

OS/390



# DFSMSdss Storage Administration Guide

*Release 10*



OS/390



# DFSMSdss Storage Administration Guide

*Release 10*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 119.

**First Edition (September 2000)**

This edition applies to Version 2 Release 10 of OS/390 (5647-A01), and any later releases until otherwise indicated in new editions. This edition replaces and makes obsolete the previous edition, *DFSMS/MVS Version 1 Release 5 DFSMSdss Storage Administration Guide* (SC26-4930-04).

© **Copyright International Business Machines Corporation 1984, 2000. All rights reserved.**

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

---

# Contents

<b>Preface</b> . . . . .	ix
Required Product Knowledge . . . . .	ix
Required Books . . . . .	ix
Related Books . . . . .	ix
DFSMS Publications . . . . .	ix
Storage Management Library . . . . .	x
MVS . . . . .	x
Other Related Publications . . . . .	x
Accessing OS/390 Books on the Internet . . . . .	x
How to Send Your Comments . . . . .	x
<b>Summary of Changes</b> . . . . .	xiii
Summary of Changes for SC35-0393-00 OS/390 DFSMSdss Storage Administration Guide . . . . .	xiii
Summary of Changes for SC26-4930-04 DFSMS/MVS Version 1 Release 5 DFSMSdss Storage Administration Guide . . . . .	xiii
Summary of Changes for SC26-4930-03 Version 1 Release 4 DFSMSdss Storage Administration Guide . . . . .	xiii
<b>Chapter 1. Introduction to the DFSMSdss Component of DFSMS</b> . . . . .	1
Stand-Alone Restore Program of DFSMSdss . . . . .	1
Understanding the Role of DFSMSdss . . . . .	1
Managing User Data with SMS . . . . .	1
Sequential Data Striping . . . . .	2
Record Counting . . . . .	3
Installation Exit Routines . . . . .	3
Authorization Checking . . . . .	3
Managing Availability with DFSMSdss . . . . .	3
Backing Up and Restoring Volumes and Data Sets . . . . .	4
Using DFSMSHsm for Backup . . . . .	4
Concurrent Copy . . . . .	4
Managing Data Movement with DFSMSdss . . . . .	6
Moving Data . . . . .	6
Moving Data in an SMS-Managed Environment . . . . .	6
Moving Data with Concurrent Copy . . . . .	6
Moving Data with FlashCopy . . . . .	7
Moving Data with Snapshot . . . . .	7
Converting Data to and from SMS Management . . . . .	7
Conversion of Data Sets with Data Movement . . . . .	7
Conversion of Volumes without Data Movement . . . . .	7
Managing Space with DFSMSdss . . . . .	8
<b>Chapter 2. Requirements for Running DFSMSdss</b> . . . . .	9
Operating Environment . . . . .	9
Storage Requirements . . . . .	9
Hardware Requirements . . . . .	11
Stand-Alone Restore Program for DFSMSdss . . . . .	11
Volume Formats . . . . .	11
Data Set Organizations . . . . .	12
Temporary Data Set Names . . . . .	12
<b>Chapter 3. Logical and Physical Processing and Data Set Filtering</b> . . . . .	15
Logical and Physical Processing . . . . .	15

