment start indicator /*) and comment end indicator (*). For example:

This illustrates a comment which continues
on the next line.
-----*/

Parent topic:[Syntax for creating a data set separation profile](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Indicating continuation and termination

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the following notational conventions to indicate continuation or termination of a data set separation group:

- Use this character to continue a SEPARATIONGROUP statement onto a following line. You cannot continue a keyword or keyword value in the middle of the keyword or keyword value. You must specify continuation characters outside of comments. For example: `SEP(PCU)-`

/*SMS Control Data Sets*/-
; You must use this termination character to end a SEPARATIONGROUP statement. For example: `SEP(PCU)-`

```
TYPE(REQ)-  
DSNLIST(SMS.PROD.SCDS-  
,SMS.PROD.ACDS);
```

Parent topic:[Syntax for creating a data set separation profile](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Example

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following example of a data set separation profile contains two comments and two data set separation groups:

```
/*-----  
SMS CONTROL Data Sets  
-----*/  
SEPARATIONGROUP(PCU)-  
  TYPE(PREF)-  
  DSNLIST(SMS.PROD.SCDS, SMS.PROD.ACDS, SMS.PROD.COMMDS);  
  
/*-----  
JES CHECKPOINT Data Sets  
-----*/  
SEP(PCU)  
  TYPE(REQ)  
  DSNS(SYS1.JESCKPT1,- !primary  
        SYS1.JESCKPT2,- !secondary  
        SYS1.JESCKPT3);   !tertiary
```

Parent topic:[Syntax for creating a data set separation profile](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Volume selection for data set separation by volume

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When separation by volume is requested for a data set, the order of preference for volume selection is as follows: Candidate volumes that are in different extent pools than volumes on the separation volume list are most preferredCandidate volumes that are not on the separation volume list but are in the same extent pools as the volumes on the separation volume list are the next preferredWhen separation by volume is required, SMS must allocate the data set onto the volumes that are not used by any of the data sets in the same separation groupWhen separation by volume is preferred, the candidate volumes that are already occupied by other data sets in the separation group are ranked lower and are less preferred.

Parent topic:[Using data set separation](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data set requirements for a data set separation profile

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The profile data set has these requirements: Organization can be either sequential or partitionedRecord format must be fixed or fixed-block (RECFM=FB).It must be catalogued. Catalog the profile data set in the MASTERCAT to avoid a deadlock between SMS and CATALOG during IPL with contention in the SMS ASID, or a SYSZTIOT might occur.

When modifying the profile, use a disposition of OLD (DISP=OLD). The data in the profile might be corrupted if you use a disposition SHR.

Parent topic:[Using data set separation](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Creating multiple data set separation profiles

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can create multiple data set separation profiles, but you can specify only one in an SMS base configuration (SCDS).

If you have created multiple SMS configurations, each configuration can share the same data set separation profile, or each can specify a different data set separation profile. Once an SMS configuration is activated, all systems within the SMS complex will share the same data set separation profile that is specified in the base configuration.

SMS reads the data set separation profile when SMS initializes with an active configuration, when SMS restarts, and when a configuration is activated or switched.

Parent topic:[Using data set separation](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using data set separation with generation data groups and striping

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Data set striping can use multiple controllers when multistriped. When using data set separation on the PCU level that references multistriped data sets, ensure that the storage group ACS routines select storage groups that contain a sufficient number of PCUs.

Generation data groups (GDGs) that are listed in the data set separation profile must specify the absolute generation and version number (for example, A.B.C.G0001V00). Data set separation does not support relative generation numbers and GDS base names.

When specifying separation of GDGs on the PCU level, ensure that the number of generations that reside on pool volumes does not exceed the number of available PCUs.

Recommendation: Migrate older GDG generations to allow a sufficient number of PCUs for the non-migrated GDG generations.

For more information about GDGs, refer to the discussion of GDG Management Attributes in [Defining management class migration attributes](#). For more information about data set striping, refer to [Striping volume selection](#).

Parent topic:[Using data set separation](#)



©Copyright IBM Corporation 1990, 2014

Environmental conditions affecting data set separation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Under the following environmental conditions, SMS might not separate two data sets that are specified for required or preferred separation: Allocation is not SMS-managed. Allocation is performed on a system in the SMS complex that runs a prior version of SMS that does not support data set separation. During switching or activation of an SMS configuration, the data set separation profile cannot be accessed or fails validation. A service that is allocating a temporary data set name does not provide the real data set name to SMS. Two data sets that are specified to be separated are allocated at the same time by two different tasks or two different systems. For performance reasons, SMS does not serialize data set allocations. A volume is varied online during allocation. An IODF change occurs during allocation. A data set name that is not listed in the data set separation profile is specified during DFSShsm recover. A data set separation profile might have been modified after the configuration was activated. This can result in differences between the data set separation profile data set and the active copy of the data set separation profile. SMS does not perform data set separation during RENAME. SMS does not perform data set separation during DFSShsm migration to level 1 or level 2 storage. SMS *does* perform data set separation during DFSShsm recall. SMS does not perform data set separation during full volume image copy. SMS might not issue the corresponding warning message IGD17372I in those cases where allocation was successful and data set separation was not honored.

Parent topic:[Using data set separation](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Factors affecting code path length

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following factors affect code path length when data set separation is requested:
The number of data set names listed in the data set separation profile. (SMS must scan the profile to determine whether data set separation processing is needed for the current allocation.)
The number of data set names listed within a data set group when data set separation is performed for that group
The use of wildcard characters in the separation data set names
The number of volumes that are eligible for selection.

Parent topic:[Using data set separation](#)



©Copyright IBM Corporation 1990, 2014

Using NaviQuest

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DFSMS NaviQuest is a data and storage management tool for implementing, testing, and verifying the SMS environment. NaviQuest is installed under the Interactive Storage Management Facility (ISMF) Primary Option Menu and uses the standard Interactive System Productivity Facility (ISPF) panel interface.

With NaviQuest you can: Create SMS implementation test cases Initially, you might use NaviQuest in the design and testing phase of your first automatic class selection (ACS) routines and SMS configuration.

Run selected SMS testing functions in batchUpdate SMS configuration values in batch NaviQuest lets you run jobs in batch to list, alter, delete or define a data class, storage class, management class, storage group, aggregate group, or copy pool. You can also use batch mode to display a data class, storage class, management class, and aggregate group.

Create reports interactively or in batch NaviQuest reports can be used during your SMS planning and design, for both DASD and for tape data migration.

Perform ongoing storage administration activities Use NaviQuest for production assurance prior to any changes to your SMS environment. The cross-reference facilities simplify the tasks of testing and verifying SMS configuration changes.

To simplify ongoing activities, the Model Command Generator option, on the Enhanced ACS Management option menu, can be used to create a series of commands or control cards tailored with the data set name or volume serial number that is provided in an ISMF list or DCOLLECT report.

The power and capabilities of SMS require that ACS routines assign the correct data class, storage class, management class, and storage group to each data set. Implementing SMS, or any change that affects the SMS environment, must be tested. The results from the SMS configuration are unpredictable if the ACS routines have logic errors or if there are data sets that the routines are not coded to handle.

Before a SMS configuration is activated on the production system, you can: Easily create test cases to perform extensive testing against test data that represent actual data sets.Run the tests in batch, freeing the workstation for other work.Compare the test results against the *expected* results.

Starting NaviQuest

[Methods for collecting test data](#)

[Data classification methodology](#)

[Testing ACS routines](#)

[Performing storage administration tasks in batch](#)

[Creating SMS online reports](#)

[Additional storage administration functions](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Starting NaviQuest

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Start NaviQuest by selecting option 11, Enhanced ACS Management, from the ISMF Primary Option Menu. This displays the NaviQuest Primary Option Menu. When you select one of the options from the NaviQuest Primary Option Menu, each successive panel guides you through the choices available for that function.

[Terminology](#)

Parent topic:[Using NaviQuest](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Terminology

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Several new terms are unique to NaviQuest. You will need to familiarize yourself with them before you begin to use this facility.

The following terminology and concepts are used throughout this topic: **base line**

test setA special case test set, which includes all test cases for data sets not planned for SMS management. Expected values are nulls (""); the associated special subtype prefix is NEVR.**data classification**The process of determining the data types in your installation and identifying data subtypes that require specific SMS data services.**data subtype**Groupings of data sets from a data type needing identical SMS services, such as performance, backup, migration, or deletion.

NaviQuest testing is managed at the subtype level.**data type**Major groupings of data. TSO, test, batch production, online production, system, and temporary are examples of data types that most installations have.**errors**Results that are different from the expected results. Errors are a subset of exceptions.**exceptions**A test case whose results differ from the saved expected results. For a regression test case, this exception is also an error; for an initial test case, this exception might or might not be an error. If the result is equal to the expected value, there is no error.

expected resultsValues that you want assigned by the ACS routines for data class, storage class, management class, and storage group for a specific subtype when the ACS routines perform correctly.**initial test**First-time testing of a single data subtype performed prior to converting the data to SMS. The test may include regression tests for other data subtypes that have already been tested successfully.

phase or implementation phaseData conversion to SMS management of one or more data subtypes with the same data type.**phase test set**Group of subtype test sets that define all data subtypes that make up a phase.

Phase test sets are converted to SMS management within a single phase.

regression testTesting of data subtypes that have already been successfully tested, along with the initial test of the current phase

subtype resultsValues that are assigned by the ACS routines for data class, storage class, management class, and storage group for a specific test case. These results might or might not be correct.**saved expected results**Results that have been saved in the test case after the successful initial test of the subtype test set. The results are used by the ACS comparison function during regression testing.

subtype prefixA unique 1-to-4 character prefix associated with each data subtype. This prefix is used to relate the data classification data subtype to the subtype test set and to group all test cases together for a single subtype test set.

subtype test setA group of test cases for all data sets associated with the same data subtype. All test cases within one subtype test set have the same expected results.

testbed libraryA partitioned data set (PDS) that contains one member to define each test case.

test caseParameters associated with a single data set that is tested using ISMF option 7.4.3.

Parent topic:[Starting NaviQuest](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Methods for collecting test data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You must become familiar with the methods used to create test data before you begin testing ACS routines.

The procedures for collecting test data are documented in [Testing procedures](#).

NaviQuest collects data for testing from any of four sources: ISMF listsDCOLLECT dataSMF dataVMA data

Although you can successfully use all four sources to create test data, there are restrictions or considerations associated with choosing one method over another.

[ISMF lists](#)

[DCOLLECT data](#)

[SMF data](#)

[VMA data](#)

Parent topic:[Using NaviQuest](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ISMF lists

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Using the standard facilities of ISMF-saved lists, you can select the data sets that you want NaviQuest to enter into the testbed library as SMS test cases. ISMF filtering capabilities help you to select data associated with one category of data (data subtype) at a time. When ISMF lists are used to create test cases, you can use NaviQuest's ISMF batch capabilities to generate and save the list.

Consideration: ISMF lists provide greater flexibility. For example, you can tailor the list by removing data sets.

Parent topic:[Methods for collecting test data](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DCOLLECT data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

DCOLLECT data can be used as input to create test cases. The DCOLLECT input is created as an independent batch job by using either the IDCAMS utility program or the ISMF "Data Collection" (option C).

Consideration: The IDCAMS DCOLLECT function runs in batch and is faster than ISMF lists.

Parent topic:[Methods for collecting test data](#)

 | 

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SMF data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Input from system management facility (SMF) data on new allocations can be used to create test cases. This SMF input data is created by using the storage class ACS exit code available in the sample library SYS1.SACBCNTL. Once the exit is installed and a SMS minimal or null configuration is running, all new allocations are captured and written to SMF. A program (ACSTST) for postprocessing of the SMF data, which NaviQuest uses to generate standard SMS test cases, is also included.

Restriction: Use the SMF data created out of the IGDACSSC exit in selected situations. For example, the extensive testing of system temporary data sets.

Parent topic:[Methods for collecting test data](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

VMA data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

VMA (volume mount analyzer) data to create test cases. VMA data is gathered in the process of analyzing your tape data. This may be done in preparation for either a tape mount management methodology (TMM) implementation or the design of your system-managed tape environment.

Restriction: Use only VMA data to create test cases for tape data sets.

Parent topic:[Methods for collecting test data](#)



©Copyright IBM Corporation 1990, 2014

Data classification methodology

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Data classification is a methodology for SMS implementation that translates high-level requirements into data classes, storage classes, and management classes and identifies the decision points needed in the ACS routines.

Use the data classification technique to build a tree diagram where each major branch (data type) is a collection of data to be migrated to SMS management, each in a separate phase of the implementation process.

[Figure 1](#) shows six data types to be managed by SMS.

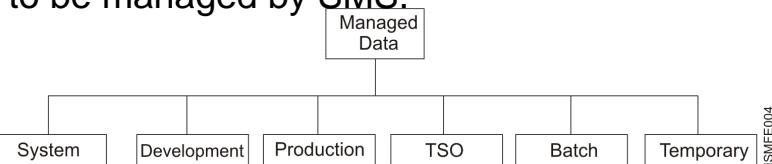
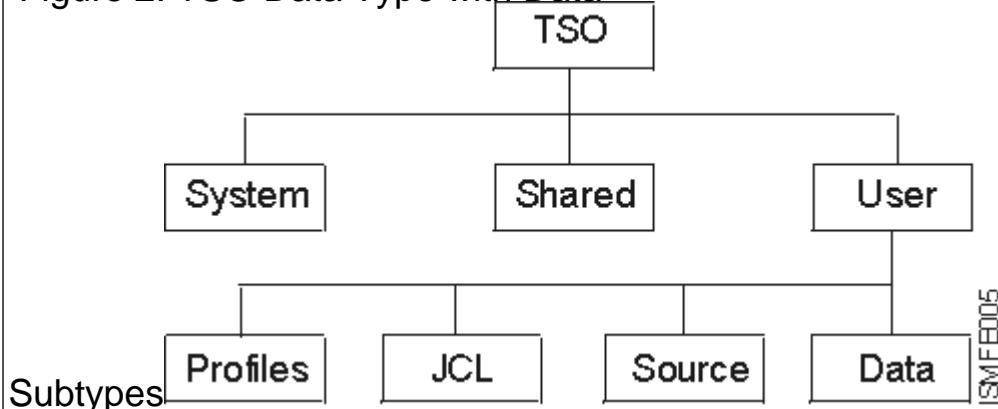


Figure 1. Typical Data Types

After the initial tree diagram has been built, each data type is split further until no more downward splits need to be made. Further splits result in the creation of *data subtypes*. Each data subtype is a distinct set of data sets requiring a unique set of assigned classes. [Figure 2](#) is an example of how a TSO data type can be split into different data subtypes.

Figure 2. TSO Data Type with Data



NaviQuest is designed to support testing of data subtypes determined during data classification. All data sets associated with a single data subtype will have the same expected ACS results. NaviQuest testing at the subtype level lets you manage all test cases for a data subtype and validate the results for that subtype.

Parent topic:[Using NaviQuest](#)

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Testing ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following is an overview of the procedures required to complete the ACS routine testing phase. Apply this process to each new data subtype until all data has been migrated. For the specific procedure, see [Testing procedures](#). After a sample of the data sets that are to be tested has been chosen (or created), a base line test set is generated, to represent the ACS routines *before* they are changed. After the base line test set has been created, the ACS routines are changed (but not activated) to manage the data type. The same set of test cases is then run through ACS testing, generating a new ACS listing. A comparison test is run against the base line test set, and the new listing and exceptions are reported in a comparison report. If the change to the ACS routines was made correctly, the exceptions reported should contain only the new test cases. If no errors occur after converting the data to SMS management, the test cases are updated to reflect the expected results for future regression testing. The current configuration is saved and the new configuration is activated. After the activation, conversion of data to SMS management can proceed.

[Setting up the test environment](#)

[Testing procedures](#)

[NaviQuest testing scenario](#)

Parent topic:[Using NaviQuest](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Setting up the test environment

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Prepare your SMS testing environment and create the test case library. Perform data classification to determine the data types. While this step is optional, if you do not perform data classification for the data types, you must still identify the data subtypes that you want NaviQuest to test.

Assign the subtype prefix. If you performed data classification (step 1), you must assign a 1-to-4 character member name prefix to each data classification data subtype. Even if you do not use data classification, you still must identify the different types of tests you want done and assign them a subtype prefix, because each test runs against a different data subtype and produces a different expected result.

Requirement: For the base line test set, you must use the subtype prefix NEVR.

Determine the amount of testing to be done. You can choose to conduct tests of all data sets in a given data subtype or you can conduct tests with a selected percentage of the data sets. If you use all the data sets, the testing is more extensive, with less chance for error. However, you can use a sample of the data to minimize the resources used while testing.

Determine when to place test cases into the testbed library. The testbed library stores test cases for later use in bulk or regression testing. You can either initially place test cases for all data subtypes into the testbed library or wait to add each subtype test set to the testbed library at the start of each phase. Fully loading the testbed library with all test cases prior to any testing results in more thorough testing. Alternatively, adding the test cases at the start of each phase reduces the total time taken for test runs.

Determine how the data for the test cases is to be created. You must also decide how you want to collect the data set test cases. For each data subtype, decide if you want ISMF lists, DCOLLECT, VMA, or SMF to be the source of the test cases. See [Methods for collecting test data](#) for an explanation of the data collection methods.

The percentage of test data and method of collecting test case data can be tailored for each data subtype. This is independent of your choice of storing all test cases as you initially build the testbed library or at the start of each phase's testing. For example, you may decide to test all production data sets for production data, but to use only a ten percent sample of your test data sets.

Determine the order of testing. Because the order of testing is normally the same as the order of data conversion to SMS management, you must determine the order in which you want to test data subtypes with NaviQuest. Only a single data subtype can be initially tested at one time. Tip: Other data subtypes that have already been tested are regression tested at this time. NaviQuest testing is usually done against all subtypes of one data type before starting the next.

Allocate the testbed library. You can either create a testbed library or let NaviQuest create one for you. The following instructions will guide you in how to create one. To create a testbed library, create a PDS with the following attributes: LRECL = 80BLKSIZE = 0

Each test case is a member of the PDS. All test cases associated with a single data type have the same subtype prefix and have the identical expected results. The amount of space allocated to the library depends on how many data sets the installation has and how many test cases you want to create. For example, on a

3380 device, 1000 test cases take approximately 8 cylinders and 52 directory blocks.

Parent topic: Testing ACS routines



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Testing procedures

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When the set up tasks are complete ([Setting up the test environment](#)), you can begin NaviQuest testing. The initial testing establishes the base line test set against data sets that will never be SMS managed. Because they are not managed, they have an expected result of null ("") for each storage class, storage group, data class, and management class.

Requirement: For these test cases, you must use the subtype prefix NEVR. After the base line test is complete, you can test each phase, or cycle, of the SMS implementation, one subtype at a time. Normally, a SMS implementation phase is made up of either a single data type or several data subtypes. Each data subtype is tested independently. Once each data subtype tests correctly, you can begin data conversion for SMS for that phase.

Recommendation: Put all test cases into the initial base line test set. This will save you from having to repeat adding data types or subtypes one at a time.

Use the following procedure to test your base line test set or any other phase.

Collect data set information for input. With NaviQuest, you can create many test cases at once using input from the following sources: ISMF listsDCOLLECT dataSMF data created by a storage class ACS exitVMA data The data set test cases must all be representative of the data type that you will migrate to SMS management and require the same SMS services.

From the ISMF Primary Option menu, "Data Set" (option 1) offers you two ways to generate data sets samplings: From a saved listingFrom a new listing created from the criteria you specify, such as a VTOC or catalog.

The multivolume variable is always set to "Yes" in an ISMF table if the data set is not open at the time the table is saved. The value is set correctly at the time the data set is opened, which can sometimes cause errors in the bulk test case generator.

Recommendations: Generate SMF test cases from the ACSTST program for temporary data sets, because saving tables of temporary data sets might produce errors in bulk test case creation ("Test Case Generation from Saved ISMF List" option 11.1.1). Set the ACQUIRE DATA FROM VOLUME and ACQUIRE DATA IF DFHSM MIGRATED options under the ISMF "Data Set Selection Entry" panel (ISMF option 1) to Y before generating the list.