Logical Processing . . . . .	15
Physical Processing . . . . .	16
Data Integrity Considerations . . . . .	17
Choosing Data Sets for Processing—Filtering . . . . .	18
Filtering by Data Set Names . . . . .	18
Filtering by Data Set Characteristics . . . . .	19
The FILTERDD Keyword . . . . .	20
Some Uses of Filtering . . . . .	20
<b>Chapter 4. Invoking DFSMSdss . . . . .</b>	<b>23</b>
Invoking DFSMSdss with ISMF . . . . .	23
How to Invoke ISMF . . . . .	23
Invoking DFSMSdss with JCL . . . . .	23
Invoking DFSMSdss with the Application Interface . . . . .	23
User Interaction Module Exit Functions . . . . .	24
<b>Chapter 5. Protecting DFSMSdss Functions . . . . .</b>	<b>25</b>
Protecting DFSMSdss/ISMF Functions with the OS/390 Security Server RACF Element . . . . .	25
ISMF Functions You Might Want to Protect . . . . .	25
Setting Up the Authorization Structure . . . . .	25
Protecting DFSMSdss Keywords with RACF . . . . .	27
<b>Chapter 6. Managing Availability with DFSMSdss . . . . .</b>	<b>29</b>
Planning an Availability Strategy . . . . .	29
Backup and Recovery . . . . .	29
Disaster Recovery . . . . .	30
Vital Records . . . . .	32
Archiving . . . . .	32
Backing Up Data Sets . . . . .	33
Logical Data Set Dump . . . . .	34
Physical Data Set Dump . . . . .	34
Backup with Concurrent Copy . . . . .	35
A Backup Scenario . . . . .	36
Backing Up Data Sets with Special Requirements . . . . .	37
Dumping HFS Data Sets . . . . .	37
Dumping Multivolume Data Sets . . . . .	37
Dumping Integrated Catalog Facility User Catalogs . . . . .	39
Dumping Non-VSAM Data Sets That Have Aliases . . . . .	39
Dumping VSAM Spheres . . . . .	40
Dumping Indexed VSAM Data Sets . . . . .	40
Dumping SYS1 System Data Sets . . . . .	40
Dumping Data Sets Containing Records Past the Last-Used-Block Pointer . . . . .	40
Backing Up SMS-Managed Data Sets . . . . .	41
Backing Up Volumes . . . . .	42
Logical Volume DUMP . . . . .	42
Physical Volume Dump . . . . .	42
Backing up System Volumes . . . . .	42
Backing up VM-Format Volumes . . . . .	42
Dumping Data Efficiently . . . . .	43
Space Considerations . . . . .	43
Performance Considerations . . . . .	44
Shared DASD Considerations . . . . .	47
Restoring Data Sets . . . . .	48
Logical Data Set Restore . . . . .	48
Expiration Date Handling during Logical Restore . . . . .	51

Physical Data Set Restore . . . . .	51
Restoring Data Sets with Special Requirements . . . . .	53
Restoring Multivolume Data Sets and Restoring Data Sets Using Multiple Target Volumes (Spill Volumes) . . . . .	53
Restoring Integrated Catalog Facility Catalogs . . . . .	54
Restoring Non-VSAM Data Sets That Have Aliases . . . . .	55
Restoring Indexed Sequential, Unmovable, Direct, and Absolute Track Data Sets . . . . .	55
Restoring an Undefined DSORG Data Set . . . . .	57
Restoring a VSAM Sphere . . . . .	57
Restoring a Preallocated VSAM Cluster . . . . .	58
Restoring the VVDS and the VTOCIX . . . . .	58
Restoring a PDSE . . . . .	58
Restoring a Damaged PDS . . . . .	59
Restoring Data Sets in an SMS-Managed Environment . . . . .	59
Converting Non-VSAM Data Sets to Multivolume . . . . .	60
Restoring SMS-Managed Data Sets . . . . .	60
Restoring GDG Data Sets . . . . .	63
Restoring Non-SMS-Managed Data Sets . . . . .	64
Logical Restore of Data Sets with Phantom Catalog Entries . . . . .	64
Restoring Volumes . . . . .	65
Specifying Output Volumes . . . . .	66
Recovering VM-Format Volumes . . . . .	67
<b>Chapter 7. Managing Data Movement with DFSMSdss . . . . .</b>	<b>69</b>
Preparing for Data Movement . . . . .	69
Evaluating the Use of Logical and Physical Copy . . . . .	69
Controlling What DFSMSdss Copies . . . . .	70
Moving Data Sets . . . . .	70
Specifying Input Volumes . . . . .	70
Selecting Output Volumes . . . . .	71
Renaming Data Sets . . . . .	72
Expiration Date Handling . . . . .	73
Defining RACF Profiles . . . . .	74
Moving Data Sets with Utilities . . . . .	74
Moving Data Sets with Concurrent Copy . . . . .	75
Moving Data Sets with Snapshot . . . . .	76
Moving Data Sets with Special Requirements . . . . .	77
Moving Undefined DSORG and Empty non-VSAM Data Sets . . . . .	77
Moving System Data Sets . . . . .	78
Moving Catalogs . . . . .	78
Moving Non-VSAM Data Sets That Have Aliases . . . . .	79
Moving Multivolume Data Sets . . . . .	79
Converting VSAM and Non-VSAM Data Sets to Multivolume . . . . .	80
Moving VSAM Data Sets . . . . .	80
Moving a PDSE . . . . .	82
Moving a Damaged PDS . . . . .	82
Moving Unmovable Data Sets . . . . .	82
Moving Data Sets to Unlike Devices . . . . .	83
Moving Indexed Sequential Data Sets . . . . .	83
Moving Direct Access Data Sets . . . . .	83
Moving GDG Data Sets . . . . .	84
Moving SMS-Managed Data Sets . . . . .	85
Moving Non-SMS-Managed Data Sets . . . . .	88
Moving to Preallocated Data Sets . . . . .	88
Moving Volumes . . . . .	90

Logical Volume Copy Operation . . . . .	90
Physical Volume Copy Operation . . . . .	91
Moving Volumes with FlashCopy . . . . .	92
Moving Volumes with Snapshot . . . . .	92
VTOC Considerations . . . . .	92
Moving Volumes to Like Devices of Equal Capacity . . . . .	93
Moving Volumes to Like Devices of Greater Capacity . . . . .	93
Moving Volumes to Unlike Devices . . . . .	93
Moving VM-Format Volumes . . . . .	94
<b>Chapter 8. Converting Data to and from SMS Management . . . . .</b>	<b>95</b>
Evaluating Conversion to SMS Management . . . . .	95
Data Sets Ineligible for Conversion to SMS . . . . .	95
Data Sets Ineligible for Conversion from SMS . . . . .	96
Volume Eligibility for Conversion to SMS . . . . .	96
Conversion by Data Movement . . . . .	96
Converting to SMS Management by Data Movement . . . . .	96
Conversion from SMS Management by Data Movement . . . . .	97
Conversion without Data Movement . . . . .	97
Simulating Conversion . . . . .	97
Preparing a Volume for Conversion . . . . .	98
Converting to SMS Management without Data Movement . . . . .	99
SMS Report . . . . .	100
Special Data Set Requirements for Conversion to SMS . . . . .	100
VSAM Sphere Eligibility . . . . .	100
Multivolume Data Sets . . . . .	101
GDG Data Sets . . . . .	102
Temporary Data Sets . . . . .	102
VTOC and VVDS . . . . .	102
Converting from SMS Management without Data Movement . . . . .	102
Special Data Set Requirements for Conversion from SMS . . . . .	102
Multivolume Data Sets . . . . .	102
GDG Data Sets . . . . .	103
Temporary Data Sets . . . . .	103
VTOC and VVDS . . . . .	103
Special Considerations for Using Non-SMS-Managed Targets . . . . .	103
<b>Chapter 9. Managing Space with DFSMSdss . . . . .</b>	<b>105</b>
Reclaiming DASD Space. . . . .	105
Releasing Unused Space in Data Sets. . . . .	105
Compressing a PDS . . . . .	106
Deleting Unwanted Data Sets . . . . .	106
Combining Data Set Extents . . . . .	107
Consolidating Free Space on Volumes. . . . .	108
When to Run the DEFrag Function . . . . .	109
Data Sets Excluded from DEFrag Processing . . . . .	109
DEFrag Options . . . . .	110
Serialization. . . . .	111
Security Considerations . . . . .	113
Maximizing Track Utilization by Reblocking Data Sets . . . . .	114
<b>Appendix. ACS Routine Information . . . . .</b>	<b>115</b>
ACS Variables Available during Copy Function . . . . .	115
ACS Variables Available during RESTORE and CONVERTV Processing . . . . .	116
Using SIZE and MAXSIZE Variables . . . . .	118

<b>Notices</b> . . . . .	119
Programming Interface Information . . . . .	120
Trademarks. . . . .	120
<b>Glossary</b> . . . . .	121
<b>Index</b> . . . . .	131



---

## Preface

This book describes how to use the DFSMSDss™ component of DFSMS™ to perform various storage management tasks. It is intended primarily for system programmers.

For information about the syntax of DFSMSDss commands, refer to *OS/390 DFSMSDss Storage Administration Reference*. For information about DFSMSDss messages, refer to *OS/390 MVS System Messages, Vol 1 (ABA-ASA)*.

---

## Required Product Knowledge

To use this manual you need some knowledge of DFSMSdfp™, DFSMShsm™, RACF® (a component of the SecureWay® Security Server for OS/390®), and Job Control Language (JCL).