After you have created the list, enter the SAVE command on the command line to save the list into a table. For information on the SAVE command, see z/OS DFSMS Using the Interactive Storage Management Facility.

Generate the test cases. Use the Test Case Generation Selection Menu panel to turn the ISMF list, DCOLLECT data, SMS data generated by the storage class ACS exit, and VMA data into standard SMS test cases.

To generate test cases from saved ISMF tables, select option 1, Saved ISMF List, and then use the Test Case Generation from Saved ISMF List Entry Panel to Enter the following information: Saved list name previously saved in step 1 Member name prefix (subtype prefix)PDS that contains the test casesWhether you want to replace the existing test cases with the output test cases Also select additional values that you want included in the test cases.

Recommendation: If you do not enter a PDS name, NaviQuest will generate one based on the format *userid.Tnn.TESTCASE*. Instead, specify a name so that the test case library conforms to your installation's naming standards.

To generate test cases from DCOLLECT data, select option 2, DCOLLECT Data, and then use Test Case Generation from DCOLLECT Data Entry panel.

Enter the following information: Data set nameNumber of test cases you want includedMember name prefix (subtype prefix) of the DCOLLECT test casesPDS that contains the test casesWhether to replace the existing test cases with the output test casesBefore you can use this function, you must have DCOLLECT data

that includes D (data set) records.

To generate test cases from the ACSTST program, select option 3, SMF Data, and then use Test Case Generation from SMF Data Entry panel.

Enter both the data set name containing the system management facility (SMF) data and the name of the test case PDS. **Recommendation:** This function requires that you have the IGDACSSC storage class exit installed and have extracted the SMF type 127 records. The ACSTST program is also required. Both are available in the sample library SYS1.SACBCNTL.

To generate a test case from a VMA extract file, select option 4, VMA Extract Data, and then use the Test Case Generation from VMA Extract Data Entry panel.

Enter the following information: Name of the data set containing the VMA Extract dataNumber of test cases you want generatedMember name prefix (subtype prefix) of the VMA test casesProgram name, if you want to test the implementation for a particular program Also include the name of the test case PDS and whether to replace the existing test cases. **Requirement:** To use this function, you must have already run GFTAXTR from your saved SMF records (types 14, 15, 21 and 30). JCL for GFTAXTR can be found in SYS1.SAMPLIB member GFTAXTRP.

In addition, you can also use the following batch options to create test cases: For option 11.1.1, use the ACBQBAG3 CLIST. See [Generate test cases from ISMF-saved data set lists: ACBQBAG3](#). For option 11.1.2, use the ACBQBAG1 EXEC. See [Generate test cases from DCOLLECT data: ACBQBAG1](#). For option 11.1.3, use the ACBQBAI1 EXEC. See [Generate test cases from SMF data](#). For option 11.1.4, use the ACBQBAO3 EXEC. See [Generate test cases from VMA extract data: ACBQBAO3](#).

Add a 1-to-4 character subtype prefix to each test case member. The prefix must be unique for each data subtype. For example, the first group of TSO data could have subtype prefix TSOA, the second TSOB, and so on.

See step 1 for creating the ISMF table. ACBJBAG2, ACBJBAG1, ACBJBAOW, and ACBJBAI1 in SYS1.SACBCNTL JCL library can perform this task in batch (see [How to run storage administration tasks in batch](#)).

Make ACS routine and construct changes. You must change (but not activate) the ACS code and constructs to reflect the new phase of implementation that you want to test. Before you change the ACS code and construct definitions contained in the source control data set (SCDS), save the old source in case it is needed for recovery.

For information on recovering the ACDS, refer to [Recovering Storage Management Subsystem information](#).

You can now update the ACS routines to reflect the new data subtype you want migrated to SMS.

Update the FILTLISTS. When the ACS code is changed, you might want to use the COPYFILT function of NaviQuest to update all the ACS routines from a common definition of the filter lists. You will be prompted to provide a change log entry that reflects changes you are making to the ACS routines. This entry will be automatically placed into the change log in the ACS routines.

To use the COPYFILT macro, see [COPYFILT macro: COPYLIB facility for FILTLISTS](#).

Translate and validate the ACS routines. You must translate and validate (but not activate) the ACS routines. refids="vs76475 ts64795">The ISMF translate function transforms ACS routines into a table format. Translation checks for syntax errors and transforms the ACS routines into a format suitable for input to the validation. The ISMF validate function verifies that all possible constructs that can be assigned

with the ACS logic have been defined to the SCDS used for testing. ACS routines must be translated before they can be validated; however, validation of ACS routines is optional.

To translate and validate, you can either use the online ISMF functions or you can use the NaviQuest ISMF-in-batch EXEC.

For online translation and validation, choose option 7 (ACS Class Selection) from the ISMF Primary Option menu. To translate, choose option 2 (Translate). To validate, choose option 3 (Validate).

To use the translate facility in batch, see [ACS routine translate: ACBQBAO1](#).

For more information on translation, see [Translating ACS routines](#). For more information on validation, see [Validating ACS routines or an entire SCDS](#).

Run the test cases. Create a **new** ACS listing by using the ISMF Test ACS Routines option (7.4.3). The testbed library contains the test cases. Specify an asterisk (*) to run *a//test* cases in the library.

The **new** ACS listing represents the SMS configuration *after* the ACS routines have been changed for the new data subtype.

Recommendations:Include the prefix of the subtype tested in the ACS listing data set name, to make it easier to identify which data subtype the listing represents.Run test cases in batch whenever possible. For sample JCL for testing ACS routines in batch, see [Test ACS routines: ACBQBAIA](#).

Compare the results of the regression testing. After the base line test, every test includes both testing of new data subtypes and regression testing of previously tested data subtypes, including the base line test set.

At this time, you use the NaviQuest ACS comparison test function to compare the results of all test cases in the testbed library with their expected results. The ACS comparison test produces a report of exceptions. Because you have not yet stored the expected results of the test cases for this data subtype, these test cases appear as exceptions. Later, in step 9, you will store the expected results for the current data subtype test cases. But for now, the exceptions you get are either these valid initial (that is, first run) test cases, or they are errors.

To run the ACS comparison test, choose option 2 from the NaviQuest Primary Option Menu.

On the ACS Comparison Report panel, enter the following information: Name of your base test case resultsName of your new test case resultsPDS that contains the test casesPDS that contains the exception test casesName of the comparison results data set

After running the ACS Test Listings Comparison Entry panel, you must verify the following items: The number of exceptions should be the same as the number of test cases you are currently testing.Exceptions should all have the same subtype prefix.Each listed test case should have the listed results that you expect. If changes have been made correctly to the ACS routines, the differences between the two should be only the data subtype that is being initially tested.

Specify a comparison data set name to be used to store the results of the comparison. Also input whether you want to write over the data set specified if it already exists. If N is specified, and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set. Then press the Enter key.

You will be automatically placed into ISPF "browse" when the comparison completes. The comparison data set you are browsing lists only the test cases identified as exceptions.

If exceptions other than the test cases for the subtype you are initially testing are listed, you have probably made an error in coding the revisions to your ACS routines. Changes in coding that have caused errors must be corrected before you can proceed. This means repeating the operations until the test cases match the exceptions.

The following files are created or updated as output: Exception PDSComparison data set

Important: Each test is an initial test for one data subtype but may include many regression tests for previously tested data subtypes. Expected values are not stored in the initially tested data subtype until its testing completes successfully.

The ACS comparison performed in step 7 has two functions: It validates the regression tests. Current test results of each previously tested data subtype should match the saved expected results previously stored with the test cases. If the results are the same, the regression test is successful. If the results differ, there is an error in the new ACS logic; that is, the ACS routine is assigning different values. It indicates the subtype test set that is being initially tested. Because this is an initial test, this test case has no expected results stored in the test cases, other than null. Thus, during the comparison in step 7, all test cases for this new data subtype show an exception; that is, new results will no longer be null.

For more information about running the ACBQBAC1 EXEC in batch, see [ACS test listings comparison: ACBQBAC1](#).

Validate the test results and determine errors. You must manually compare the new test cases to their expected results for the single data subtype that has been initially tested. This comparison determines if there are initial test errors. If the exceptions contain any test cases from the data subtypes previously tested correctly (in regression testing), these exceptions are also errors.

It is the manual verification of the results that makes sure that the values are the expected results. When all test cases are correct, the test values are stored in the test cases as save expected results, to be used for later regression testing.

If you find errors, you can generate the NaviQuest ACS cross-reference report for additional information about the specific test cases that produced the errors. Use this report to help you debug the ACS logic. If you find errors (from either step 7 or step 8), you must correct the ACS code before returning to step 3 and retest until the data subtype results have no errors.

If you do not find errors, the test is complete, as all the test cases in the subtype test set have the correct expected results.

To create an ACS cross-reference report, choose Enhanced ACS Test Listing Entry panel (option 3) from the NaviQuest Primary Option Menu. On the Enhanced ACS Test Listing Entry panel, fill in the fields with the following names: ISMF test case listing (generated through option 7.4.3)Data set for cross-reference listing (name of data set to contain cross-reference)

Indicate whether the specified data set should be written over if it already exists. If N is specified, and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.

Specify with a Y or an N which variables you want included in your report. Once you have specified all variables that you want, press the Enter key and the report will be produced.

Save the expected results. Once the subtype test is correct, you can use NaviQuest to place the results of the test (that is, the expected results for later regression testing) into the test case definition as the saved expected results for

later regression testing. Test results are only saved after all test cases in the subtype test set have completed with the expected results for that data subtype. The saved expected results will be used for later regression testing, as explained in step 7.

To save the test results, choose Test Case Update With Test Results Entry Panel (option 4) from the ISMF Primary Option menu, which takes you to the Test Case Update With Test Results Entry panel.

Enter the names of the testbed PDS library, the exception test case PDS, the PDS created in the ACS comparison report, and the new ACS test case listing.

The test case members for the exceptions are read and copied into the testbed library. The saved expected results are obtained from the comparison report and are also saved in the testbed library.

You have now completed testing for this data subtype and can now start testing the next data subtype.

Delete the following data sets at the end of this step: Comparison report generated in step 7Exception PDS created in step 7Base and new ACS listing can be deleted (or printed and deleted)

For more information about running the ACBQBAU1 EXEC in batch, see [Update test cases with expected results: ACBQBAU1](#).

Test the next data subtype in the current phase. Continue NaviQuest testing for each data subtype in the current SMS implementation phase. This testing either repeats Steps 1 through 10 or repeats Steps 3 through 10, depending on whether all subtype test sets are initially placed into the testbed.

After the initial test of the base line, all additional tests include regression testing along with initial testing.

Activate your new SMS configuration. Once an entire phase (that is, all the subtypes within the implementation phase) have tested correctly, you can activate the new configuration by using the SETSMS command at an MVS console.

For more information on activating your configuration, see [Activating Storage Management Subsystem configurations](#).

You might want to use the NaviQuest reporting capabilities to determine the amount of DASD space required to convert the data in each phase, prior to attempting conversion. Use this information to ensure that enough DASD is available for the conversion.

Convert data to SMS management. After activation of your new configuration, you can now migrate the data to SMS management. There are several options for doing this data migration: DFSMSdss COPYDUMP/RESTORENormal allocation processingMIGRATE/RECALLCONVERTV

Parent topic: [Testing ACS routines](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

NaviQuest testing scenario

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This customer, initially running with a minimal configuration, plans to convert two data types to SMS management in two phases, one for each data type. The following scenario helps clarify the various testing phases.

[Data testing](#)

[Method of testing](#)

Parent topic:[Testing ACS routines](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data testing

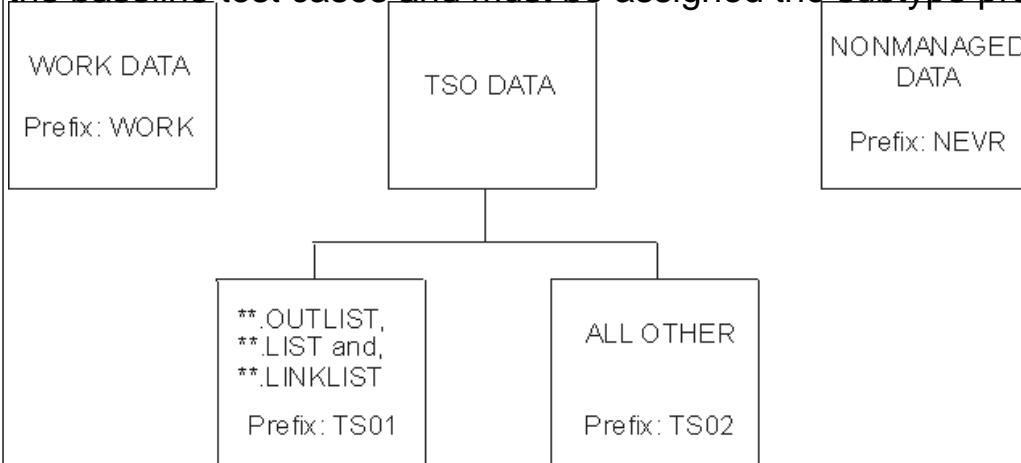
z/OS DFSMSdfp Storage Administration
SC23-6860-01

In this example, the customer wants to test two data types with NaviQuest: work data and TSO data. Because these two data types are the initial data types to be tested, establishing and testing non-SMS data must be done first to create the base line test.

The two data types are shown below with typical data classification assignments. Work data is made up of a single data subtype, each work data set having a single expected result from the ACS routines. TSO data is made up of two data subtypes, each subtype having a different set of expected results from the ACS routines.

The subtype prefix assigned for the work data type is WORK. The subtype prefix for the two TSO data subtypes are TSO1 and TSO2.

There is also a data type for all nonmanaged data. This data type is used to create the baseline test cases and must be assigned the subtype prefix NEVR.



Parent topic:[NaviQuest testing scenario](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Method of testing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Testing with NaviQuest can be done with a sampling of data sets from each data subtype. In the following testing example, however, only two or three data sets are shown.

All test cases are built *before* any NaviQuest testing begins. As NaviQuest testing begins, there are four prefixes in the test bed library: NEVR, WORK, TSO1, and TSO2, with NEVR representing non-SMS-managed (base line) data sets. NaviQuest is used to create the test cases from data set lists for each of the four subtype test sets, in this scenario.

Parent topic:[NaviQuest testing scenario](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Performing storage administration tasks in batch

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Storage administration tasks that are performed using ISMF options can also be done in batch with JCL, CLISTS, and REXX EXECs that are provided by NaviQuest.

For example:

- Testing the SMS configuration
- Performing data set and volume maintenance activities
- Diagnosing data set and volume problems

[How to run storage administration tasks in batch](#)

[Using the sample JCL for batch](#)

[NaviQuest CLISTS and REXX EXECs](#)

[Storage group volume add/Delete: ACBQBAI9](#)

[ACS routine translate: ACBQBAO1](#)

[SCDS validation: ACBQBAO2](#)

[Test ACS routines: ACBQBAIA](#)

[Generate test cases from ISMF-saved data set lists: ACBQBAG3](#)

[Generate test cases from DCOLLECT data: ACBQBAG1](#)

[Generate test cases from SMF data](#)

[Generate test cases from VMA extract data: ACBQBAO3](#)

[ACS test listings comparison: ACBQBAC1](#)

[Enhanced ACS test listing: ACBQBAX1](#)

[Update test cases with expected results: ACBQBAU1](#)

[Generate report from ISMF-saved data set list: ACBQBAR1](#)

[Generate report from ISMF-saved DASD volume list: ACBQVAR1](#)

[Generate report from ISMF-saved tape list: ACBQBAR4](#)

[Generate data set report from DCOLLECT data: ACBQBAR7](#)

[Generate model commands from ISMF-saved list: ACBQBAM1](#)

Parent topic: [Using NaviQuest](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

How to run storage administration tasks in batch

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To run in batch mode the user is required to: Know the specific ISMF option that will be run in batch and the task it performsProvide the parameters that describe each task. These parameters can be modified by the user.Know the required ISPF statements. *Do not modify these ISPF statements.*

[ISMF option 11.7 batch testing and configuration management](#)
Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ISMF option 11.7 batch testing and configuration management

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ISMF Option 11.7, Batch Testing/Configuration Management Selection Menu, lets you select and run batch jobs from an ISMF panel.

Perform the following steps to select and run batch jobs.

Select an option and press Enter. You will be advanced to a menu of batch samples. Or fill in the Data Set to Edit field with the name of the data set containing JCL you want to run. Press Enter. You will be placed in ISPF EDIT mode. Go to step 3.

Select one of the batch sample options or fill in the Data Set to Edit field with the name of the data set containing JCL you want to run. Press Enter. You will be placed in ISPF EDIT mode.

Edit the JCL if needed. While in ISPF EDIT, type SUBMIT at the command line and press ENTER to run the JCL. Refer to [Using the sample JCL for batch](#), step 2, for information on modifying the JCL samples shipped with NaviQuest.

To save the edited JCL, press PF3 or the END command while in ISPF Edit. Fill in the fields and press Enter.

Parent topic:[How to run storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Using the sample JCL for batch

z/OS DFSMSdfp Storage Administration
SC23-6860-01

NaviQuest provides sample JCL in the SYS1.SACBCNTL library. The JCL can be modified with the parameters for the task that is to be performed. [Table 1](#) lists SYS1.SACBCNTL members.

To use the JCL that is provided in SYS1.SACBCNTL, perform the following tasks:
Copy the SYS1.SACBCNTL library member that contains the sample JCL for the task that is to be run in batch. See [Table 1](#) for a list of members and the tasks they perform.

Update the JCL with the appropriate parameters for that task, after DD statement:
`//SYSTSIN DD *`

The parameters and their valid values are on the sample JCL.

Requirements: Each member contains sample syntax and parameters. Change only the job statement and the syntax and parameters. *Do not change any other JCL.ACBJBAOB is called by the other SYS1.SACBCNTL library members during batch processing. Do not modify ACBJBAOB when it is called by the other JCL members of SYS1.SACBCNTL.* [Figure 1](#) shows the JCL in ACBJBAOB.

ISPSTART batch parameters for NaviQuest
SYS1.SACBCNTL sample JCL library
ACBJBAOB JCL

Parent topic: Performing storage administration tasks in batch



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ISPSTART batch parameters for NaviQuest

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following ISPSTART batch parameters have been coded with the appropriate values for NaviQuest. Refer to [z/OS ISPF Dialog Developer's Guide and Reference](#) for additional information about these parameters.

BATSCRDScreen depth
BATSCRWScreen width
BDISPMAXMaximum number of panel displays for a session
BDISPMAX represents the total number of panel display calls. This value is coded to avoid loops.
BREDISPMAXMaximum number of times the same panel can be displayed The job will terminate if this limit is reached. This value is coded to avoid loops.

Parent topic:[Using the sample JCL for batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SYS1.SACBCNTL sample JCL library

z/OS DFSMSdfp Storage Administration
SC23-6860-01

[Table 1](#) lists the sample job library members:

Table 1. SYS1.SACBCNTL Sample JCL Library Member List

Parent topic:[Using the sample JCL for batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ACBJBAOB JCL

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1 shows the JCL for ACBJBAOB.

Figure 1. SYS1.SACBCNTL member ACBJBAOB JCL for Batch Functions Part 1 of 2 //ACBJBAOB PROC CLIST1='SYS1.DGTCLIB',

```

/* ACBJBAOB PROC CLIST1- SIST.DGTLCLIB ,
   CLIST1 SHOULD BE THE FILE #1 FROM THE INSTALLATION TAPE */
/* NOTE THAT THIS IS A FB CLIST LIBRARY; IF YOUR INSTALLATION */
/* USES VB CLIST LIBRARIES, YOU MUST CONVERT THEM YOURSELF */
/*
PLIB1='SYS1.DGTPLIB',
/* PLIB1 SHOULD BE THE FILE #3 FROM THE INSTALLATION TAPE */
/*
LOAD1='SYS1.DGTLLIB',
/* LOAD1 SHOULD BE THE FILE #5 FROM THE INSTALLATION TAPE */
/*
MLIB1='SYS1.DGTMLIB',
/* MLIB1 SHOULD BE THE FILE #6 FROM THE INSTALLATION TAPE */
/*
TABL2='userid.TEST.ISPTABL'
/* TABL2 IS THE DATA SET FOR SAVING ISMF TABLES; YOU SHOULD
/* ALLOCATE THIS DATA SET WITH THE SAME DCB PARAMETERS AS THE
/* ISMF DGTLLIB DATA SET; TABLE CAN BE LARGE - ALLOCATE A LARGE
/* DATA SET. Be sure you change 'userid' to YOUR userid.
/*
*****$MAC(ACBJBAOB) COMP(5695DF123): BATCH - CALLED PROC
/*
/* PROPRIETARY V3 STATEMENT
/* LICENSED MATERIALS - PROPERTY OF IBM
/* *5694-A01
/* //** (C) COPYRIGHT 1998, 2008 IBM CORP.
*/
/* END PROPRIETARY V3 STATEMENT
/*

```

Figure 2. SYS1.SACBCNTL member ACBJBAOB JCL for Batch Functions Part 2 of
2///* CHANGE ACTIVITY: */

```

//CHARGE ACTIVIT
//*$K0=NAVIQUEST,HACS110,95/08/04,SNJADR: INITIAL VERSION @KO4/*
//*$K1=KNQ0003,HACS110,95/08/04,SNJTCS: COMMENT CORRECTION @K1C*/
//*/02/29/96 TCS - HACS110: REMOVED SEQUENCE NUMBERS @WA18945/*
//*$L0=NAVIQUEST,HACS120,96/06/18,SNJTCS: RELEASE 2 CHANGES @LOA/*
//*$L1=KN20035,HACS120,96/07/12,SNJTCS: TYPO CORRECTED @LOA/*
//*/10/14/96 CCY - HDZ11C0: Ship as SPE @WA22861/*
//*/06/13/97 CCY - HDZ11D0: Change IBMUSER to 'userid' @WA27246/*
//*$T1=NQ5,HDZ11E0,061997,SNJTCS: ISP V3R5M0 Datasets deleted @T1A/*
//*/08/27/98 TCS - HDZ11D0: BLKSIZE=0 fix @WA34895/*
//*/11/23/98 TCS - HDZ11E0: Add missing comma @WA36007/*
//*/01/21/99 TCS - HDZ11E0: Changing UNIT from 3380 to SYSDA @WA36897/*
//* ****
//***** ****
//** PROC STEP1 - INVOKES IKJEFT01 *
//***** ****
//STEP1 EXEC PGM=IKJEFT01,DYNAMNBR=50,REGION=(6144K),TIME=(300)
//* -----
//STEPLIB DD DSN=&LOAD1,DISP=SHR
//* -----
//** IN ISPPLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST PANELS */
//* -----
//ISPPLIB DD DSN=&PLIB1,DISP=SHR,BLKSIZE=0
// DD DSN=ISP.SISPPENU,DISP=SHR /* ISPF PANELS */
// DD DSN=SYS1.DGTPLIB,DISP=SHR /* ISMF PANELS */
//* -----
//** IN ISPMLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST MESSAGES */
//* -----
//ISPMILIB DD DSN=&MLIB1,DISP=SHR,BLKSIZE=0
// DD DSN=ISP.SISPMMENU,DISP=SHR /* ISPF MESSAGES */
// DD DSN=SYS1.DGTMILIB,DISP=SHR /* ISMF MESSAGES */
//* -----
//** IN ISPSLIB, INCLUDE THE ISPF, PDF AND ISMF SKELETONS */
//* -----
//ISPSLIB DD DSN=ISP.SISPSEN,DISP=SHR, /* ISPF SKELETONS */
// BLKSIZE=0
// DD DSN=SYS1.DGTSLIB,DISP=SHR /* ISMF SKELETONS */
//* -----
//** IN ISPTLIB, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST TABLES */
//* -----
//ISPTLIB DD DSN=&&TEMP,DISP=NEW,UNIT=SYSDA,BLKSIZE=0,
// SPACE=(TRK,(1,1,1)),DCB=(ISP.SISPTENU)
// DD DSN=&TABL2,DISP=SHR
// DD DSN=ISP.SISPTENU,DISP=SHR /* ISPF TABLES */
// DD DSN=SYS1.DGTTLIB,DISP=SHR /* ISMF TABLES */
//* -----
//** IN SYSPROC, INCLUDE THE ISPF, PDF, ISMF AND NAVIQUEST CLISTS */
//* -----
//SYSPROC DD DSN=&CLIST1,DISP=SHR,BLKSIZE=0
// DD DSN=ISP.SISPCLIB,DISP=SHR /* ISPF CLISTS */
// DD DSN=SYS1.DGTCLIB,DISP=SHR /* ISMF CLISTS */
//* -----
//ISPTABL DD DSN=&TABL2,DISP=SHR,BLKSIZE=0
//SYSPRINT DD SYSOUT=(,)
//SYSUDUMP DD SYSOUT=*
//SYSTSPRT DD SYSOUT=(,),OUTLIM=20000
//ISPLLOG DD SYSOUT=(,),DCB=(LRECL=125,BLKSIZE=129,RECFM=VA)
//ISPPROF DD DSN=&&PROF,DISP=(NEW,DELETE,DELETE),
// DCB=(ISP.SISPTENU),SPACE=(TRK,(1,1,1)),UNIT=SYSDA
// PEND

```

Parent topic:Using the sample JCL for batch

◀ ▶ | ▷ ➔

©Copyright IBM Corporation 1990, 2014

NaviQuest CLISTS and REXX EXECs

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The CLISTS and REXX EXECs that are called by the sample JCL shipped with NaviQuest, can also be called by JCL that *you* code, to perform storage administration tasks. Your JCL must call the appropriate CLIST or REXX EXEC for the task that is to be performed. Use [Table 1](#) to find the CLIST or REXX EXEC you want to use, then update the ISPSTART statement with its name and parameters.

Tip: The naming convention for data set names in the ISPSTART command in the sample jobs is the same as in ISMF. Quoted names are fully qualified and unquoted names will have the PREFIX added as a high-level qualifier.

Refer to the JCL sample jobs in SYS1.SACBCNTL to see the complete syntax for each task.

Recommendations: Do not modify these REXX EXECs and CLISTS. The REXX EXECs and CLISTS do not create a listing; they create an ISMF-saved table (except the test ACS routines task), which is similar to running ISMF interactively and then issuing a SAVE 'xxxxxxxx' command. Save the table, then use ISMF options to produce a flat file of the table for printing: Option 11.7.1 for any tableOption 11.5.n for data set and volume tables.

Table 1. CLISTS and REXX EXECs for Storage Administration Tasks

[Generate a data set list: ACBQBAI2](#)

[Generate a DASD volume list: ACBQBAI4](#)

[Generate a tape volume list: ACBQBAI6](#)

[Define/Alter/Display management class: ACBQBAJ1](#)

Parent topic: [Performing storage administration tasks in batch](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate a data set list: ACBQBAI2

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAI2 is called by the following SYS1.SACBCNTL members to generate the data set list in batch: **ACBJBAI2**Generate data set list and save it in a table**ACBJBAI7**Generate data set list, save it in a table, and save the query**ACBJBAOU**Generate data set list, save it in a table, and generate report See [Figure 1](#) for the sample JCL and parameters.

Use the parameters in [Table 1](#) with ACBQBAI2. At least one OP and one value should be included when specifying a parameter.

Table 1. Using ACBQBAI2 Parameters

[Save or delete the data set list](#)

[Sample JCL for generating a data set list](#)

Parent topic:[NaviQuest CLISTS and REXX EXECs](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Save or delete the data set list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Use the following command to specify whether or not the ISMF table should be saved after it has been generated:

ISPSTART CMD(ACBQBAI2 SAVE|DELETE *tab/nm parameters* ...)

SAVE *tab/nm*Indicates that the generated data set list is to be saved in data set *tab/nm* with the specified parameters.**DELETE**Indicates that the specified data sets in the generated list are to be deleted. **Rule:** Use the DELETE option carefully. Before you use this option, generate the table (with the SAVE option), print it, and then examine the table to see which data sets are to be deleted.

Parent topic:[Generate a data set list: ACBQBAI2](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating a data set list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAI2

```
//*****  
/*  
/* SAMPLE JCL TO CREATE ISMF DATA SET LIST IN BATCH AND SAVE IT  
/*  
/* INSTRUCTIONS BEFORE SUBMITTING:  
/*  
/* CHANGE JOBCARD  
/* CHANGE PREFIX  
/* CHANGE PARAMETERS  
/*  
/* PARAMETERS:  
/*  
/*  
/* 46A/C @WA28670*  
/*  
/* PARAMETER FOLLOWING SAVE - NAME OF THE SAVED LIST (OUTPUT) *  
/* ALLOCSP - 0 to 9999999 (in Kilo Bytes) *  
/* ALLOCUT - Allocation Unit, BLK/TRK/ABS/CYL/MB/KB/BYT @U1A*  
/* BLKSIZE - 0 to 99999 (in Bytes) *  
/* BLKUNUSED - 0 to 999999 (in Kilo Bytes) *  
/* CATNAME - ICF CATALOG name *  
/* CATVOL - VOLUME used in catalog list *  
/* CHGIND - YES/NO *  
/* COMPFMT - YES/NO *  
/* CREATEDT - YYYY/MM/DD (1900/01/01 to 2155/12/31) *  
/* YY/MM/DD ( (19)00/01/01 to (19)99/12/31 ) *  
/* CCSIDDESC - CCSID Description 3@U1A*  
/* 0 to 65534 or A 1 to 17 character string *  
/* that is not a number 0-65534 *  
/* CFCSTNM - CF Cache Structure name @U1A*  
/* CFCTNM - CF Cache Set name @U1A*  
/* CFMOST - CF Monitor status ON/OFF @U1A*  
/* CFSTIND - CF status indicator CE/CQ/VRRQ/VRLS/VQ @U1A*  
/*  
/* DATACLS - DATA CLASS name *  
/* DDMATTR - DDM Attributes YES/NO @U1A*  
/* DEVTYPE - 3380/3390/9345 for DASD types *  
/* 3480/3480X/3490/3590-1/3592 for TAPE types *  
/* DSENV - Dataset Environment MANAGED/UNMANAGED @U1A*  
/* DSNTYP - DS name type EXTENDED/HFS/LIBRARY/OTHERS @U1A*  
/* DSN - DATA SET NAME *  
/* ('**' requires catalog name) *  
/* DSORG - DA/DAU/IS/ISU/PO/POU/PS/PSU/UN/VS *  
/* ENTRYTYP - AIX/ALIAS/CATALOG/DATA/INDEX/NONVSAM/PATH/ *  
/* DEFERRED/GDG/GDS/ROLLOFF *  
/* EXPIREDT - YYYY/MM/DD (1900/01/01 to 2155/12/31) or *  
/* - YY/MM/DD ( (19)00/01/01 TO (19)99/12/31 ) or *  
/* - NEVER or 1999/00/00 *  
/* HSMDATA - Y / N (Catalog List) *  
/* LASTBKUP - YYYY/MM/DD (1900/01/01 to 2155/12/31) *  
/* YY/MM/DD ( (19)00/01/01 to (19)99/12/31 ) *  
/* LASTREF - YYYY/MM/DD (1900/01/01 to 2155/12/31) *  
/* YY/MM/DD ( (19)00/01/01 to (19)99/12/31 ) *  
/* LISTTYP - Inclusive/Exclusive or I/E 3@U1A*  
/* Inclusive: Display list by Inclusive criteria *  
/* Exclusive: Display list by Exclusive criteria *  
/*
```

Figure 2. Sample JCL for ACBJBAI2 (continued)

```
/*  
/* MGMTCLS - MANAGEMENT CLASS name *  
/* MULTVOL - YES/NO *  
/* NOTUSED% - 0 to 100 *  
/* NUMEXT - 0 to 999 *  
/* NUMSTRIPE - 1 to 16 *  
/* OPTIMAL - 0 to 99999 (in Bytes) *  
/* OWNER - OWNER of the datasets *  
/* QSAVE - QUERY Name to be saved *  
/* QUERY - QUERY Name to be used *  
/* REBLOCK - YES/NO *  
/* RECFMT - A/B/D/F/M/FS/VS/T/U/V *  
/* SECALLOC - 0 to 99999999 (in Kilo Bytes) *  
/* SOURCEGL - 1 / 2 (2-new list) *  
/* SOURCENL - 1 / 2 (1-VTOC 2-catalog) *  
/* STORCLS - STORAGE CLASS name *  
/* STORGRP - STORAGE GROUP name *  
/* USEDSPC - 0 to 9999999 (in Kilo Bytes) *  
/* USERDATAREDUCT% - 0 to 99 *  
/* VTOCDATA - Y / N (Catalog List) *  
/* VTOCVSER - VTOC VOLUME SERIAL NUMBERS (VTOC list) *  
/*  
/******  
//SAVELIST EXEC ACBJBAOB,  
// PLIB1=SYS1.DGTPLIB,  
// TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(ACBQBAI2 SAVE DSNLIST +  
VTOCDATA(Y) HSMDATA(Y) +  
ALLOCSP(GT 500) DSORG(NE PS) +  
DSN('IBMUSER.*')) +  
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
/******
```

Parent topic:Generate a data set list: ACBQBAI2

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate a DASD volume list: ACBQBAI4

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAI4 is called by the following SYS1.SACBCNTL members to generate the list of DASD volumes in batch: **ACBJBAI4**Generate a DASD volume list, save it in a table, and save the query**ACBJBAI5**Generate a DASD volume list and save it in a table**ACBJBAI8**Generate a DASD volume list, save it in a table, and generate a report from it

For the sample JCL and parameters, see [Figure 1](#).

For ACBQBAI4, use the parameters listed in [Table 1](#). At least one OP and one value should be included when specifying a parameter.

Table 1. ACBQBAI4 Parameters

[Sample JCL for generating a volume list](#)

Parent topic:[NaviQuest CLISTS and REXX EXECs](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating a volume list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAI4 Part 1 of

```

3//*****                                                 *
//** SAMPLE JCL TO GENERATE ISMF DASD VOLUME LIST IN BATCH, SAVE IT,   *
//** AND SAVE THE QUERY ALSO                                         *
//**                                                               *
//** INSTRUCTIONS BEFORE SUBMITTING:                                *
//**                                                               *
//**     CHANGE JOBCARD                                         *
//**     CHANGE PREFIX                                           *
//**     CHANGE PARAMETERS                                     *
//**                                                               *
//** PARAMETERS:                                              36A/C @WA*
//**                                                               *
//** PARAMETER FOLLOWING SAVE - NAME OF SAVED LIST (OUTPUT)      *
//** ALLOCSP    - 0 to 99999999 followed by K (kilobytes) or       *
//**               M (megabytes). If K or M aren't specified        *
//**               the value will be defaulted to K.                 *
//** ALSPTRK   - 0 to 99999999 followed by K (kilobytes) or       *
//**               the value will be defaulted to K.                 * //*
//** CDSNAME   - Control Dataset Name                           *
//** CFVOLST   - CF Volume status ENABLED/QUIESCING/QUIESCED   @U1A*
//** CFWSTAT   - ACTIVE/INACTIVE/NONE/PENDING/PINNED           *
//** DEVTYPE   - 3380/3390/9345                                 *
//**             - Installation defined esoteric names(like SYSDA ..) *
//** DFWSTAT   - ACTIVE/INACTIVE/NONE/PENDING/PINNED           *
//** DUPLXSTAT - NONE / PPRI-FAI / PPRI-PEN / PPRI-SUS / PPRIMARY/ *
//**               PRI-PEN / PRI-SUS / PRIMARY / PSEC-FAI / PSEC-PEN/ *
//**               PSEC-SUS / PSECNDRY/ SEC-PEN / SEC-SUS / SECONDRY/ *
//**               SIMPLEX / SPAR-BRK / SPAR-PEN / SPARE          *
//** FRAG      - 1 to 999                                         *
//** FREEDSCB  - 0 to 999999                                     *

```

Figure 2. Sample JCL for ACBJBAI4 Part 2 of 3

```

(*kilobytes) or
/*
   M (megabytes). If K or M aren't specified      *
   the value will be defaulted to K.                *
/** FRSPTRK - 0 to 99999999 followed by K (kilobytes) or      *
/** M (megabytes). If K or M aren't specified      *
/** the value will be defaulted to K.                *
/** FREESPC% - 0 to 100 (with no % sign)           *
/** FREEVIR - 0 to 99999               *
/** FREEXT - 0 to 9999999               *
/** FROMDEV - 1 to 4 HexaDecimal digits          *
/** INDEX - DISABLED/ENABLED/NONE                 *
/** INIASRES - Initialized as Reserved (YES / NO)    *
/** LASTDEV - 1 to 4 HexaDecimal digits          *
/** LISTTYP - Inclusive/Exclusive or I/E          *
/**          Inclusive: Display list by Inclusive criteria *
/**          Exclusive: Display list by Exclusive criteria *
/** LRGEEXT - 1 to 99999999 followed by K (kilobytes) or      *
/** M (megabytes). If K or M aren't specified      *
/** the value will be defaulted to K.                *
/** LGEXTRK - 1 to 99999999 followed by K (kilobytes) or      *
/** M (megabytes). If K or M aren't specified      *
/** the value will be defaulted to K.                *
/** OTHERDEV - 1 to 4 Hex.Decimal digits          *
/** OWNERID - Owner ID, 1 to 14 characters        *
/** PHYDATA - Y / N                                *

```

Figure 3. Sample JCL for ACBJBAI4 Part 3 of 3

```

//** QSAVE      - Query Name to be saved          *
//** QUERY      - Query Name to be used           *
//** RDCACHE    - ACTIVE/INACTIVE/NONE/PENDING    *
//** SHRDASD   - YES / NO                         *
//** SOURCEGL  - 1 / 2 (2-New List)               *
//** SOURCENL  - 1 / 2 (1-Physical, 2-SMS)        *
//** SPCDATA    - Y / N                           *
//** STORGRP    - Storage Group name (SMS only)   *
//** SUBSYSID  - 0001 to FFFF (in HexaDecimal)     *
//** USEATTR    - PRIV/PUB/STOR                   *
//** VOL        - Volume serial                  *
//** VOLSTYPE   - 1 / 2 / 3 (1-On Line 2-Not On Line 3-Either)
//**                                         *
//*****                                                 *
//DASDLST EXEC ACJBBAOB,
//          PLIB1=SYSL.DGTPLIB,
//          TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER) MSGID
ISPSTART CMD(ACBQBA14 +
SAVE CHKFrag QSAVE(CHKFrag) +
SPCDATA(Y) PHYDATA(Y) +
VOL(TSC*) FRAG(GT 450)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****

```

Parent topic: Generate a DASD volume list: ACBQBAI4

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate a tape volume list: ACBQBAI6

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAI6 is called by the following SYS1.SACBCNTL members to list the mountable tape volumes in batch: **ACBJBAID**Creates ISMF table of scratch tapes in a library, and then prints the table.**ACBJBAIA**Creates ISMF table of all tapes in a library.

For the sample JCL and parameters for ACBJBAIA, see [Figure 1](#).

For ACBQBAID, use the parameters listed in [Table 1](#).