---

## Required Books

You should be familiar with the information presented in the following books:

- *OS/390 DFSMS Introduction* (SC26-7344) introduces you to DFSMSDss and helps in evaluating its use.
- *OS/390 DFSMSDss Storage Administration Reference* (SC35-0394) describes DFSMSDss command syntax.
- *OS/390 DFSMSDss Diagnosis Guide* (LY35-0113) tells how to diagnose errors you might encounter while using DFSMSDss.
- *OS/390 DFSMS Migration*; (SC26-7329) describes the tasks required to ensure a successful installation of DFSMSDss/ISMF.
- *OS/390 DFSMS Installation Exits* (SY27-7613) describes the installation user exits available with DFSMS.
- *OS/390 DFSMS: Using the Interactive Storage Management Facility* (SC26-7340) describes how to use ISMF to manage data sets and storage.
- *OS/390 MVS System Messages, Vol 1 (ABA-ASA)* (GC28-1784) lists DFSMSDss messages.

---

## Related Books

The following books are associated with procedures described in this guide.

---

## DFSMS Publications

Publication Title	Order Number
<i>OS/390 DFSMS Macro Instructions for Data Sets</i>	SC26-7337
<i>OS/390 DFSMS: Managing Catalogs</i>	SC26-7338
<i>OS/390 DFSMSdfp Storage Administration Reference</i>	SC26-7331
<i>OS/390 DFSMShsm Storage Administration Guide</i>	SC35-0388
<i>OS/390 DFSMShsm Storage Administration Reference</i>	SC35-0389
<i>OS/390 DFSMSdfp Utilities</i>	SC26-7343

## Storage Management Library

Publication Title	Order Number
<i>OS/390 DFSMS: Implementing System-Managed Storage</i>	SC26-7336
<i>MVS/ESA Storage Management Library: Managing Data</i>	SC26-3124

## MVS

Publication Title	Order Number
<i>OS/390 MVS Programming: Assembler Services Guide</i>	GC28-1762
<i>OS/390 MVS Programming: Assembler Services Reference</i>	GC28-1910
<i>OS/390 MVS JCL Reference</i>	GC28-1757
<i>OS/390 MVS JCL User's Guide</i>	GC28-1758
<i>OS/390 MVS Planning: Global Resource Serialization</i>	GC28-1759

## Other Related Publications

Publication Title	Order Number
<i>ITSO Implementing DFSMSdss Snapshot and Virtual Concurrent Copy</i>	SG24-5268
<i>OS/390 MVS Planning: Global Resource Serialization</i>	GC28-1759
<i>OS/390 SecureWay Security Server RACF Security Administrator's Guide</i>	SC28-1915

## Accessing OS/390 Books on the Internet

In addition to making softcopy books available on CD-ROM, IBM® provides access to unlicensed OS/390 softcopy books on the Internet. To find OS/390 books on the Internet, first go to the OS/390 home page: <http://www.ibm.com/s390/os390/> From this Web site, you can link directly to the OS/390 softcopy books by selecting the Library icon. You can also link to IBM Direct to order hardcopy books.

## How to Send Your Comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other DFSMS documentation:

- Visit our home page at:  
<http://www.storage.ibm.com/software/sms/sms/home.htm>  
There you will find the feedback page where you can enter and submit your comments
- Send your comments by e-mail to:
  - IBMLink™ from US: [starpubs@us.ibm.com](mailto:starpubs@us.ibm.com)
  - IBMLink from Canada: STARPUBS at TORIBM
  - IBM Mail Exchange: USIB3VVD at IBMMAIL
  - Internet: [starpubs@us.ibm.com](mailto:starpubs@us.ibm.com)

Be sure to include the name of the book, the part number of the book, version and product name, and if applicable, the specific location of the text you are commenting on (for example, a page number or a table number).

- Fill out one of the forms at the back of this book and return it by mail or by giving it to an IBM representative. If the form has been removed, address your comments to IBM Corporation, Department 61C, 9000 South Rita Road, Tucson, Arizona 85744-0001, U.S.A.



---

## Summary of Changes

This summary of changes contains specific release updates to this book. All significant changes are marked in the text by revision bars in the left margin.

---

### Summary of Changes for SC35-0393-00 OS/390 DFSMSdss Storage Administration Guide

This book contains information previously presented in *DFSMS/MVS Version 1 Release 5 DFSMSdss Storage Administration Guide*, SC26-4930-04.