Table 1. ACBQBAID Parameters

[Sample JCL for generating a tape list](#)

Parent topic:[NaviQuest CLISTS and REXX EXECs](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating a tape list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAIA

```
/*
/* SAMPLE JCL TO GENERATE AN ISMF MOUNTABLE TAPE VOLUME LIST,
/* SAVE IT, AND GENERATE A TAPE VOLUME REPORT FROM IT
*/
/* INSTRUCTIONS BEFORE SUBMITTING:
*/
/* CHANGE JOBCARD
/* CHANGE PROFILE PREFIX COMMAND
*/
/*
/* STEP TO GENERATE TAPE VOLUME LIST AND SAVE IT
*/
/* VOL - TAPE VOLUMES TO BE SELECTED
/* LIBNAME - LIBRARY NAMES TO BE SELECTED
/* STORGRP - STORAGE GROUPS TO BE SELECTED
/* PARAMETER FOLLOWING SAVE - NAME FOR THE SAVED LIST (OUTPUT)
*/
/*
//GENLIST EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
DELETE TAPEVOL.REPORT
ISPSTART CMD(ACBQBAI6 SAVE TAPELIST +
VOL(*) LIBNAME(L*) STORGRP(*)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(99999999)
/*
/*
/* STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED
/* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY
/* DATA SET.
*/
/*
//ALCISPFL EXEC PGM=IEFBR14
//ISPFILE DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=(NEW,CATLG),
// BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
/*
/*
/* STEP TO GENERATE THE TAPE VOLUME REPORT FROM THE SAVED LIST
*/
/* PARAMETER FOR ACBQBAR4 - SAVED LIST NAME (FROM GENLIST STEP)
/* ISPFILE - TAPE VOLUME REPORT (OUTPUT, FROM ALCISPFL STEP)
/* SYSIN - KEY WORDS SPECIFYING COLUMNS TO BE INCLUDED IN THE
/* GENERATED REPORT
*/
/*
//GENREP EXEC
ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR4 TAPELIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(99999999)
/*
//ISPFILE DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=OLD
//SYSIN DD *
VOLSER
USEATTR
VOLERROR
CKPT
LIBNAME
STORGRP
MEDIA
RECTECH
COMPTYPE
LASTWRITE
LASTMOUNT
LASTEJECT
EXPDTVOL
CRTDTVOL
VOLLOC
TITLE=STATUS OF TAPES AS ON 06/01/96
/*
/*
/* COPY THE REPORT
*/
/* SYSUT1 - INPUT (FROM PREVIOUS STEP)
/* SYSUT2 - OUTPUT
*/
/*
//TAPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
/*

```

Figure 2. Sample JCL for ACBJBAIA part 2 of 2

```
//GENREP EXEC
ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR4 TAPELIST) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(99999999)
/*
//ISPFILE DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=OLD
//SYSIN DD *
VOLSER
USEATTR
VOLERROR
CKPT
LIBNAME
STORGRP
MEDIA
RECTECH
COMPTYPE
LASTWRITE
LASTMOUNT
LASTEJECT
EXPDTVOL
CRTDTVOL
VOLLOC
TITLE=STATUS OF TAPES AS ON 06/01/96
/*
/*
/* COPY THE REPORT
*/
/* SYSUT1 - INPUT (FROM PREVIOUS STEP)
/* SYSUT2 - OUTPUT
*/
/*
//TAPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
/*

```

Parent topic:Generate a tape volume list: ACBQBAI6

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Define/Alter/Display management class: ACBQBAJ1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SYS1.SACBCNTL member ACBJBAJ1 calls ACBQBAJ1 to define, alter, or display data set management class in batch.

See [Figure 1](#) for the sample JCL and parameters.

[Sample JCL for define/Alter/Display management class](#)
Parent topic:[NaviQuest CLISTS and REXX EXECs](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for define/Alter/Display management class

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAJ1

```
/*
/* SAMPLE JCL TO DEFINE/ALTER/DISPLAY MANAGEMENT CLASSES IN BATCH *
/*
/* INSTRUCTIONS BEFORE SUBMITTING:
/*
/* CHANGE JOBCARD *
/* CHANGE PREFIX *
/* CHANGE PARAMETERS *
/*
/* PARAMETER FOLLOWING ACBQAJ1 - DEFINE or ALTER or DISPLAY *
/* ***** ADD BEG ***** *
/* (Define or Alter) *
/*
/* Required Fields: 2@WA49380 *
/*
/* SCDS - SCDS in which MANAGEMENT CLASS is to be DEF/ALT/DISP *
/*
/* MGMTCLAS - MANAGEMENT CLASS to be DEFINED/ALTERED/DISPLAYED *
/*
/* Optional Fields: 2@WA49380 *
/*
/* DESCRL : Type in remarks about the MGMTCLAS which is being *
/* defined/alterred, not exceeding 120 chars. *
/*
/* EXPNOUSE : The datasets will expire if they are not used for * /**
* /* @D1C * /* Possible values 1 - 93000, NOLIMIT. If
NOLIMIT is * /* specified the DS would not expire. * /* Valid only if retention
period or expiration date * /* is not specified by the end user or is not derived * /* from the data
class. * /* * /* EXPDTDY : Datasets expires after DATE/DAYS entered here. @D1C * /* Possible values 0 - 93000, YYYY/MM/DD or NOLIMIT. * /*
* /* RETNLIM : Possible values 0 - 93000, NOLIMIT. @D1C * /* Use this field to control what a user or
Data class * /* can specify for retention period or expiration date * /* during allocation. The affect
of the values entered * /* in this field are explained below. * /* the user or Dataclass
specified. * /* 0 -> Do not use the RETPD and EXPDT that * /* EXPDT is
more than this limit. * /* 1 - 93000 -> Use this value only if the RETPD or * /* NOLIMIT -> Do not set a limit to RETPD or EXPDT. * /*
* /* PARTREL : Possible values Y, C, YI, CI or
N . *
/*
/* Use this field (PARTIAL RELEASE) to specify whether * *
/* allocated but unused space can be released for DSS * *
/* in this MGMTCLS. This one applies only to VSAM DSS * *
/* in extended format or NON-VSAM datasets. The values * *
/* entered would have following results. *
/*
/* Y -> Release unused space automatically during the * *
/* Space Management cycle. *
/*
/* C -> Unused space can be released automatically * *
/* only if a secondary allocation exists for the * *
/* dataset. *
/*
/* YI -> Release unused space when a dataset is closed * *
/* or during the Space Management cycle,whichever * *
/* comes first. *
/*
/* CI -> Unused space for data sets with secondary * *
/* allocation is released either when a data set * *
/* is closed or during the Space Management * *
/* cycle, whichever comes first. *
/*
/* N -> Do not release unused space. *
/*
/* PRINOUSE : Use this field to specify when to migrate the DSS * *
/* in this class. The possible values are *
/*
/* 0 -> To Migrate data sets as soon as the * *
/* space management function of DFMSHSM * *
/* is run and data integrity age is met. *
/*
/* 1 to 9999 -> Migrate data sets out of primary * *
/* storage if they have been unused for * *
/* this number of days or longer. *
/*
/* BLANK -> *
/*
/* LVINOUSE : Use this filed to specify whether DSS can migrate * *
/* to LEVEL 1 storage and how long they can remain * *
/* there. The possible values are, *
/*
/* 0 -> No migration to Level 1. DSS migrate * *
/* directly from primary storage to LVL 2 *
/*
/* 1 to 9999 -> The total number of consecutive days * *
/* that datasets must remain unaccessed * *
/* before becoming eligible to migrate * *
/* from LVL 1 to LVL 2. *
/*
/* NOLIMIT -> Datasets can not migrate to LEVEL 2 * *
/* automatically, and remain in LVL 1 for * *
/* an unlimited period. *
/*
/* BLANK -> *
/*
/* CMDRAUT : If migration is allowed, this field determines how * *
/* the migration is initiated. Possible values are, *
/*
/* BOTH -> DSS can migrate either automatically * *
/* or by command. *
/*
/* COMMAND -> Data sets can migrate by command only. *
/*
/*
```

Figure 2. Sample JCL for ACBJBAJ1 (continued)

```
/*
/* PARTREL : Possible values Y, C, YI, CI or
N . *
/*
/* Use this field (PARTIAL RELEASE) to specify whether * *
/* allocated but unused space can be released for DSS * *
/* in this MGMTCLS. This one applies only to VSAM DSS * *
/* in extended format or NON-VSAM datasets. The values * *
/* entered would have following results. *
/*
/* Y -> Release unused space automatically during the * *
/* Space Management cycle. *
/*
/* C -> Unused space can be released automatically * *
/* only if a secondary allocation exists for the * *
/* dataset. *
/*
/* YI -> Release unused space when a dataset is closed * *
/* or during the Space Management cycle,whichever * *
/* comes first. *
/*
/* CI -> Unused space for data sets with secondary * *
/* allocation is released either when a data set * *
/* is closed or during the Space Management * *
/* cycle, whichever comes first. *
/*
/* N -> Do not release unused space. *
/*
/* PRINOUSE : Use this field to specify when to migrate the DSS * *
/* in this class. The possible values are *
/*
/* 0 -> To Migrate data sets as soon as the * *
/* space management function of DFMSHSM * *
/* is run and data integrity age is met. *
/*
/* 1 to 9999 -> Migrate data sets out of primary * *
/* storage if they have been unused for * *
/* this number of days or longer. *
/*
/* BLANK -> *
/*
/* LVINOUSE : Use this filed to specify whether DSS can migrate * *
/* to LEVEL 1 storage and how long they can remain * *
/* there. The possible values are, *
/*
/* 0 -> No migration to Level 1. DSS migrate * *
/* directly from primary storage to LVL 2 *
/*
/* 1 to 9999 -> The total number of consecutive days * *
/* that datasets must remain unaccessed * *
/* before becoming eligible to migrate * *
/* from LVL 1 to LVL 2. *
/*
/* NOLIMIT -> Datasets can not migrate to LEVEL 2 * *
/* automatically, and remain in LVL 1 for * *
/* an unlimited period. *
/*
/* BLANK -> *
/*
/* CMDRAUT : If migration is allowed, this field determines how * *
/* the migration is initiated. Possible values are, *
/*
/* BOTH -> DSS can migrate either automatically * *
/* or by command. *
/*
/* COMMAND -> Data sets can migrate by command only. *
/*
/*
```

```

/*
NONE      -> Data sets cannot migrate.
*/
/* Group (GDG) DSS only. */
/* This field specifies how many of the newest *
/* generations of a GDG are to have normal priority. *
/* Possible values are 0 - 255 or blank. For Example *
/* enter 100 if you want GDG generations older *
/* than the most recent 100, to migrate before non *
/* generation datasets.
*/
/* GDROLL : This field specifies whether the Generation DSS *
/* in this MGMTCLS will expire or migrate after they *
/* have been removed from the GDG. The possible *
/* values are, MIGRATE,EXPIRE or blank.
*/
/* BACKUPFR : This field specifies the backup frequency. The *
/* possible values are,
*/
/* 0      -> Backup each dataset only when the volume*
/* it resides on is backed up.
*/
/* 1 - 9999 -> If dataset is changed in the interval *
/* between backups, extend the interval *
/* for atleast this many number of days.
*/
/* BLANK    ->
*/
/* NUMBKDSE : Maximum number of Backups that can be kept *
/* concurrently. Possible values are, 1 - 100,BLANK.
*/
/* NUMBKDSD : Specifies the maximum no of Backups to keep after *
/* the dataset is deleted. Possible values are
*/
/* 0      -> All backups that were created are erased *
/* after the dataset is deleted.
*/
/* 1 - 100 -> The maximum no. of backups to keep after *
/* a dataset has been deleted.
*/
/* BLANK    ->
*/
/* RETDYDSD : Specifies how long a most recent backup version of *
/* a deleted dataset will be kept. Possible values are
*/
/* 1 - 9999 -> After a dataset is deleted keep its most*
/* recent backup version for these many *
/* days.
*/
/* NOLIMIT  -> The backup version will be kept for *
/* unlimited period.
*/
/* BLANK    ->
*/
/* RETDYEXT : Specifies the retention period for a dataset that *
/* pre-date the most recent backup. Possible values *
/* are,
*/
/* 1 - 9999 -> Each backup version of a dataset other *
/* than the recent copy will be kept for *
/* these many days.
*/
/* NOLIMIT  -> All backup versions will be kept for *
/* unlimited period.
*/
/* BLANK    ->
*/

```

Figure 4. Sample JCL for ACBJBAJ1 (continued)

```

//*          backups. Possible values are,           *
//*          *                                     *
//*          ADMIN   -> Only Storage Administrator ,      *
//*          *                                     *
//*          BOTH    -> Both Storage Administrator and end users. *
//*          *                                     *
//*          NONE    -> Neither end user nor Storage Administrator*
//*          *                                     *
//*          AUTOBKUP : Specifies whether the datasets in this MGMTCLS are *
//*          eligible for automatic backup. Possible values are *
//*          *                                     *
//*          Y -> Yes , N -> No
//*          *                                     *
//*          BKUPTECH : Specifies BACKUP COPY TECHNIQUE to be used.
//*          Possible values are,
//*          *                                     *
//*          R -> Concurrent copy technique must be used.      * //*
used.      * //** S -> Without the concurrent copy technique.      * //*
technique 9@AOA      * //** must be used.      * //** P -> Concurrent copy technique should be
concurrent copy technique      * //**      should be used.      * //** VR -> Virtual concurrent copy
Cache-based concurrent copy technique      * //**      must be used.      * //** VP -> Virtual
CP -> Cache-based concurrent copy technique      * //**      should be used.      * //** CR ->
//*
//*          TMSCYRS : No of years that must pass since the creation date *
//*          before class transition occurs. Possible values are *
//*          *                                     *
//*          0 - 9999, or BLANK.
//*
//*          TMSCMTH : No of months that must pass since the creation date *
//*          before class transition occurs. Possible values are *
//*          *                                     *
//*          0 - 9999, or BLANK.
//*
//*          TMSCDYS : No of days that must pass since the creation date *
//*          before class transition occurs. Possible values are *
//*          *                                     *
//*          0 - 9999, or BLANK.
//*

```

```

/* TMSLUYRS : No of years that must pass since the last reference *
/* date before class transition occurs.Possible values *
/* are 0 - 9999, or BLANK.
*/
/*
/* TMSLUMTH : No of months that must pass since the last reference*
/* date before class transition occurs.Possible values *
/* are 0 - 9999, or BLANK.
*/
/*
/* TMSLUDYS : No of days that must pass since the last reference *
/* date before class transition occurs.Possible values *
/* are 0 - 9999, or BLANK.
*/
/*
/* PMTHODAY : The day of the month that class transition occurs. *
/* Possible values, 1 - 31, FIRST, LAST or BLANK
*/
/*
/* PQUAODAY : The day of the each quarter the CT occurs. *
/* Possible values, 1 - 92, FIRST, LAST or BLANK
*/
/*
/* PQUAIMTH : Month of each quarter the CT occurs. *
/* Possible values, 1 - 3, or BLANK
*/
/*
/* PYRLODAY : The day of each year the CT occurs. *
/* Possible values, 1 - 366,FIRST,LAST or BLANK
*/

```

Figure 5. Sample JCL for ACBJBAJ1 (continued)

```

/*
PYRLIMTH : The month of each year the CT occurs. *
/* Possible values, 1 - 12, or BLANK
*/
/*
VERSIONS : Specify how many versions of an aggregate group *
/* associated with the management class are to be *
/* maintained. Possible values are 1 - 9999, NOLIMIT or *
/* BLANK. If BLANK is specified no aggregate group BKP *
/* is maintained.
*/
/*
RTNOVERS : Specify how long the only version of an aggregate *
/* group is kept. Possible values are 1 - 9999,NOLIMIT *
/* or BLANK.
*/
/*
RTOVUNIT : Specify the unit of measure for the length of time*
/* specified in the above field. Possible values are *
/* D -> Days, W -> Weeks, M -> Months, Y -> Years and *
/* BLANK.
*/
/*
RTNEVERS : Specify the time periods for which backup versions*
/* of an aggregate group are to be kept.Possible values*
/* are 1 - 9999,NOLIMIT and BLANK.
*/
/*
RTEVUNIT : Specify the unit of measure for the length of time *
/* specified in the above field. Possible values are *
/* D -> Days, W -> Weeks, M -> Months, Y -> Years and *
/* BLANK.
*/
/*
CPYSERLN : Specifies whether you want processing of a backup *
/* copy of an aggregate group to continue if a shared *
/* enqueue cannot be obtained for the datasets being *
/* backed up. Possible values are,
/* C -> Continue, F -> Fail or BLANK.
*/
/*
ACPYTECH : Specifies ABACKUP COPY TECHNIQUE to be used. *
/* Possible values are,
/*
      R -> Concurrent copy technique must be used. * /**
      used.      * /**
      S -> Without the concurrent copy technique. * /**
      technique 9@AOA      * /**
      must be used.      * /**
      concurrent copy technique      * /**
      Cache-based concurrent copy technique      * /**
      CP -> Cache-based concurrent copy technique      * /**
      should be used.      * /**
      must be used.      * /**
      should be used.      * /**
      CR ->
/*
      TRCPYTECH : Specifies which copy technique should be used 9@E1A* /**
      associated with      * /**
      this management class. Possible values are      * /**
      FRR -> FR REQUIRED.      * /**
      PMP -> FC PRESMIRPREF.      * /**
      for the class transition of data
      invokes an user exit, or      * /**
      there is a serialization error. Possible values are * /**
      NONE.      * /**
      * /**
      UPDHLVLSCDS: When modifying an SCDS, that was formatted with a *
      higher level of SMS, using a lower level of SMS *
      will make this application fail unless you *
      specify the UPDHLVLSCDS parameter as 'Y'.
      Default is 'N'.      @WA41441*
/*
      Possible values : Y/N/BLANK      @WA41441*
***** ADD END *****

```

Figure 6. Sample JCL for ACBJBAJ1 (continued)

```

/*
/* STEP1 - SET UP PARAMETERS
/*
***** STEP1 - EXEC ACBJBAOB,
// TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//TEMPFILE DD DSN=&TEMPFILE,DISP=(NEW,PASS),
// SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJ1 DEFINE/ALTER +
SCDS(TEMP.SCDS) +
MGMTCLAS() +
DESCR() +

```

```

DESCR() +
EXPNOUSE() +
EXPDTDY() +RETNLIM() +
PARTREL() +
PRINOUSE() +
LVINOUSE() +
CMDORAUT() +
PRIGDTEL() +
GDGROLL() +
BACKUPFR() +
NUMBKDSE() +
NUMBKDS() +
RETDYDSD() +
RETDYEXT() +
CMDBKUP() +
AUTOBKUP() +
BKUPTECH() +
TMSCYRS() +
TMSCMTH() +
TMSCDYS() +
TMSLUYRS() +
TMSLUMTH() +
TMSLUDYS() +
PMTHDAY() +
PQUAOADAY() +
PQUAIMTH() +
PYRLDAY() +
PYRLIMTH() +
VERSIONS() +
RTNOVERS() +
RTOVUNIT() +
RTNEVERS() +
RTEVUNIT() +
CPYSERLN() +
ACPYTECH() +
TRCPYTECH() + SERIALERREX() + <|
UPDHVLSCDS() +
)

```

Figure 7. Sample JCL for ACBJBAJ1 (continued)

```

/*
//STEP2 - EXECUTE THE DEFINE/ALTER
/*
//***** STEP2 - EXEC ACBJBAOB
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD DSN=&TEMPFILE,DISP=(OLD,DELETE,DELETE)
//***** STEP3 - SET UP PARAMETERS
/*
//***** STEP3 - EXEC ACBJBAOB,
//          TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//TEMPFILE DD DSN=&TEMPFILE,DISP=(NEW,PASS),
//          SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAJ1 DISPLAY +
SCDS(TEMP.SCDS) +
MGMTCLAS() +
*/

```

Parent topic: Define/Alter/Display management class: ACBQBAJ1



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Storage group volume add/Delete: ACBQBAI9

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SYS1.SACBCNTL member ACBJBAIB calls ACBQBAI9 to add SMS volumes from a storage group. When performing an ADD or a DELETE on an SMS volume in an SMS complex, run ACBQBAI9 for each MVS image in the SMS complex.

See [Figure 1](#) for the sample JCL and parameters.

Use one of the following DDNAMES to indicate which operation is to occur:

VOLADD lists the volumes to add to a storage group in an SCDS
VOLALT lists the volumes to be altered in a storage group in an SCDS
VOLDEL lists the volumes to delete from a storage group in an SCDS

[Sample JCL for storage group volume add/Delete](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for storage group volume add/Delete

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To delete a volume, specify the following: SCDSNAMESGVOLVOLDEL for the DDNAME

Figure 1. Sample JCL for
ACBJBAIB //*****

```
/*  
/* SAMPLE JCL TO ADD NEW VOLUMES AND THEIR STATUS  
/*  
/* INSTRUCTIONS BEFORE SUBMITTING:  
/*  
/* CHANGE JOBCARD  
/* CHANGE PREFIX  
/* CHANGE PARAMETERS  
/*  
/* TEMPFILE - LISTING DATA SET (OUTPUT)  
/* VOLADD - VOLUMES TO BE ADDED (INPUT)  
/*  
/* PARAMETERS:  
/*  
/* Required Fields: 2@WA49380  
/*  
/* SCDSNAME - SOURCE CONTROL DATA SET *  
/* VOL - VOLUME *  
/* SG - STORAGE GROUP *  
/* 3@WA49380  
/* Optional Fields:  
/*  
/* STATUS - STATUS (ENABLE/NOTCON/DISALL/DISNEW/QUIALL/  
/* QUINEW) *  
/* Up to 32 statuses can be specified separated by *  
/* commas to match the 32 systems. If a status is *  
/* skipped, the system status that falls in between 2 *  
/* commas will have default value of ENABLE. @WA29937*  
/* 13A@WA53756*  
/* STATUSALL - STATUSALL (ENABLE/NOTCON/DISALL/DISNEW/QUIALL/  
/* QUINEW) *  
/* If Volume status in all the Systems needs to be *  
/* set to a single value (for example ENABLE), *  
/* STATUSALL is an easier option compared to the *  
/* parameter STATUS. *  
/* Note: STATUSALL and STATUS are mutually exclusive. *  
/* And so, while specifying value for one of these *  
/* parameters, either the other parameter should *  
/* not be specified or if specified, it should not *  
/* have any value specified. *  
/*  
/* UPDHLVLSCDS - When modifying an SCDS, that was formatted *  
/* with a higher level of SMS, using a lower *  
/* level of SMS will make this application fail *  
/* unless you specify the UPDHLVLSCDS parameter *  
/* as 'Y'. Default is 'N'. @WA41441*  
/*  
/* If specified, this should be the first *  
/* parameter on either VOLDEL or ADDVOL DD *  
/* names. @WA41441*  
/*  
/* Possible values : Y/N/BLANK @WA41441*
```

Figure 2. Sample JCL for ACBJBAIB (continued) //*

```
/* To DELETE volume, specify SCDSNAME, VOL and SG, *  
/* and use VOLDEL for DDname. @WA29937*  
/*  
/* To ALTER Volume, specify all the parameters as in VOLUME 11@U1A*  
/* ADD and use VOLALT as DDname.  
/*  
/* Specify '+' as the continuation character. When splitting a *  
/* word or a parameter to the next line, and there should not *  
/* be any spaces, don't leave any blank spaces before the *  
/* continuation character '+'. @WA41026*  
/*  
/* If you specify all the three types of commands DELETE, ADD and *  
/* DEFINE then DELETEs are processed first, ALTERs are processed *  
/* next and ADDs are processed finally irrespective of the order *  
/* of the commands in the JOB.  
//*****  
//ADDVOL EXEC ACBJBAOB,  
// PLIB1='SYS1.DGTPLIB',  
// TABL2=userid.TEST.ISPTABL  
//SYSUDUMP DD SYSOUT=*  
//TEMPFILE DD DSN=&&VOLADDS,DISP=(NEW,KEEP),  
// SPACE=(TRK,(1,1)),LRECL=300,RECFM=F,BLKSIZE=300  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(ACBQBA19) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(99999999)  
/*  
//VOLADD DD *  
UPDHLVLSCDS()  
SCDSNAME(MYCD) VOL(SMS900) SG(PRIMARY) STATUS(ENABLE)  
SCDSNAME(MYCD) VOL(SMS901) SG(PRIMARY) STATUS(DISALL,+  
ENABLE,,DISNEW,,NOTCON,DISALL,ENABLE,QUIALL)  
SCDSNAME(MYCD) VOL(SMS902) SG(PRIMARY) +  
STATUS(DISNEW)  
SCDSNAME(MYCD) VOL(SMS903) SG(PRIMARY) +  
STATUS(DISALL,DISNEW,ENABLE,NOTCON,DISALL,DISNEW,DISALL,QUIALL,+  
DISALL,QUINEW,ENABLE,NOTCON,QUIALL,DISNEW,DISALL,QUIALL,+  
DISNEW,ENABLE,ENABLE,DISNEW,QUIALL,DISALL,DISALL,NOTCON,+  
NOTCON,QUINEW,ENABLE,NOTCON,QUIALL,DISNEW,ENABLE,QUIALL)  
SCDS(MYCD) VOL(SMS) FROM(14) TO(20) SUF(1) TYPE() SG(PRIMARY) STATUS(ENABLE) SCDS(MYCD) VOL(SMSA) FROM(A) TO(X) SUF(Y) TYPE(A)  
SG(PRIMARY) STATUS(ENABLE) SCDS(MYCD) VOL(SMSX) FROM(1) TO(C) SUF(G) TYPE(X) SG(PRIMARY) STATUS(ENABLE)
```

```

SG(PRIMARY) STATUS(ENABLE) SCDS(MYCDS) VOL(SMSX) FROM(1) TO(C) SUF(G) TYPE(X) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS904) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS905) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS906) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS907) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS908) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS909) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS910) SG(PRIMARY) STATUS(ENABLE)
SCDSNAME(MYCDS) VOL(SMS911) SG(PRIMARY) STATUSALL(ENABLE)
SCDSNAME(MYCDS) VOL(SMS912) SG(PRIMARY) STATUS(ENABLE) STATUSALL()
SCDSNAME(MYCDS) VOL(SMS913) SG(PRIMARY) STATUS() STATUSALL(ENABLE)
/*
//VOLALT DD *                                     /* 4@U1A*/
/*
//VOLDEL DD *
/*
//ADDVOL2 EXEC ACBJBAOB,
//          PLIB1='SYS1.DGTPLIB',
//          TABL2=userid.TEST.ISPTABL
//SYSUDUMP DD SYSOUT=*
//SYSTSIN DD DSN=&&VOLADDS,DISP=(OLD,DELETE)
//*****
```

Parent topic:Storage group volume add/Delete: ACBQBAI9



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ACS routine translate: ACBQBAO1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can use the ISMF translate option to translate the ACS routines and store them in an SCDS. After successful translation and validation, that SCDS can be activated and used by the system to manage storage. ACBQBAO1 is called by the following SYS1.SACBCNTL members to translate the ACS routines: **ACBJBAC2**Translate ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility**ACBJBAOQ**Translate ACS routines into an SCDS, followed by validating the SCDS. See [Figure 1](#) for the sample JCL and parameters.

[Sample JCL for ACS routine translate](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for ACS routine translate

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACJBAC2

```
/*
/* SAMPLE JCL TO DO FOUR OPERATIONS:
/*
/* 1. TRANSLATE ACS ROUTINES (ISMF OPTION 7.2)
/* 2. VALIDATE ACS ROUTINES (ISMF OPTION 7.3)
/* 3. TEST ACS ROUTINES (ISMF OPTION 7.4.3)
/* 4. COMPARE BASE & NEW ACS LISTINGS
/*
/* INSTRUCTIONS BEFORE SUBMITTING:
/*
/* CHANGE JOBCARD
/* CHANGE PREFIX
/* CHANGE PARAMETERS
/*
//***** ****
/*
/* TRANSLATE STEP:
/*
/* ACSSRC - PDS CONTAINING ACS ROUTINES TO BE
/* TRANSLATED (INPUT)
/* MEMBER - MEMBER NAME OF THE ROUTINE TO BE
/* TRANSLATED (INPUT)
/* SCDSNAME - NAME OF SCDS INTO WHICH THE ACS ROUTINES ARE
/* TO BE TRANSLATED (OUTPUT)
/* LISTNAME - TRANSLATE LISTING (OUTPUT)
/* UPDHLVLSCDS - CONFIRM OPERATION ON AN UPLEVEL SCDS @WA41441*
/* When modifying an SCDS, that was formatted with a *
/* higher level of SMS, using a lower level of SMS *
/* will make this application fail unless you *
/* specify the UPDHLVLSCDS parameter as 'Y'.
/* Default is 'N'. @WA41441*
/* Possible values : Y/N/BLANK @WA41441*
//***** ****
//TRANSLAT EXEC ACJBBAOB,
//      PLIB1='SYS1.DGTPLIB',
//      TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
```

Figure 2. Sample JCL for ACJBAC2 (continued)

```
DEL MGMTCLAS.LISTING
DEL STORGRP.LISTING
DEL STORCLAS.LISTING
ISPSTART CMD(ACBQBA01 +
ACSSRC(ACS.SOURCE) MEMBER(DATACLAS) +
SCDSNAME(MYSCDS) LISTNAME(DATACLAS.LISTING) +
UPDHLVLSCDS() +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(99999999)

ISPSTART CMD(ACBQBA01 +
ACSSRC(ACS.SOURCE) MEMBER(STORCLAS) +
SCDSNAME(MYSCDS) LISTNAME(STORCLAS.LISTING) +
UPDHLVLSCDS() +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(99999999)

ISPSTART CMD(ACBQBA01 +
ACSSRC(ACS.SOURCE) MEMBER(MGMTCLAS) +
SCDSNAME(MYSCDS) LISTNAME(MGMTCLAS.LISTING) +
UPDHLVLSCDS() +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(99999999)

ISPSTART CMD(ACBQBA01 +
ACSSRC(ACS.SOURCE) MEMBER(STORGRP) +
SCDSNAME(MYSCDS) LISTNAME(STORGRP.LISTING) +
UPDHLVLSCDS() +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(99999999)

/*
//***** ****
/*
/* COPY TRANSLATE LISTINGS
/*
/* SYSUT1 - INPUT (FROM PREVIOUS STEP)
/* SYSUT2 - OUTPUT
/*
//***** ****
//TRANGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.DATACLAS.LISTING,DISP=SHR
//          DD DSN=IBMUSER.STORCLAS.LISTING,DISP=SHR
//          DD DSN=IBMUSER.MGMTCLAS.LISTING,DISP=SHR
//          DD DSN=IBMUSER.STORGRP.LISTING,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//***** ****
/*
/* VALIDATE STEP:
/*
/* SCDSNAME - NAME OF SCDS THAT CONTAINS THE TRANSLATED ACS
/* ROUTINES TO BE VALIDATED (INPUT)
/* TYPE - TYPE OF ACS ROUTINE TO BE VALIDATED (INPUT)
/* LISTNAME - VALIDATE LISTING (OUTPUT)
/* UPDHLVLSCDS - CONFIRM OPERATION ON AN UPLEVEL SCDS(Y/N) @WA41441*
//***** ****
//VALIDAT EXEC ACJBBAOB,
//      PLIB1='SYS1.DGTPLIB',
//      TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
```

Figure 3. Sample JCL for ACJBAC2 (continued)

```

PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBA02 SCDSNAME(MYSCDS) TYPE(*) +
LISTNAME(VALIDAT.LISTING) +
UPDHLVLSCDS() +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****COPY VALIDATE LISTING
//*
//**      SYSUT1 - INPUT (FROM PREVIOUS STEP)
//**      SYSUT2 - OUTPUT
//*
//*****VALGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.VALIDAT.LISTING,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//*****TEST STEP
//*
//**      SCDSNAME - NAME OF SCDS (INPUT)
//**      TESTBED - PDS CONTAINING TEST CASES (INPUT)
//**      MEMBER - MEMBERS TO BE TESTED IN TESTBED (INPUT)
//**      DC,SC,MC,SG - ROUTINES TO BE TESTED (INPUT)
//**      LISTNAME - TEST LISTING (OUTPUT)
//*
//*****TEST EXEC ACBJBAOB,
//          PLIB1='SYS1.DGTPLIB',
//          TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
DEL NEW.LISTING
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAIA +
SCDSNAME(MYSCDS) +
TESTBED(TESTCASE LIBRARY) MEMBER(*) +
DC(Y) SC(Y) MC(Y) SG(Y) +
LISTNAME(NEW.LISTING)) +
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****COPY TEST LISTING
//*
//**      SYSUT1 - INPUT (FROM PREVIOUS STEP)
//**      SYSUT2 - OUTPUT
//*
//*****TSTGEN EXEC PGM=IEBGENER
//SYSUT1 DD DSN=IBMUSER.NEW.LISTING,DISP=SHR
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*

```

Figure 4. Sample JCL for ACJBAC2 (continued)

```

//*
//** COMPARE ACS LISTINGS
//*
//** PARAMETERS:
//*
//**      BASELIST - BASE ACS TEST LIST (INPUT)
//**      NEWLIST - NEW ACS TEST LIST (INPUT)
//**      TESTBED - PDS CONTAINING TEST CASES (REFERENCE INPUT)
//**      RSLTDSN - COMPARISON RESULTS DATA SET (OUTPUT)
//**      XCPTPDS - EXCEPTION TEST CASE PDS (OUTPUT)
//**      XCSPACE - SPACE values of Except DS (Optional) 3a@WA32832*
//**                  Values: (Primary Tracks,Secondary Tracks,Directory Blocks) *
//**                  which are positional and optional. Defaults: (3,1,20).
//*
//** NOTE: If you receive message IEC217I B14-0C on your exception *
//**        data set, you need to increase your data set size by using *
//**        the XCSPACE parameter (specially the directory blocks) *
//*
//*****COMPARE EXEC ACBJBAOB,
//          PLIB1='SYS1.DGTPLIB',
//          TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
DEL PDS4
DEL RESULTS.LISTING
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACBQBAC1 +
BASELIST(BASE.TESTLIST) +
NEWLIST(NEW.TESTLIST) +
TESTBED(TESTCASE LIBRARY) +
RSLTDSN(COMPARE.LISTING) +
XCPTPDS(TESTCASE.EXCP) +
XCSPACE(5,3,30)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)
/*
//*****COPY COMPARISON RESULTS
//*
//**      SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*

```

```
/*      SYSUT2 - OUTPUT          *
/*      *
/******EXEC PGM=IEBGENER
//SYSUT1   DD DISP=SHR,DSN=IBMUSER.COMPARE.LISTING
//SYSUT2   DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=*
/*
```

Parent topic:[ACS routine translate: ACBQBAO1](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SCDS validation: ACBQBAO2

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAO2 is called by the following SYS1.SACBCNTL members to validate the constructs:
ACBJBAOQ Translate all 4 ACS routines into an SCDS and validate
SCDSACBJBAC2 Translate all 4 ACS routines into an SCDS, validate the SCDS, run option 7.4.3 (test ACS routines) in batch, run the NaviQuest ACS comparison utility
ACBJBAOSISMF batch EXEC for validating the SCDS
See [Figure 1](#) for the sample JCL and parameters.

[Sample JCL for SCDS validation](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for SCDS validation

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAOS

```
//*****  
//** SAMPLE JCL TO PERFORM ISMF ACS VALIDATE IN BATCH *  
//**  
//** INSTRUCTIONS BEFORE SUBMITTING: *  
//**  
//**     CHANGE JOBCARD *  
//**     CHANGE PREFIX *  
//**     CHANGE PARAMETERS *  
//**  
//*****  
//*****  
//**  
//** VALIDATE STEP: *  
//**  
//**     SCDSNAME - NAME OF SCDS THAT CONTAINS THE TRANSLATED ACS *  
//**             ROUTINES TO BE VALIDATED (INPUT) *  
//**     TYPE      - TYPE OF ACS ROUTINE TO BE VALIDATED (INPUT) *  
//**     LISTNAME  - VALIDATE LISTING (OUTPUT) *  
//**     UPDHLVLSCDS - When modifying an SCDS, that was formatted with *  
//**                     a higher level of SMS, using a lower level of *  
//**                     SMS will make this application fail unless you *  
//**                     specify the UPDHLVLSCDS parameter as 'Y'. *  
//**                     Default is 'N'. @WA41441*  
//**  
//**             Possible values : Y/N/BLANK @WA41441*  
//*****  
//VALIDAT EXEC ACBJBAOB,  
//           PLIB1='SYS1.DGTPLIB',  
//           TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
DEL VALIDAT.LISTING  
ISPSTART CMD(ACBQBA02 +  
SCDSNAME(DSNREPN) TYPE(*) +  
LISTNAME(VALIDAT.LISTING) +  
UPDHLVLSCDS() +  
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//*****  
//**  
//** COPY THE VALIDATE LISTING DATA SET *  
//**  
//**     SYSUT1 - INPUT (FROM PREVIOUS STEP) *  
//**     SYSUT2 - OUTPUT *  
//**  
//*****  
//STEP2 EXEC PGM=IEBGENER  
//SYSUT1 DD DSN=IBMUSER.VALIDAT.LISTING,DISP=SHR  
//SYSUT2 DD SYSOUT=*  
//SYSIN DD DUMMY  
//SYSPRINT DD SYSOUT=*  
//*****
```

Parent topic:[SCDS validation: ACBQBA02](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Test ACS routines: ACBQBAIA

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAIA is called by the following SYS1.SACBCNTL members to test the ACS routines in batch: **ACBJBAC2**Translate all 4 ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility**ACBJBAIC**Test ACS routines

See [Figure 1](#) for the sample JCL and parameters.

[Sample JCL for test ACS routines](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for test ACS routines

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAIC

```
//*****  
//** SAMPLE JCL TO TEST ACS ROUTINES IN BATCH *  
//** *  
//** INSTRUCTIONS BEFORE SUBMITTING: *  
//** *  
//**     CHANGE JOBCARD *  
//**     CHANGE PREFIX *  
//** *  
//** TEST STEP *  
//** *  
//** SCDSNAME - NAME OF SCDS THAT CONTAINS THE TRANSLATED, *  
//**             VALIDATED ACS ROUTINES TO BE TESTED (INPUT) *  
//** TESTBED - PDS CONTAINING TEST CASES THAT THE ACS ROUTINES *  
//**             SHOULD BE TESTED FOR (INPUT) *  
//** MEMBER - MEMBERS TO BE TESTED IN TESTBED (INPUT) *  
//** DC,SC,MC,SG - ROUTINES TO BE TESTED Y OR N (INPUT) *  
//** LISTNAME - TEST LISTING (OUTPUT) *  
//** *  
//*****  
//TESTACS EXEC ACBJBAOB,  
//          PLIBL='SYS1.DGTPLIB',  
//          TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
DEL NEW.TESTLIST  
ISPSTART CMD(ACBQBAIA +  
SCDSNAME(MYSCDS) +  
TESTBED(TESTCASE.LIBRARY) MEMBER(*) +  
LISTNAME(NEW.TESTLIST) +  
DC(Y) SC(Y) MC(Y) SG(Y)) +  
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMax(9999999)  
/*  
//*****  
//** COPY TEST LISTING */  
//** */  
//** */  
//** SYSUT1 - INPUT (FROM PREVIOUS STEP) */  
//** SYSUT2 - OUTPUT */  
//** */  
//*****  
//TESTGEN EXEC PGM=IEBGENER  
//SYSUT1 DD DSN=IBMUSER.NEW.TESTLIST,DISP=SHR  
//SYSUT2 DD SYSOUT=*  
//SYSIN DD DUMMY  
//SYSPRINT DD SYSOUT=*
```

Parent topic: Test ACS routines: ACBQBAIA



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate test cases from ISMF-saved data set lists: ACBQBAG3

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAG3 is called by SYS1.SACBCNTL member ACBQBAG2 to take an ISMF-saved data set table and generate test case members in a PDS library based on the attributes of the data sets in the ISMF table.

See [Figure 1](#) for the sample JCL and parameters.

Recommendations and Restrictions: Saving tables of temporary data sets might produce errors in the bulk test case generate option (11.1.1). Instead, generate test cases from the ACSTST program for temporary data sets. Before generating the list, set the ACQUIRE DATA FROM VOLUME and ACQUIRE DATA IF DFHSM options under the ISMF data set selection entry panel to Y. The MULTVOL variable is always set to YES in an ISMF table if the data set has not been opened at the time the table is saved. The value is set correctly at OPEN time. This can sometimes cause errors in the bulk test case generator.