The information that has been added to this edition supports the following product enhancements:

- Dumping hierarchical file system (HFS) data sets.
- Using striped VSAM data sets.
- Using FlashCopy on an Enterprise Storage Server (ESS) subsystem to quickly move volumes from a source location to a target location.
- Using snapshot on a RAMAC Virtual Array (RVA) subsystem to quickly move data sets and volumes from a source location to a target location.
- Using the CONSOLIDATE keyword of the DEFRAg command to perform extent reduction by consolidating multiple data sets.
- Using the DEBUG=FRMSG parameter with the EXEC statement as a diagnostic aid.

Additionally, this edition includes current examples of printed output for an SMS report and for dump and restore operations. The appendix “DFSMSdss Command Specifications Not Supported by DFDSS 2.5.0” was removed, as it is no longer applicable. Editorial changes have been made, as usual, to improve the usability of this book.

---

### Summary of Changes for SC26-4930-04 DFSMS/MVS Version 1 Release 5 DFSMSdss Storage Administration Guide

Information has been added to this DFSMS/MVS Version 1 Release 5 edition to support the STORGRP keyword for data set copy, logical data set dump, and logical release. In addition, fresh examples of printed output are included, as well as technical and editorial changes.

---

### Summary of Changes for SC26-4930-03 Version 1 Release 4 DFSMSdss Storage Administration Guide

The following information has been added to update the original DFSMS/MVS Version 1 Release 4 edition in a subsequent, softcopy-only release:

- DFSMSdss can now take advantage of the snapshot function to move data. Snapshot supports data movement in either native or virtual concurrent copy modes.
- DUMP and COPY commands both support snapshot.
- VM minivolumes are supported if you are using RAMAC Virtual Array (RVA) devices to the extent that they are supported by IBM Extended Facilities Product (IXFP) device reporting.

The following information has been added with the original DFSMS/MVS Version 1 Release 4 edition:

- The new Stand-Alone Services program replaces the existing Stand-alone restore function.
- The SELECTMULTI keyword replaces the ALLMULTI keyword for the CONVERTV, COPY, and DUMP commands. Use the SELECTMULTI keyword to control selection of multivolume data sets when input volumes are specified.
- The FORCECP keyword allows you to determine whether or not to process IMS BDAM and MVS checkpoint/restart data sets with the CONVERTV, COPY, DUMP, DEFrag, RELEASE, and RESTORE commands.
- The CPVOLUME keyword supports VM-format volumes for copy, backup, and restore functions.
- The VOLCOUNT keyword allows you to better manage the allocation of SMS data sets during RESTORE and COPY.

In addition, new examples of printed output are included for an SMS report, and for dump and restore operation.

---

# Chapter 1. Introduction to the DFSMSdss Component of DFSMS

DFSMSdss is a direct access storage device (DASD) data and space management tool. DFSMSdss works on DASD volumes only in the MVS™ environment. You can use DFSMSdss to:

- Copy and move data sets between volumes of like and unlike device types<sup>1</sup>
- Dump and restore data sets, entire volumes, or specific tracks
- Convert data sets and volumes to and from SMS management
- Compress partitioned data sets
- Release unused space in data sets
- Reduce or eliminate DASD free-space fragmentation by consolidating free space on a volume

---

## Stand-Alone Restore Program of DFSMSdss

From a physical dump, you can restore:

- A full volume or ranges of tracks
- Your system residence (SYSRES) volume, if your operating system fails to IPL

Refer to *OS/390 DFSMSdss Storage Administration Reference* for more information.

---

## Understanding the Role of DFSMSdss

The role DFSMSdss plays at your site depends on what other DFSMS components you use. The way you use DFSMSdss depends on whether you also use the DFSMSshm component. To understand the role of DFSMSdss in an SMS environment, you need a basic understanding of SMS.

## Managing User Data with SMS

SMS allows you to match users' data characteristics (like data set organization, size, and format) to the characteristics of storage devices, without requiring users to know or to understand the hardware configuration at your site. With SMS, end users can store and retrieve data without being aware of space limitations, device characteristics, or volume serial numbers.

Using SMS, you can define allocation management criteria for the different types of data at your site. The values you specify identify your users' requirements for space, availability, and performance. You define these values to SMS as:

**Data Class** A named list of data set allocation attributes that SMS assigns to a data set when it is created.

**Storage Class** A named list of data set storage service attributes that identify performance and availability requirements. SMS uses these attributes to control data placement.