[Sample JCL for generating test cases from ISMF-saved data set lists](#)

Parent topic: Performing storage administration tasks in batch



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating test cases from ISMF-saved data set lists

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for

ACBJBAG2//*****
/*
/* GENERATE TEST CASES FROM A PREVIOUSLY SAVED ISMF DATA SET LIST *
/*
/* TABLENM - SAVED ISMF DATA SET LIST (INPUT) *
/* PRFX - MEMBER NAME PREFIX *
/* TESTPDS - TEST CASE PDS (OUTPUT) *
/* REPLACE - REPLACE CONTENTS IF DSN EXISTS *
/* INTEST - DEBUG MODE YES/NO (KEEP IT AS NO) *
/*
/******
//GENTC EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(%ACQBAG3 TABLENM(DSNLIST) +
INTEST(NO) PRFX(TEST) +
TESTPDS('IBMUSER.TESTCASE LIBRARY') REPLACE(Y)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(9999999)
/*
//*****

Parent topic:[Generate test cases from ISMF-saved data set lists: ACQBAG3](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate test cases from DCOLLECT data: ACBQBAG1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAG1 is called by SYS1.SACBCNTL member ACBJBAG1 to generate test cases from DCOLLECT data.

See [Figure 1](#) for the sample JCL and parameters:

[Sample JCL for generating test cases from DCOLLECT data](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating test cases from DCOLLECT data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for

ACBQBAG1

```
//*****  
//** SAMPLE JCL TO GENERATE TEST CASES FROM DCOLLECT DATA *  
//** *  
//** INSTRUCTIONS BEFORE SUBMITTING: *  
//** *  
//** CHANGE JOBCARD *  
//** CHANGE PREFIX *  
//** CHANGE PARAMETERS *  
//** *  
//** THE PARAMETERS TO ACBQBAG1 ARE AS FOLLOWS: *  
//** *  
//** PARAMETER 1 IS DATA SET CONTAINING DCOLLECT DATA (INPUT) *  
//** PARAMETER 2 IS NUMBER OF TEST CASES TO BE GENERATED *  
//** PARAMETER 3 IS MEMBER NAME PREFIX *  
//** PARAMETER 4 IS TEST CASE PDS (OUTPUT) *  
//** PARAMETER 5 IS REPLACE CONTENTS IF DSN EXISTS *  
//** *  
//*****  
//GENTEST EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAG1 'IBMUSER.DCOLLECT.DATA' 10 TEST +  
'IBMUSER.TESTCASE.LIBRARY' Y) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*  
//*****
```

Parent topic:[Generate test cases from DCOLLECT data: ACBQBAG1](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate test cases from SMF data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SYS1.SACBCNTL member ACBJBAI1 contains JCL to generate test cases from SMF type 127 type records in batch.

You will need to provide the names of the input and output data sets on the INDD and OUTDD DD statements: **INDD**Data set containing SMF data
(input)**OUTDD**Test case PDS (output)

See [Figure 1](#) for the sample JCL and parameters.

SMF type 127 records are written by storage class exit IGDACSSC. Use a utility like IFASMFDP to unload the SMF type 127 records from the SMF log data sets. You can also use ACSTST program (from CBIP0).

[Sample JCL for generating test cases from SMF data](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating test cases from SMF data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for

ACBJBAI1 //*****
/*
/* SAMPLE JCL TO GENERATE TEST CASES FROM SMF TYPE 127 RECORDS *
/* WRITTEN BY THE STORAGE CLASS EXIT IGDACSSC; USE A STANDARD *
/* UTILITY LIKE IFASMFDP TO UNLOAD THE TYPE 127 RECORDS FROM THE *
/* SMF LOG DATA SETS; ALSO ACSTST PROGRAM (FROM CBIPO) SHOULD BE *
/* AVAILABLE *
/* *
/* INDD - DATA SET CONTAINING SMF DATA (INPUT) *
/* OUTDD - TEST CASE PDS (OUTPUT) *
/* *
//*****
//GENTEST EXEC PGM=ACSTST,REGION=512K,
// COND=(0,NE,DATA1)
//SYSPRINT DD SYSOUT=*
//INDD DD DSN=IBMUSER.SMF.TYPE127.DATA,DISP=SHR
//OUTDD DD DSN=IBMUSER.TESTCASE.LIBRARY,DISP=(NEW,CATLG,DELETE),
// SPACE=(80,(1250,200,50),,ROUND),
// UNIT=3380,VOL=SER=M4RS05,
// DCB=(BLKSIZE=80,LRECL=80,RECFM=F)
//*****

Parent topic: Generate test cases from SMF data



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate test cases from VMA extract data: ACBQBAO3

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAO3 is called by SYS1.SACBCNTL member ACBJBAOW to generate test cases from records in the VMA extract file of previously created VMA data.

See [Figure 1](#) for the sample JCL and parameters.

Tip: You can also use the ACSTST program to generate test cases.

[Sample JCL for generating test cases from VMA](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating test cases from VMA

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for
ACBJBAOW

```
*****  
/* *  
/* * SAMPLE JCL TO GENERATE TEST CASES FROM VMA EXTRACT DATA *  
/* *  
/* * INSTRUCTIONS BEFORE SUBMITTING: *  
/* *  
/* *      CHANGE JOBCARD *  
/* *      CHANGE PREFIX *  
/* *      CHANGE PARAMETERS *  
/* *  
/* * PARAMETERS: *  
/* *  
/* *      PARAMETER 1 - DATA SET CONTAINING VMA EXTRACT DATA *  
/* *      PARAMETER 2 - NUMBER OF TEST CASES TO BE GENERATED *  
/* *      PARAMETER 3 - MEMBER NAME PREFIX *  
/* *      PARAMETER 4 - PROGRAM NAME TO FILTER ON *  
/* *  
/* *****  
//TESTGEN EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAO3 'IBMUSER.VMA.DATA' 100 TEST) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(9999999)  
/*
```

Parent topic:[Generate test cases from VMA extract data: ACBQBAO3](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

ACS test listings comparison: ACQBAC1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACQBAC1 is called by the following SYS1.SACBCNTL members to perform a detailed comparison of the differences between the "base" and "new" ACS listing:

ACBJBAC1Run the NaviQuest ACS Comparison utility.**ACBJBAC2**Translate all 4 ACS routines into an SCDS, validate the SCDS, test ACS routines, and run the NaviQuest ACS comparison utility.

See [Figure 1](#) for the sample JCL and parameters:

[Sample JCL for test listings comparison](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for test listings comparison

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for

ACBJBAC1 //*****
/*
/* SAMPLE JCL TO COMPARE ACS TEST LISTINGS IN BATCH
/*
/* INSTRUCTIONS BEFORE SUBMITTING:
/*
/* CHANGE JOBCARD
/* CHANGE PREFIX
/* CHANGE PARAMETERS
/*
/* PARAMETERS:
/*
/* BASELIST - BASE ACS TEST LISTING (INPUT)
/* NEWLIST - NEW ACS TEST LISTING (INPUT)
/* TESTBED - TEST CASE PDS (REFERENCE INPUT)
/* RSLTDSN - COMPARISON RESULTS DATA SET (OUTPUT)
/* XCPTPDS - EXCEPTION TEST CASE PDS (OUTPUT)
/* XCSPACE - SPACE values of Except DS (Optional) 3a@WA32832*
/* Values: (Primary Tracks,Secondary Tracks,Directory Blocks)
/* which are positional and optional. Defaults: (3,1,20).
/*
/* NOTE: If you receive message IEC217I B14-0C on your exception
/* data set, you need to increase your data set size by using
/* the XCSPACE parameter (specially the directory blocks)
/*
/* *****
//CMPRSTEP EXEC ACBJBAOB,PLIB=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
DEL COMPARE.LISTING
DEL TESTCASE.EXCP
ISPSTART CMD(%ACBQBAC1 +
BASELIST(BASE.TESTLIST) +
NEWLIST(NEW.TESTLIST) +
TESTBED(TESTCASE LIBRARY) +
RSLTDSN(COMPARE.LISTING) +
XCPTPDS(TESTCASE.EXCP) +
XCSPACE(5,3,30)) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDSPMAX(9999999)
/*
/* *****
/*
/* COPY THE COMPARISON LISTINGS DATA SET
/*
/* SYSUT1 - INPUT (FROM PREVIOUS STEP)
/* SYSUT2 - OUTPUT
/*
/* *****
//REPGEN EXEC PGM=IEBGENER
//SYSUT1 DD DISP=SHR,DSN=IBMUSER.CMPARE.LISTING
//SYSUT2 DD SYSOUT=*
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//

Parent topic:[ACS test listings comparison: ACBQBAC1](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Enhanced ACS test listing: ACBQBAX1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAX1 is called by SYS1.SACBCNTL member ACBJBAX1 to generate a detailed cross-reference listing report from the original ACS listing. The cross-reference listing report is used to help determine where there are logic errors in an ACS routine. The report can include the data set name, unit type, data set size, expiration date, job name, and program name for each exception test case.

See [Figure 1](#) for the sample JCL and parameters.

Use the following parameters with the ACBQBAX1:

[Sample JCL for generating an enhanced ACS test listing](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating an enhanced ACS test listing

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACJBAX1

```
/* SAMPLE JCL TO GENERATE ENHANCED ACS TEST LISTING IN BATCH
 */
//***** STEP TO GENERATE ENHANCED ACS TEST LISTING
/*
/* PARAMETERS:
/*
/*      ACSDSN - ACS TEST LISTING (INPUT)
/*      XREFDSN - ENHANCED ACS TEST LISTING (OUTPUT)
/*      XREFSPC - The Space parameters can be specified 12@WA42064*
/*                  for XREFDSN dataset using this parameter.
/*
/*      Syntax ==> XREFSPC(PRI,SEC) Where
/*                  PRI -> Primary space in Tracks
/*                  SEC -> Secondary space in Tracks
/*                  to be allocated.
/*      Eg. XREFSPC(10,20)
/*
/*      If this parameter is not specified, by default
/*      ACSDSN dataset space parameters are used to
/*      allocate XREFDSN dataset.
/*
/*      OUTPUT SELECTION CRITERIA: (DATA COMES FROM TEST CASE PDS)
/*
/*      DSNINFO - DSN - DATA SET NAME
/*
/*      UNITINFO - UNIT - UNIT ON WHICH THE DATA SET RESIDES
/*
/*      SIZEINFO - SIZE - SIZE OF DATA SET
/*
/*      EXPTINFO - EXPDT - EXPIRY DATE OF THE DATA SET
/*
/*      JOBINFO - JOB - JOB WHICH ALLOCATED THE DATA SET
/*
/*      PGMINFO - PGM - PROGRAM WHICH ALLOCATED THE DATA SET
/*
*/
//*****
```

Figure 2. Sample JCL for ACJBAX1 (continued)

```
//PLIB1=SYS1.DGTPLIB,
//TABL2=userid.TEST.ISPTABL
//SYSTSIN DD *
PROFILE PREFIX(IBMUSER)
DEL RSLT1
%ACQBAX1 +
ACSDSN(BASE.LISTING) +
XREFDSN(RSLT1) +
XREFSPC() +
DSNINFO(Y) +
UNITINFO(Y) +
SIZEINFO(Y) +
EXPTINFO(Y) +
JOBINFO(Y) +
PGMINFO(Y)
/*
//***** COPY ENHANCED ACS TEST LISTING
/*
/*      SYSUT1 - INPUT (FROM PREVIOUS STEP)
/*      SYSUT2 - OUTPUT
/*
//*****/REPGEN EXEC PGM=IEBGENER
//SYSIN DD DUMMY
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD DISP=SHR,DSN=IBMUSER.ENHANCED.LISTING
//SYSUT2 DD SYSOUT=*
/*
//*****
```

Parent topic:[Enhanced ACS test listing: ACQBAX1](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Update test cases with expected results: ACBQBAU1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAU1 is called by SYS1.SACBCNTL member ACBJBAU2 to update the testbed library with the expected results for the new data set type that has been placed in the *new* ACS listing.

Tip: ACBQBAU1 is I/O intensive, as it must update each test case member in the PDS with the new expected results.

See [Figure 1](#) for the sample JCL and parameters:

[Sample JCL to update test cases](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL to update test cases

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for

ACBJBAU2

```
//*****  
//** SAMPLE JCL TO UPDATE TEST CASES WITH TEST RESULTS IN BATCH *  
//** *  
//** INSTRUCTIONS BEFORE SUBMITTING: *  
//** *  
//**     CHANGE JOBCARD *  
//**     CHANGE PREFIX *  
//**     CHANGE PARAMETERS *  
//** *  
//** PARAMETERS: *  
//** *  
//**     NEWLIST - NEW ACS TEST LISTING *  
//**     TESTBED - TEST CASE PDS *  
//**     XCPTDSN - EXCEPTION TEST CASE PDS *  
//**     RSLTDSN - COMPARISON RESULTS DATA SET *  
//** *  
//*****  
//UPDTE EXEC ACBQBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAU1 +  
NEWLIST(NEW.TESTLIST) +  
XCPTPDS(TESTCASE.EXCP) +  
TESTBED(TESTCASE.LIBRARY) +  
RSLTDSN(COMPARE,LISTING)) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)  
/*
```

Parent topic:[Update test cases with expected results: ACBQBAU1](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate report from ISMF-saved data set list: ACBQBAR1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAR1 EXEC is called by SYS1.SACBCNTL member ACBJBAOD to generate a flat file from an ISMF-saved data set table, listing the fields of your choice, in the order you specify.

See [Figure 1](#) for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR1 EXEC. Each parameter must be on a separate line.

[Sample JCL for generating a report from ISMF-saved data set list](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating a report from ISMF-saved data set list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAOD

```
/*
/* SAMPLE JCL TO GENERATE DATA SET REPORT FROM A PREVIOUSLY SAVED *
/* ISMF DATA SET LIST *
/*
/* INSTRUCTIONS BEFORE SUBMITTING: *
/*
/* CHANGE JOBCARD *
/* CHANGE PREFIX *
/* CHANGE PARAMETERS *
/*
***** *
/*
/* DELETE STEP TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY *
/*
***** *
//DELREP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE IBMUSER.DATASET.REPORT
/*
***** *
/*
/* STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED *
/* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY *
/* DATA SET. *
/*
***** *
//ALCISPFL EXEC PGM=IEFBR14
//ISPFILE DD DSN=IBMUSER.DATASET.REPORT,DISP=(NEW,CATLG),
// BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
/*
***** *
/*
/* DATA SET REPORT GENERATION STEP *
/*
/* PARAMETER FOLLOWING ACQBART - SAVED ISMF LIST (INPUT) *
/* ISPFILE - DATA SET REPORT (OUTPUT, FROM ALCISPFL STEP) *
/* SYSIN - KEY WORDS SPECIFYING DATA IN THE REPORT *
/*
/* The Following parameters can be specified in SYSIN        42@U1A*
/*
/*
/* %NOTUSED          -> % Space Not Used. *
/* ALLOCSP | ALLOCSPC -> Allocated Space. *
/* ALLOCUSED         -> Used Space. *
/* BACKUP | LASTBKUP -> Last Backup Date. *
/* BLKSIZE | CISIZE   -> Block/CI Size. *
/* BLKUNUSED         -> Blocks Unused. *
/* CCSIDES           -> CCSID Description. *
/* CFMONST           -> CF Monitor Status. *
/* CFSTATUS          -> CF Status Indicator. *
/* CHANGE | CHGIND   -> Change Indicator. *
/* COMPRESS          -> Compressed Format. *
/* CREATE | CREATEDT -> Creation Date. *
/* CSETNAME          -> CF Cache set name.      1@U2C*
/* CSTRNAME          -> CF Cache Structure Name. *
```

Figure 2. Sample JCL for ACBJBAOD (continued)

Name.	*	/* DC	DATACLAS	-> Data Class
DDMATTR				
DEVICE	DEVTYPE	-> DDM attribute.		*
DSNAME	DSN	-> Device Type.		*
DSNAMETY		-> Dataset Name.		*
DSORG		-> Dataset Name Type.		*
ENTRY	ENTRYTYPE	-> Data Set Organization.		*
ENVIRONMANT	ENVIR	-> Dataset Entry Type.		*
EXPIRE	EXPIREDT	-> Dataset Environment.		*
LASTREF	LASTREFDT	-> Expiration Date.		*
LRECL		-> Last referenced Date.		*
MC	MGMTCLAS	-> Record Length.		*
MULTVOL		-> Management Class Name.		*
NUMEXT	EXTNUM	-> Multi Volume Status.		*
NUMSTR		-> Number of Extents.		*
OPTIMAL		-> Number of Stripes.		*
OWNER		-> Optimal Block/CI Size.		*
REBLK	REBLOCK	-> Owner.		*
RECFM	RECFMT	-> Reblock Indicator.		*
SC	STORCLAS	-> Record Format.		*
SECALLOC	ALLOCSEC	-> Storage Class Name.		*
TOTALS		-> Secondary Allocation.		*
UNIT	ALLOCUNI	-> Total Allocation.		*
USERRED%		-> Allocation Unit.		*
VOLSER		-> % User Data Reduction.		*
LSETNAME		-> Volume Serial.		*
		-> CF Lock set name. @A1A*	/* LSTRNAME	-> CF Lock Structure Name.

//GENREP EXEC ACBJBAOB,				
// PLIB1=SYS1.DGTPLIB,				
// TABL2=userid.TEST.ISPTABL				
//ISPFILE DD DSN=IBMUSER.DATASET.REPORT,DISP=OLD				
//SYSTSIN DD *				
PROFILE PREFIX(IBMUSER)				
ISPSTART CMD(ACQBART DSNLIST) +				
BATSCRW(132) BATCRD(27) BREDIMAX(3) BDISPMAX(9999999)				
/*				
//SYSIN DD *				
TITLE=LIST OF DATASETS FOR HIGHLEVEL QUALIFIER IBMUSER				
DSN				
VOLSER				
LASTREF				
EXTNUM				
TOTALS				

```
TOTALS
/*
//*****COPY THE DATA SET REPORT*****
//*
//*      SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*      SYSUT2 - OUTPUT
//*
//*****STEP2 EXEC PGM=IEBGENER
//SYSUT1   DD DSN=IBMUSER.DATASET.REPORT,DISP=SHR
//SYSUT2   DD SYSOUT=*
//SYSIN    DD DUMMY
//SYSPRINT DD SYSOUT=*
```

Parent topic:Generate report from ISMF-saved data set list: ACBQBAR1



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate report from ISMF-saved DASD volume list: ACBQVAR1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQVAR1 is called by SYS1.SACBCNTL member ACBJBAOF to generate a flat file from an ISMF-saved DASD volume table and lists the fields of your choice, in the order you specify. You can sort on a field.

See [Figure 1](#) for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQVAR1 EXEC in batch. Each parameter must be on a separate line.