---

1. Like devices have the same track capacity and number of tracks per cylinder (for example, 3380 Model D, Model E, and Model K). Unlike DASD have different track capacities (for example, 3380 and 3390), a different number of tracks per cylinder, or both.

### **Management Class**

A named list of management attributes that SMS uses to control DFSMShsm action for data set retention, migration, backup, and release of allocated but unused space.

### **Storage Group**

A named list of DASD volumes used for allocation of new SMS-managed data sets or for a dummy storage group.

Automatic class selection (ACS) is the SMS mechanism for assigning SMS classes and storage groups (also known as constructs). Depending on the DFSMSdss command you are using, SMS invokes some or all of the ACS routines in the following order:

1. Storage class ACS routine
2. Management class ACS routine
3. Storage group ACS routine

SMS uses the assigned constructs to automatically place and manage data and storage. For example, you can use a storage class to keep performance-sensitive data on high-speed storage devices and use management classes to control the movement of less active data to tape.

If there are WRITE statements in the SMS ACS routines, these are only displayed in the DFSMSdss output when the ACS routines return a nonzero return code. If DFSMSdss processes a data set successfully, then no WRITE messages are displayed.

For a more complete discussion of SMS and how to use it, refer to ACS routine information found in *OS/390 DFSMSdss Storage Administration Reference*.

## **Sequential Data Striping**

Extended-sequential data sets and extended-format VSAM data sets, which must reside on SMS volumes, can be striped. Striping is a subtype of the basic record organizations: sequential and VSAM. With striping, the data is written across multiple volumes, with consecutive “loading units” being striped (applied) to different volumes. The “loading unit” for extended-sequential data sets is a track. The “loading unit” for striped VSAM data sets is a control interval (CI).

Striping can reduce the processing time for batch jobs that process large data sets sequentially. For details on how striped data sets are processed, refer to *OS/390 DFSMS: Implementing System-Managed Storage*.

DFSMSdss can dump, restore, copy, or release space from a striped data set.

### **Notes:**

1. No new keywords or commands are required to support striped data sets.
2. DFSMSdss treats a striped data set in the same way as it does other multivolume SMS data sets.
3. DFSMSdss can convert a striped extended-format VSAM data set to extended-format during RESTORE processing. DFSMSdss can convert an extended-sequential data set to sequential during RESTORE or COPY processing.

## Record Counting

DFSMSDss provides a means for verifying results of certain operations:

- **Extended-sequential data sets**—DFSMSDss performs and reports byte counting for logical data set COPY, DUMP, and RESTORE operations. Refer to *OS/390 MVS System Messages, Vol 1 (ABA-ASA)* for further information. The byte counts are reported in message ADR902I for copy, ADR903I for dump, and ADR906I for restore.
- **Indexed VSAM data sets**—DFSMSDss performs and reports record counting for logical data set DUMP and RESTORE operations if the VALIDATE support is used during the dump processing. VALIDATE processing is the default for dump. During dump processing, the record count is reported in message ADR788I. In restore processing, message ADR788I is issued if the restore record count matches the dump count. Message ADR789W is issued if the dump and restore record counts differ and both the dump and restore record counts are provided.

## Installation Exit Routines

You can customize DFSMSDss by coding exit routines. The following installation exit routines are supplied with DFSMSDss:

### **Authorization installation exit routine (ADRUPSWD)**

Forces authorization checking of protected data sets

### **Enqueue installation exit routine (ADRUENQ)**

Forces enqueueing of the VTOC

### **Options installation exit routine (ADRUIXIT)**

Can override any default or user-specified command options in the input stream

### **Reblock installation exit routine (ADRREBLK)**

Allows DFSMSDss, during a data set copy or data set restore operation, to use the block size it selects for the target data set

Refer to *OS/390 DFSMS Installation Exits* for details about these routines.

## Authorization Checking

For information on authorization checking, refer to *OS/390 DFSMSDss Storage Administration Reference*.

---

## Managing Availability with DFSMSDss

DFSMSDss availability management consists of backing up data on DASD to tape and restoring from the backup if the original is lost, damaged, or inadvertently changed.