[Sample JCL for generating a report from ISMF-saved DASD volume list](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating a report from ISMF-saved DASD volume list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAOF

```
/*
/* SAMPLE JCL TO SORT A PREVIOUSLY SAVED DASD VOLUME LIST AND      @D1C*
/* GENERATE A DASD VOLUME REPORT FROM IT.                            @D1C*
/*
/* INSTRUCTIONS BEFORE SUBMITTING:
/*
/*     CHANGE JOBCARD
/*     CHANGE PREFIX
/*     CHANGE PARAMETERS
/*
/* ****
/* **** DELETE STEP TO DELETE THE REPORT IF IT EXISTS ALREADY      *
/*
/* ****
//DELREP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE IBMUSER.DASDVOL.REPORT
/*
/* ****
/* STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED    *
/* NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY      *
/* DATA SET.
/*
/* ****
//ALCISPFL EXEC PGM=IEFBR14
//ISPFILE DD DSN=IBMUSER.DASDVOL.REPORT,DISP=(NEW,CATLG),
//          BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
/*
/* ****
/* STEP TO SORT A PREVIOUSLY SAVED DASD VOLUME LIST AND            @D1C*
/* GENERATE A DASD VOLUME REPORT FROM IT.                            @D1A*
/*
/* PARAMETERS FOLLOWING ACBQVAR1 - ISMF SAVED LIST (INPUT);           @D1C*
/*                      SORT(column name) (INPUT) - 3@D1A*
/*                      to sort list by the data column   *
/*                      in ascending order.               *
/*
/* ISPFILE - DASD VOLUME REPORT (OUTPUT, FROM ALCISPFL STEP)        *
/* SYSIN - KEY WORDS TO SPECIFY THE DATA IN THE REPORT              *
/*
/* The following parameters can be specified in SYSIN.      25@U1A *
/*
*/
```

Figure 2. Sample JCL for ACBJBAOF (continued)

Serial	*	VOLSER	-> Volume
FREESPC	FREESP	-> Free Space	*
FRSPCTRK	FRSPTRK	-> Free Space in TRK-Managed	@A0A*
%FREE		-> % Free	*
ALLOCSPC	ALLOCSP	-> Allocated Space	*
ALSPCTRK	ALSPTRK	-> Alloc Space in TRK-Managed	@A0A*
FRAG	FRAGINDX	-> Fragmentation Index	*
LRGEXT	EXTLRG	-> Largest Extent	*
LGEXTRK	EXTLGTRK	-> Largest Ext in TRK-Managed	@A0A*
FREEXT	EXTFREE	-> Free Extents	*
IDXSTAT	STATINDX	-> Index Status	*
FREEDSCB	DSCBFREE	-> Free DSCBS	*
FREEVIRS	VIRSFREE	-> Free Vtoc Index Records	*
DEVICE	DEVICETYPE	-> Device Type	*
DEVMNUM	ADDRESS	-> Device Number	*
SHARE	SHRDASD	-> Shared DASD	*
USE	USEATTR	-> Use Attributes	*
RDCACHE	RDSTAT	-> RD Cache Status	*
DASDFW		-> DASD FW Status	*
CACHEFW		-> Cache FW Status	*
DUPLEX	DUPLEXST	-> Duplex Status	*
OTHER	OTHERDEV	-> Other Device	*
SUBSYS	SUBSYSID	-> Subsys ID	*
PHYSTAT	STATPHYS	-> Physical Status	*
STORGRP	SG	-> Storage Group Name	*
CFVOLST		-> CF Volume Status	*
INIASRES		-> Initialized as Reserved	* /* OWNERID
-> Owner ID			

GENREP EXEC ACBJBAOB,			
PLIB1=SYS1.DGTPLIB,			
TABL2=userid.TEST.ISPTABL			
ISPFILE DD DSN=IBMUSER.DASDVOL.REPORT,DISP=OLD			
SYSTSIN DD *			
PROFILE PREFIX(IBMUSER)			
ISPSTART CMD(ACBQVAR1 DASDLIST SORT(column name)) +			
NEWAPPL(DGT) BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(99999999)			
/*			
//SYSIN DD *			
VOLSER			
IDXSTAT			
FRAG			
USEATTR			
SG			
TITLE=STATUS OF VOLUMES THAT ARE SMS MANAGED			
/*			
/* ****			
/* */			
/* /* COPY THE DASD VOLUME REPORT			*
/* */			*

```
/*  
/*   SYSUT1 - INPUT (FROM PREVIOUS STEP)  
/*   SYSUT2 - OUTPUT  
/*  
*****  
//REPGEN EXEC PGM=IEBGENER  
//SYSUT1 DD DSN=IBMUSER.DASDVOL.REPORT,DISP=SHR  
//SYSUT2 DD SYSOUT=*  
//SYSIN DD DUMMY  
//SYSPRINT DD SYSOUT=*  
/*
```

Parent topic:Generate report from ISMF-saved DASD volume list: ACBQVAR1



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate report from ISMF-saved tape list: ACBQBAR4

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAR4 is called by SYS1.SACBCNTL member ACBJBAOT to generate a flat file from an ISMF-saved tape table and lists the fields of your choice, in the order you specify.

See [Figure 1](#) for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR4 EXEC in batch. Each parameter must be on a separate line.

[Sample JCL for generating a report from ISMF-saved tape list](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating a report from ISMF-saved tape list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACBJBAOT

```
//*****  
//** SAMPLE JCL TO GENERATE TAPE VOLUME REPORT FROM A PREVIOUSLY *  
//** SAVED ISMF MOUNTABLE TAPE VOLUME LIST *  
//** *  
//** INSTRUCTIONS BEFORE SUBMITTING: *  
//** *  
//**     CHANGE JOBCARD *  
//**     CHANGE PREFIX *  
//**     CHANGE PARAMETERS *  
//** *  
//*****  
//*****  
//** DELETE STEP, TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY *  
//** *  
//*****  
//DELREP EXEC PGM=IDCAMS  
//SYSPRINT DD SYSOUT=*  
//SYSIN DD *  
DELETE IBMUSER.TAPEVOL.REPORT  
/*  
//*****  
//** *  
//** STEP TO ALLOCATE ISPFILE, WHERE THE GENERATED REPORT IS SAVED *  
//** NOTE: THE DATA SET BEING ALLOCATED SHOULD NOT BE A TEMPORARY *  
//** DATA SET. *  
//** *  
//*****  
//ALCISPFL EXEC PGM=IEFBR14  
//ISPFILE DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=(NEW,CATLG),  
//    BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA  
//*****  
//** REPORT GENERATION STEP *  
//** *  
//**     PARAMETER FOLLOWING ACBQBAR4 - ISMF SAVED LIST NAME (INPUT) *  
//**     ISPFILE - TAPE VOLUME REPORT (OUTPUT, FROM ALCISPFL STEP) *  
//**     SYSIN - KEYWORDS SPECIFYING THE DATA IN THE REPORT *  
//** *  
//*****  
//GENREP EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//ISPFILE DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=OLD  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(ACBQBAR4 TAPELIST) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(9999999)  
/*  
//SYSIN DD *  
VOLSER  
LASTMOUNT  
LASTEJECT  
USEATTR  
STORGRP  
LIBNAME  
TITLE=STATUS OF TAPES AS ON 06/01/96  
/*
```

Figure 2. Sample JCL for ACBJBAOT (continued)

```
//*****  
//** COPY THE TAPE VOLUME REPORT *  
//** *  
//**     SYSUT1 - INPUT (FROM PREVIOUS STEP) *  
//**     SYSUT2 - OUTPUT *  
//** *  
//*****  
//REPGEN EXEC PGM=IEBGENER  
//SYSUT1 DD DSN=IBMUSER.TAPEVOL.REPORT,DISP=SHR  
//SYSUT2 DD SYSOUT=*  
//SYSIN DD DUMMY  
//SYSPRINT DD SYSOUT=*  
//*****
```

Parent topic:Generate report from ISMF-saved tape list: ACBQBAR4



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate data set report from DCOLLECT data: ACBQBAR7

z/OS DFSMSdfp Storage Administration
SC23-6860-01

ACBQBAR7 is called by SYS1.SACBCNTL member ACBJBARD to generate a flat file from DCOLLECT data taken from data set records and lists the fields of your choice, in the order you specify.

See [Figure 1](#) for the sample JCL and parameters.

Use the following parameters on the SYSIN DDNAME statement when you run the ACBQBAR7 EXEC in batch. Each parameter must be on a separate line.

[Sample JCL for generating data set report from DCOLLECT data](#)

Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating data set report from DCOLLECT data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for ACJBARD

```
/*
** SAMPLE JCL TO GENERATE DATA SET REPORT FROM DCOLLECT DATA
*/
/* INSTRUCTIONS BEFORE SUBMITTING:
 */
/* CHANGE JOBCARD
 */
/* CHANGE PREFIX
 */
/* CHANGE PARAMETERS
 */
/*
** DELETE STEP - TO DELETE THE REPORT DATA SET IF IT EXISTS ALREADY *
*/
/*
***** DELREP EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DELETE IBMUSER.DATASET.REPORT
/*
***** ALCISPFL EXEC PGM=IEFBR14
//ISPFIL DD DSN=IBMUSER.DATASET.REPORT,DISP=(NEW,CATLG),
//BLKSIZE=0,SPACE=(TRK,(3,1)),RECFM=FBA,LRECL=133,UNIT=SYSDA
/*
** REPORT GENERATION STEP
 */
/* DCOLIN - DCOLLECT DATA (INPUT)
 */
/* ISPFIL - DATA SET REPORT (OUTPUT, FROM ALCISPFL STEP)
 */
/* SYSIN - KEY WORDS TO SPECIFY THE DATA IN THE OUTPUT
 */
/* CHOOSE FROM THE FOLLOWING PARAMETERS FOR DATA TO BE LISTED
 */
/* THE ORDER CHOSEN DETERMINES THE ORDER IN THE REPORT
 */
/* THE TOTAL REPORT WIDTH MAY NOT EXCEED 133
 */
/* PARAMETERS:
 */
/* ALLOCSP | ALLOCSPC - ALLOCATED SPACE
 */
/* BACKUP | LASTBKUP - DATA SET'S LAST BACKUP DATE
 */
/* BLKUNUSED - BLOCKS UNUSED
 */
/* CHANGE | CHGIND - CHANGE INDICATOR FOR DATA SET
 */
/* CREATE | CREATEDT - DATA SET CREATION DATE
 */
/* DATACLAS | DC - DATACLAS OF DATA SET
 */
/* DSNAME | DSN - DATASETNAME
 */
/* DSLENGTH - LIMIT DATASET NAME TO THIS NUMBER
 */
/* OF CHARACTERS
 */

ORGANIZATION * DSORG - DATA SET
/*
 */
/* ENTRYTYPE - ENTRY TYPE OF DATA SET
 */
/* EXPIRE | EXPIREDT - DATA SET EXPIRATION DATE
 */
/* LASTREF | LASTREFDT - DATA SET LAST REFERENCE DATE
 */
/* LRECL - RECORD LENGTH
 */
/* MGMTCLAS | MC - MANAGEMENT CLASS FOR DATA SET
 */
/* MULTVOL - IS THE DATA SET MULTI-VOLUME?
 */
/* NUMEXT | EXTNUM - NUMBER OF EXTENTS
 */
/* PAGELENGTH - NUMBER OF LINES PER PAGE (DEF IS 60)*
 */
/* PDSE - IS THE DATA SET A PDSE?
 */
/* REBLOCK | REBLK - IS THE DATA SET REBLOCKABLE?
 */
/* RECFM | RECFMT - RECORD FORMAT
 */
/* SMS | MANAGED - IS THE DATA SET MANAGED?
 */
/* STORCLAS | SC - STORAGE CLASS FOR DATA SET
 */
/* STORGRP | SG - STORAGE GROUP FOR DATA SET
 */
/* TITLE=XXXXX - TITLE FOR REPORT
 */
/* TOTALS - PRINT DSN SPACE TOTALS
 */
/* USED% - USED SPACE %
 */
/* VOLSEQ - VOLUME SEQUENCE NUMBER
 */
/* VOLSER - VOLUME SERIAL
 */
/* VVRCHK | VVR - IF SMS, IS THERE A VVR OR NVR? *
 */
/*
 */
/* GENREP EXEC ACJBBAOB,PLIB1=SYS1.DGTLIB,TABL2=userid.TEST.ISPTABL
 */
/* DCOLIN DD DSN=IBMUSER.DCOLLECT.DATA,DISP=SHR
 */
/* ISPFIL DD DSN=IBMUSER.DATASET.REPORT,DISP=OLD
 */
/* SYSTSIN DD *
 */
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR7) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(9999999)
/*
 */
/* SYSIN DD *
 */
DSN
VOLSER
ALLOCSP
RECFM
BLKUNUSED
TITLE=DATA SET REPORT FROM DCOLLECT DATA - 06/01/96
TOTALS
BLKSIZE
EXTNUM
DSORG
STORCLAS
```

Figure 2. Sample JCL for ACJBARD (continued)

```
/*
 */
/* ENTRYTYPE - ENTRY TYPE OF DATA SET
 */
/* EXPIRE | EXPIREDT - DATA SET EXPIRATION DATE
 */
/* LASTREF | LASTREFDT - DATA SET LAST REFERENCE DATE
 */
/* LRECL - RECORD LENGTH
 */
/* MGMTCLAS | MC - MANAGEMENT CLASS FOR DATA SET
 */
/* MULTVOL - IS THE DATA SET MULTI-VOLUME?
 */
/* NUMEXT | EXTNUM - NUMBER OF EXTENTS
 */
/* PAGELENGTH - NUMBER OF LINES PER PAGE (DEF IS 60)*
 */
/* PDSE - IS THE DATA SET A PDSE?
 */
/* REBLOCK | REBLK - IS THE DATA SET REBLOCKABLE?
 */
/* RECFM | RECFMT - RECORD FORMAT
 */
/* SMS | MANAGED - IS THE DATA SET MANAGED?
 */
/* STORCLAS | SC - STORAGE CLASS FOR DATA SET
 */
/* STORGRP | SG - STORAGE GROUP FOR DATA SET
 */
/* TITLE=XXXXX - TITLE FOR REPORT
 */
/* TOTALS - PRINT DSN SPACE TOTALS
 */
/* USED% - USED SPACE %
 */
/* VOLSEQ - VOLUME SEQUENCE NUMBER
 */
/* VOLSER - VOLUME SERIAL
 */
/* VVRCHK | VVR - IF SMS, IS THERE A VVR OR NVR? *
 */
/*
 */
/* GENREP EXEC ACJBBAOB,PLIB1=SYS1.DGTLIB,TABL2=userid.TEST.ISPTABL
 */
/* DCOLIN DD DSN=IBMUSER.DCOLLECT.DATA,DISP=SHR
 */
/* ISPFIL DD DSN=IBMUSER.DATASET.REPORT,DISP=OLD
 */
/* SYSTSIN DD *
 */
PROFILE PREFIX(IBMUSER)
ISPSTART CMD(ACBQBAR7) +
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(9999999)
/*
 */
/* SYSIN DD *
 */
DSN
VOLSER
ALLOCSP
RECFM
BLKUNUSED
TITLE=DATA SET REPORT FROM DCOLLECT DATA - 06/01/96
TOTALS
BLKSIZE
EXTNUM
DSORG
STORCLAS
```

```
/*
//*****COPY THE DATA SET REPORT*****
//*
//*      COPY THE DATA SET REPORT
//*
//*          SYSUT1 - INPUT (FROM PREVIOUS STEP)
//*          SYSUT2 - OUTPUT
//*
//*****REPGEN EXEC PGM=IEBGENER
//SYSUT1   DD  DSN=IBMUSER.DATASET.REPORT,DISP=SHR
//SYSUT2   DD  SYSOUT=*
//SYSIN    DD  DUMMY
//SYSPRINT DD  SYSOUT=*
//*****
```

Parent topic:Generate data set report from DCOLLECT data: ACBQBAR7



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Generate model commands from ISMF-saved list: ACBQBAM1

z/OS DFSMSdfp Storage Administration
SC23-6860-01

SYS1.SACBCNTL member ACBJBAM1 calls ACBQBAM1 to create model commands from a saved ISMF list.

See [Figure 1](#) for the sample JCL and parameters.

[Sample JCL for generating model commands from ISMF-saved list](#)
Parent topic:[Performing storage administration tasks in batch](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample JCL for generating model commands from ISMF-saved list

z/OS DFSMSdfp Storage Administration
SC23-6860-01

Figure 1. Sample JCL for

ACBJBAM1 //*****

```
//**  
//** SAMPLE JCL TO GENERATE MODEL COMMANDS FROM A SAVED ISMF DATA *  
//** SET LIST IN BATCH *  
//** *  
//** INSTRUCTIONS BEFORE SUBMITTING: *  
//** *  
//** CHANGE JOBCARD *  
//** CHANGE PREFIX *  
//** CHANGE PARAMETERS *  
//** *  
//** PARAMETERS: *  
//** *  
//** PARAMETER FOLLOWING ACBQBAM1 - SAVED ISMF LIST NAME *  
//** DATA SET OR DASD VOLUME LIST *  
//** REST OF THE PARAMETERS - MODEL COMMAND FOR GENERATION *  
//** *  
//*****  
//GENCMDS EXEC ACBJBAOB,PLIB1=SYS1.DGTPLIB,TABL2=userid.TEST.ISPTABL  
//SYSTSIN DD *  
PROFILE PREFIX(IBMUSER)  
ISPSTART CMD(%ACBQBAM1 DSNLIST +  
DD DSN=/,VOL=SER=@ +  
) +  
BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(9999999)  
/*  
//*
```

Parent topic: Generate model commands from ISMF-saved list: ACBQBAM1



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Creating SMS online reports

z/OS DFSMSdfp Storage Administration
SC23-6860-01

With NaviQuest you can create a report of the contents of an SMS configuration, or the planned changes to a configuration, in the form of a sequential data set. The report can be printed using your choice of tool or utility.

The following reports can be generated: Customized data set reports from ISMF saved data set listCustomized volume reports from ISMF saved DASD volume listCustomized volume reports from ISMF saved tape volume listCustomized data set reports from DCOLLECT dataCustomized volume reports from DCOLLECT dataSMS configuration reports from DCOLLECT data

Note: These reports can also be generated in batch. [Performing storage administration tasks in batch](#), provides the CLISTS or EXECs required to run these reports in batch.

Before you can generate any of these reports, you must already have created either DCOLLECT data or an ISMF table. After creating your base data, you can select the fields that you want included in the report and the order you want them presented.

For example, if you want the data set, percentage used, and expired date fields printed in that particular order, you would enter a value of 1 in the data set field, a value of 2 in the percentage used field, and a value of 3 in the expired date field. You may number as many or as few fields as you need. To generate SMS reports, choose option 5 from the NaviQuest Primary Option Menu.

For more information on generating saved lists from ISMF tables, refer to [z/OS DFSMS Using the Interactive Storage Management Facility](#).

[Creating data set reports from saved ISMF lists](#)

[Creating volume reports from saved ISMF lists](#)

[Creating customized tape reports from saved ISMF lists](#)

[Data set report from DCOLLECT data](#)

[DASD volume report from DCOLLECT data](#)

[SMS configuration report from DCOLLECT data](#)

Parent topic:[Using NaviQuest](#)



Creating data set reports from saved ISMF lists

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To generate data set reports, use the "Data Set Report From Saved ISMF List Entry Panel" panel (option 11.5.1). This panel requires you to set up the parameters of the data sets and to define which fields you want in your report.

From the "SMS Report Generation Selection Menu" panel (option 11.5), choose "Data Set Report from Saved ISMF List" (option 1). After choosing option 1, a panel appears from which you can choose from a selection of items that you want to appear in your report.

Fill in the top portion of the next screen (Data Set Report From Saved ISMF List Entry Panel) with the name of the data set that you want to hold the output report, whether you want to replace the existing data in the output data set, the name of the previously saved table, number of lines per page you want in the report, and whether you want totals for the allocated and used space attributes. Note: The maximum allowed size for reports is 133 columns and for data set names is 44 bytes. If you select the data set name to appear in the report, you can code the Max Length of the DSN Print field to free some of the 44 bytes. Coding the minimum number of 11 frees up 33 extra bytes of report columns; however, only the first 11 bytes of the data set name will be included in the report.

Fill in as many fields as you want in the order you want them printed for your report. Note: Use PF7 (up) and PF8 (down) to scroll backward or forward.

For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the data set name, the amount of allocated space, the amount of allocated space actually used, block size, and the optimal size.

After you have filled in all the information on the first panel, press PF8 to scroll forward. Fill in the remaining required information. To submit the report for processing, press the Enter key. This creates a saved data set that can be browsed.

Parent topic:[Creating SMS online reports](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Creating volume reports from saved ISMF lists

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To generate DASD volume reports, use "DASD Volume Report from Saved ISMF List." (option 11.5.2).

From the "SMS Report Generation Selection Menu" panel (option 11.5), choose "DASD Volume Report from Saved ISMF List." (option 2).

Fill in the data set name, table member name, page length, and totals. Indicate whether you want to replace the specified data set if it already exists. If N is specified and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.

Select as many fields as you want in the report, using numbers to indicate the order in which you want them printed. For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the amount of free space, the fragmentation index, and the device number, in that order.

After you have filled in all the information, press the Enter key. This creates a saved data set that can be browsed.

Parent topic:[Creating SMS online reports](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Creating customized tape reports from saved ISMF lists

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To generate tape reports, use option 11.5.3:

From the "SMS Report Generation Selection Menu" panel (option 11.5), choose "Tape Volume Report from Saved ISMF List" (option 3).

Fill in the data set name, table member name, and page length. Indicate whether you want to replace the specified data set if it already exists. If N is specified and the data set name already exists, an error message will be returned. If Y is specified, the data set will be deleted, a new data set with the same name will be allocated, and the report will be written to this data set.

Select as many fields as you want in the report, using numbers to indicate the order in which you want them printed. For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the use attribute, the volume error status, the media type, and the shelf location.

After you have filled in all the information, press the Enter key. This creates a saved data set that can be browsed.

Parent topic:[Creating SMS online reports](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Data set report from DCOLLECT data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To generate data set reports from DCOLLECT data, use option 11.5.4. Select the fields that are to be included in the report with a number to indicate the order they are to be printed.

The filled in fields of the following sample panel indicate that the user wants a report that prints the data set name, the amount of allocated space, the percentage used, the data set organization, and the volume serial number, in that order.
Note: Use PF7 (up) to scroll backward and PF8 (down) to scroll forward. Press the Enter key to generate the report.

Parent topic:[Creating SMS online reports](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

DASD volume report from DCOLLECT data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To generate volume reports from DCOLLECT data, use option 11.5.5. Select the fields that are to be included in the report with a number to indicate the order they are to be printed.

For example, the filled in fields of the following sample panel indicate that the user wants a report that prints the volume serial number, the index status, the use attribute, and the number of free extents.

Parent topic:[Creating SMS online reports](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

SMS configuration report from DCOLLECT data

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To generate SMS configuration reports from DCOLLECT data, use option 11.5.6.
Select the fields that are to be included in the report with either Yes or No (Y or N).
Yinclude this record in the report**N**do not include this record in the report

Parent topic:[Creating SMS online reports](#)

|

©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Additional storage administration functions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The following are additional storage administration functions: QSAVE and QRETRIEV ISMF commands. The QSAVE and QRETRIEV ISMF commands let you save a "query" of frequently used parameters under ISMF. You can then retrieve the parameters by their query names. The QSAVE and QRETRIEV ISMF commands can be used while running the ISMF data set list or volume list options, interactively or in batch.

Model Command Generator (ACBQFLM1 EXEC for saved ISMF lists and ACBQADM2 EXEC for DCOLLECT data) The Model Command Generator option creates a "model" command against each entry in a data set saved ISMF list, a saved ISMF volume list, or from DCOLLECT data. NaviQuest does the symbolic substitution.

COPYFILT macro

Using the COPYFILT macro you can create synchronized filter lists (FILTLISTS) that can be applied across all ACS routines. Create a FILTLIST member in your ACS routine source data set. Make all filter list updates there. Then call the COPYFILT macro from the command line to have the changes replicated across all of the ACS routines.

[QSAVE and QRETRIEV commands: saving and retrieving ISMF selection criteria](#)

[Creating model commands](#)

[COPYFILT macro: COPYLIB facility for FILTLISTS](#)

Parent topic:[Using NaviQuest](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

QSAVE and QRETRIEV commands: saving and retrieving ISMF selection criteria

z/OS DFSMSdfp Storage Administration
SC23-6860-01

The QSAVE function saves all the variables of an ISMF query used for a particular run. To use the same variables in a later run, call QRETRIEV from the "Data Set Selection Entry Panel" or the "Volume Selection Entry Panel". The QRETRIEV function retrieves all the values used in the first run and reuses them for a batch or an interactive session.

[How to use the QSAVE and QRETRIEV commands](#)
Parent topic:[Additional storage administration functions](#)

 | 

©Copyright IBM Corporation 1990, 2014

How to use the QSAVE and QRETRIEV commands

z/OS DFSMSdfp Storage Administration
SC23-6860-01

When a query is saved, a new entry is created in an ISPF table. ISMFQDSN is the table name for data set queries. ISMFQVOL is the table name for volume queries. If the tables do not exist when the first query is saved, the table is created into the library pointed to by DDNAME ISPTABL.

Duplicate query names within the ISMFQDSN and ISMFQVOL tables are not allowed. An error message indicating that there is a duplicate query name will be received if an attempt is made to save a query that already exists. An attempt to retrieve a query that does not exist, will receive an error message indicating that no query has been found.

To use either command, perform the following steps:

Create and then save a query. The user has two options for creating a query:

Create the query while running ISMF in batch. Specify the QSAVE(*name*) parameter in the JCL for the batch job. Where *name* is the query name to be saved.

Generate the QUERY online. To do this, fill in all desired parameters for the query from either ISMF option 1 (data set list option) or option 2 (volume list option). Then return to the first page of the selection panels.

To save the query, enter the query name on the field marked either QUERY NAME TO SAVE or RETRIEVE. Go to the top of the panel and enter QSAVE. You will receive messages indicating that the query name has been successfully saved.

The following sample panel, "Data Set Selection Entry Panel", is the first panel with fields filled in.

After the query is saved, retrieve the query name by specifying the QUERY(*name*) parameter in a batch job or by filling in the QUERY NAME TO SAVE OR RETRIEVE field on the data set or volume option panel of ISMF. Then issue the QRETRIEV command at the top of the panel.

Parent topic:[QSAVE and QRETRIEV commands: saving and retrieving ISMF selection criteria](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Creating model commands

z/OS DFSMSdfp Storage Administration
SC23-6860-01

You can create a user-specified model command to build a set of commands for each data set or volume in the DCOLLECT data or saved ISMF list. The ACBQFLM1 EXEC uses a saved ISMF list and the ACBQADM2 EXEC uses DCOLLECT data.

[How to create model commands](#)

Parent topic:[Additional storage administration functions](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

How to create model commands

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To create DFSMSdss commands that delete selected data sets:
Use ISMF to build an ISMF table of the data sets you want deleted.

Use the model command facility to build your DSS commands.

Specify both the ISMF table and the DSS model command you want applied to all the data sets in the list. To change the management class assigned to a group of data sets of a given high level qualifier whose user requirements have changed (high level qualifier ABC to management class MCSTAND):
Use ISMF to create a list of data sets that meet the criteria you want.

Tailor the list to remove any data sets that you do not want processed by the model command function.

Save the ISMF list into a saved table. *Example of Saved Data Set*

List:
ABC.BACKUP.LAB.REVIEW
ABC.BACKUP.LAB.SCRIPT
ABC.BACKUP.LAB.JCL

Note: In steps 1 through 3 of this example, an ISMF-saved list is used as input to generate model commands. The remaining steps (4 through 8) apply to either ISMF-saved lists or DCOLLECT records as model command input.

Choose option 11.6. The ISMF " Model Commands Generation Selection Menu" panel appears.

Select either option 1 or 2 (11.6.1 or 11.6.2).

Fill in the appropriate fields: If you selected option 1 (ACBQFLM1 EXEC), enter the following information in the " Model Commands Generation from Saved ISMF List Entry Panel": In the first field, the name of the table just savedIn the second field, ALTER / MANAGEMENTCLAS(newmgmtclas)*Example of Model Command:* ALTER / MGMTCLAS(MCSTAND)

You can also generate model commands with DCOLLECT data as input, instead of using saved lists. To do this, select option 2 (11.6.2) (ACBQADM2 EXEC) and enter the following information in the "Model Commands Generation from DCOLLECT Data Entry Panel": In the first field, the name of the DCOLLECT data setIn the second field, ALTER / MANAGEMENTCLAS(newmgmtclas)The model command function uses the same conventions as ISPF option 3.4. That is, a slash (/) may be located in the command wherever the user wants the data set name substituted. An "at" sign (@) may be used in the command wherever the user wants the volume serial number, which a data set resides on substituted. However, only two slashes and one @ may be used in each model command.

Attention: There is no syntax checking performed for the command entered by the user. It is the user's responsibility to correctly enter the command to be created against the entries in the table or DCOLLECT data.

Example Output from Command Generation: ALTER ABC.BACKUP.LAB.REVIEW MGMTCLAS(MCSTAND)
ALTER ABC.BACKUP.LAB.SCRIPT MGMTCLAS(MCSTAND)
ALTER ABC.BACKUP.LAB.JCL MGMTCLAS(MCSTAND)

This output will be created in a sequential data set with the name USERID.Mxx.MODELCMD, where xx varies from 1 to 99.

Add JCL to the commands to complete the job. When the command generation completes, you are placed in ISPF "edit." *Example Job:* //ALTER JOB ----

```
//STEP1 EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN   DD *
      ALTER ABC.BACKUP.LAB.REVIEW MGMTCLAS(MCSTAND)
```

```
ALTER ABC.BACK.LAB.SCRIPT MGMTCLAS(MCSTAND)
ALTER ABC.BACK.LAB.JCL MGMTCLAS(MCSTAND)
```

/*
Note: You can also use the SYSIN DD statement to reference the sequential data set that contains the output commands instead of the SYSIN statement followed by commands as shown above.

Submit the job for execution. The model command facility can also be used with volume lists from ISPF tables or DCOLLECT data. The following example using volume lists creates a job to DEFrag all volumes in an ISMF volume list:

*Example of Saved List:*SCR001 -----

```
SCR002 -----
SCR003 -----
SCR004 -----
SDV000 -----
SDV001 -----
SHR200 -----
```

*Example of Model Command:*DEFrag DYNAM (@)

*Example Output Command Generation:*DEFrag DYNAM(SCR001)

```
DEFrag DYNAM(SCR002)
DEFrag DYNAM(SCR003)
DEFrag DYNAM(SCR004)
DEFrag DYNAM(SDV000)
DEFrag DYNAM(SDV001)
DEFrag DYNAM(SHR200)
```

Example Job://DEFrag JOB ----
//STEP1 EXEC PGM=ADRDSSU, PARM='TRACE=YES'
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
DEFrag DYNAM(SCR001)
DEFrag DYNAM(SCR002)
DEFrag DYNAM(SCR003)
DEFrag DYNAM(SCR004)
DEFrag DYNAM(SDV000)
DEFrag DYNAM(SDV001)
DEFrag DYNAM(SHR200)
/*

Note: During the model command generation, only the first two slashes (//) and the first "at" sign (@) will be substituted.

Parent topic:[Creating model commands](#)



©Copyright IBM Corporation 1990, 2014

[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

COPYFILT macro: COPYLIB facility for FILTLISTs

z/OS DFSMSdfp Storage Administration
SC23-6860-01

As you proceed in the SMS implementation, data subtypes are gradually moved to SMS management. Often, the only difference between adding one data subtype and the next is adding data to a filter list (FILTLIST) or creating a new FILTLIST. In addition, FILTLISTS must often be added or updated in multiple routines. The COPYFILT facility simplifies this process. COPYFILT is a COPYLIB facility for filter lists contained in ACS routines.

[How to set up the COPYFILT macro](#)

[How to use the COPYFILT macro after initial setup](#)

Parent topic:[Additional storage administration functions](#)



©Copyright IBM Corporation 1990, 2014

How to set up the COPYFILT macro

z/OS DFSMSdfp Storage Administration
SC23-6860-01

This section documents intended Programming Interfaces that allow you to write programs to obtain the services of SMS.

To initially setup the COPYFILT macro, the user must perform the following steps:
Create one PDS containing all the related ACS source routines for a single SMS configuration.

Group all the FILTLISTS in an ACS routine together in one contiguous section of ACS code.

Create a new member containing a consolidated list of all FILTLISTS in all related ACS routines.

Before using this macro for the first time, some initial "priming" of the FILTLLIST member must be performed. This step need only be done *once*, after that, the process is automatic. The required priming has the following format: The line just before the first FILTLLIST must have data (beginning in column 1) of /* * * * * * * * * * */ All FILTLLISTS for all ACS routines follow the line of special character strings. The line after the last FILTLLIST must have data (beginning in column 1) of /* * * * * * * * * */

Make sure each ACS routine using the macro (the data class, storage class, management class, and the storage group routines) has the text strings listed above placed in each of the ACS routines at the location you want the COPYFILT macro to place the consolidated filter list. The COPYFILT process copies the FILTLLISTS from the FILTLLIST member and places them between these two lines of special character strings.

Parent topic:[COPYFILT macro: COPYLIB facility for FILTLLISTS](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

How to use the COPYFILT macro after initial setup

z/OS DFSMSdfp Storage Administration
SC23-6860-01

To use the COPYFILT macro *after* you have completed the initial setup, perform the following steps:

Change the FILTLIST member that is used by all four ACS routines. To do this, edit the member containing the FILTLISTS. At the command line, save the member containing the FILTLISTS with the SAVE command.

Enter COPYFILT at the command line. The user must be in edit mode (that is, ISPF option 2; you cannot use ISPF option 3.4) for a member of the ACS source data set for this facility to work correctly. The panel below appears, and the user is then prompted for the following information: Member names of the data class, storage class, management class, and storage group ACS routines inside the data setThe member name containing those FILTLISTSA brief description, for change control, of the change made to the FILTLISTS

Press the Enter key. The four routines inside the data set are updated automatically, and a message indicates that the update has taken place. If an error occurs, an error message indicates what the problem is. For example, the fields in the following panel are filled in with typical information:

Figure 1. Copy FILTLISTS Entry Panel Panel Help

```
COPY FILTLISTS ENTRY PANEL
Command ===>
To copy FILTLISTS, specify the following information and press Enter:
Member Containing FILTLISTS . . . .
Data Class ACS Routine Name . . . DATACLAS
Storage Class ACS Routine Name . . STORCLAS
Management Class ACS Routine Name . MGMTCLAS
Storage Group ACS Routine Name . . STORGRP_
Change Log Entry . . Migrating DB2 tables_____

```

Use HELP Command for Help; Use END Command to Exit.

Parent topic:[COPYFILT macro: COPYLIB facility for FILTLISTS](#)



[Previous topic](#) | [Next topic](#) | [Contents](#) | [Contact z/OS](#) | [Library](#) | [PDF](#)

Sample Batch Job for CICS Definitions

z/OS DFSMSdfp Storage Administration
SC23-6860-01

In order to utilize the DFSMSdss facilities to process data sets that are open to CICS TS files, a number of definitions and parameters in CICS TS need to be defined. The sample batch job shown below defines these to CICS.

When a storage administrator indicates with SMS policies that CICS file system interfaces should be called if DFSMSdss can not obtain serialization to process the request, DFSMSdss will attempt to use the CICS file system interfaces to first close the files open to data sets in CICS and then to process the DFSMSdss request. Subsequently, these interfaces will be called to allow these files to be opened by CICS.

```
//DEFCSD JOB ,JBCAMMA,MSGCLASS=H,CLASS=A,  
// MSGLEVEL=(1,1),REGION=0M,NOTIFY=&SYSUID,TIME=1449  
//*****  
// * CONFIGURING THE CICS TS FOR DFSMSDSS TO CLOSE AND OPEN FILES:  
//  
// * YOU NEED TO SET A NUMBER OF DEFINITIONS AND PARAMETERS IN  
// * CICS TO ENABLE THE DFSMSDSS CICS SERVER CODE TO PROCESS  
// * REQUESTS TO CLOSE OR OPEN FILES.  
//  
// * TO ADD THE CICS RESOURCE DEFINITIONS THAT ARE REQUIRED  
// * BY DFSMSDSS YOU NEED TO RUN THE RDO COMMANDS IN THE EXAMPLE BELOW.  
//  
// * THIS EXAMPLE CREATES TWO RESOURCE GROUPS, DSSCICS AND DSSEXCI.  
// * BEFORE RUNNING THE JOB YOU NEED TO EDIT IT TO CHANGE THE DEFAULT  
// * LIBRARY VALUES TO MATCH THOSE OF YOUR INSTALLATION:  
//  
// * + GROUP DSSCICS CONTAINS THE DEFINITIONS FOR THE DFSMSDSS CICS  
// * SERVER PROGRAM. YOU NEED TO ADD THIS GROUP TO THE GROUP LIST  
// * THAT IS USED AT CICS STARTUP.  
//  
// * + DSSEXCI CONTAINS DEFINITIONS FOR A EXTERNAL CICS INTERFACE  
// * (EXCI) GENERIC CONNECTION BECAUSE DFSMSDSS USES THE EXCI.  
// * YOU NEED TO DEFINE THE DSSEXCI GROUP IF YOU DO NOT ALREADY  
// * HAVE A EXCI GENERIC CONNECTION DEFINED.  
//*****  
//CONSDEF EXEC PGM=DFHCSDUP  
//STEPLIB DD DSN=CICSTS.V4R1.PROD.CICS.SDFHLOAD,DISP=SHR  
//DFHCSD DD DSN=USRLCL.V4R1.DFHCSD,DISP=SHR  
//SYSPRINT DD SYSOUT=*  
DELETE LIST(MYLIST)  
DELETE GROUP(DSSTEST)  
DELETE GROUP(DSSCICS)  
DELETE GROUP(DSSEXCI)  
  
APPEND LIST(DFHLIST) TO(MYLIST)  
ADD GROUP(DSSCICS) LIST(MYLIST)  
ADD GROUP(DSSEXCI) LIST(MYLIST)  
  
DELETE TERM(MAST) GROUP(DSSCICS)  
DEFINE TERM(MAST) GROUP(DSSCICS) TYPETERM(DFHCONS)  
CONSNAMES(MASTER1) DESCRIPTION(MVS CONSOLE CONS500)  
  
DELETE TERM(TST1) GROUP(DSSCICS)  
DEFINE TERM(TST1) GROUP(DSSCICS) TYPETERM(DFHCONS)  
CONSNAMES(LOCALC02) DESCRIPTION(MVS CONSOLE LOCALC02)  
  
DELETE TRANSACTION(DSSX) GROUP(DSSCICS)  
DEFINE TRANSACTION(DSSX) GROUP(DSSCICS) PROGRAM(DFHMIRS)  
DESCRIPTION(DSS SERVER TRANSACTION REQUIRED BY DFSMSDSS)  
PROFILE(DFHCICSA)  
  
DELETE PROGRAM(ADREXCIS) GROUP(DSSCICS)  
DEFINE PROGRAM(ADREXCIS) GROUP(DSSCICS) LANGUAGE(ASSEMBLER)  
DESCRIPTION(CICS SERVER PROGRAM REQUIRED BY DFSMSDSS)  
DATALOCATION(ANY) EXECKEY(USER)  
  
DELETE CONNECTION(DSSG) GROUP(DSSEXCI)  
DEFINE CONNECTION(DSSG) GROUP(DSSEXCI)  
DESCRIPTION(EXCI GENERIC CONNECTION REQUIRED BY DFSMSDSS)  
ACCESSIONMETHOD(IRC) SINGLESESS(NO)  
PROTOCOL(EXCI) COUNTTYPE(GENERIC)  
DATASTREAM(USER) RECORDFORMAT(U) AUTOCONNECT(NO)  
INSERVICE(YES) ATTACHSEC(IDENTIFY)  
  
DELETE SESSIONS(DSSG) GROUP(DSSEXCI)  
DEFINE SESSIONS(DSSG) GROUP(DSSEXCI)  
DESCRIPTION(EXCI GENERIC SESSIONS DEFINITION REQUIRED BY DFSMSDSS)  
CONNECTION(DSSG) PROTOCOL(EXCI) MAXIMUM(0,0)  
RECEIVEPFX(RG) RECEIVECOUNT(10)  
SENDSIZE(4096) RECEIVESIZE(4096)  
SESSPRIORITY(0) AUTOCONNECT(NO)  
BUILDCHAIN(YES) IOAREALEN(4096,4096) RELREQ(NO)  
DISCREQ(NO) NEPCLASS(0)  
RECOVOPTION(SYSDEFAULT)  
  
LIST LIST(MYLIST) OBJECTS
```



©Copyright IBM Corporation 1990, 2014