There are two general forms of backup:

### **Data set backup**

Protects against the loss of individual data sets

### **Volume backup**

Protects against the loss of a volume

For data set backup, you can perform incremental backups to help reduce processing time while still meeting your backup requirements. Incremental backup means that data sets are backed up only if they have changed since their last backup.

Volume backups are used to protect against media failure. You can use volume backups in conjunction with incremental data set backups to recover a volume. As a result, you need not do volume backups as frequently. Incremental data set backups should be done daily and volume backups weekly. If a volume is lost for some reason, you can restore from the most recent volume backup and apply incremental data set backups to the volume to bring it back to its most current status.

## Backing Up and Restoring Volumes and Data Sets

You use the DFSMSdss DUMP command to back up volumes and data sets, and you use the RESTORE command to recover them. You can make incremental backups of your data sets by specifying a data set DUMP command with RESET and filtering on the data-set-changed flag.

In an SMS environment, DFSMSdss saves the class names for the data sets it dumps. When you restore the data set to an SMS-managed volume, DFSMSdss invokes ACS and passes it the class names saved with the data set. Based upon this and other input from DFSMSdss (for example, class names specified with the STORCLAS or MGMTCLAS keywords), ACS assigns SMS constructs to each data set.

Because DFSMSdss restore invokes ACS, you can restore the data sets to SMS-managed volumes. Conversely, data sets backed up as SMS-managed data sets can be restored as non-SMS-managed data sets.

In addition to providing for routine backup requirements, you can use DFSMSdss to back up application data for disaster recovery and vital records purposes. You can back up all the data sets<sup>2</sup> that are associated with a particular application for disaster recovery or vital records by using DFSMSdss logical data set dump, and filtering on data set names. If you do not want to perform a separate dump operation for disaster recovery, you can specify more than one OUTDDNAME to create up to 255 separate backup copies when you do your routine backup. These extra copies can then be used for disaster recovery or vital records purposes. The DUMP command can also be used to archive data sets that have not been accessed for long periods of time.

## Using DFSMSHsm for Backup

The DFSMSHsm component of DFSMS provides automated incremental backup, interactive recovery, and inventory of what it backs up. If DFSMSHsm is being used, you should use DFSMSdss for volume backup of data sets not supported by DFSMSHsm and for dumping SYSRES and special volumes such as the one containing the master catalog. If DFSMSHsm is not installed, you can use DFSMSdss for all volume and data set backups.

## Concurrent Copy

Many online data bases must be available at all times. If a backup is made while the data is being updated, the backup could be unusable or could require that a log be applied to the restored version to synchronize the data. The alternative is to synchronize all parts of the data base and stop all update activity during the backup.

---

2. This includes data residing on primary DASD only; DFSMSdss cannot be used to process migrated data sets.

The concurrent copy function of DFSMSdss is a hardware and software solution that allows you to back up a data base or any collection of data at a point in time and with minimum down time for the data base. The data base is unavailable only long enough for DFSMSdss to initialize a concurrent copy session for the data, which is a very small fraction of the time that the complete backup will take. The copy that is made will not include any of the update activity; it will be as if the backup were made instantaneously when it was requested. After initialization, DFSMSdss releases all the serialization it holds on the data, informs the user that the initialization is complete so that update activity may resume, and begins reading the data.

Be aware, however, that concurrent copy does not remove all data integrity exposures. For example, a DFSMSdss full-volume dump serializes the VTOC of the source volume, but does not serialize the data sets on the volume. This ensures that the existing data sets are not deleted or extended, and new data sets are not allocated. However, there is an exposure in that the data in the existing data sets can be changed. Without concurrent copy, this exposure exists for the entire duration of the dump. With concurrent copy, the exposure exists only during initialization.

**Notes:**

1. If you are using concurrent copy on VM-format volumes, DFSMSdss does not serialize VM data in any way.
2. VM mini volumes are supported if you are using RAMAC<sup>®</sup> Virtual Array (RVA) devices to the extent that they are supported by IBM Extended Facilities Product (IXFP) device reporting.

If a dump requestor does not stop all updating of the data sets during the concurrent copy session initialization, the backup data integrity is compromised.

If a concurrent copy (CC) operation fails after signaling that the concurrent copy initialization was complete (and update activity on the data has resumed), it is not possible to recover the data at the point-in-time at which the concurrent copy operation was started. This is because the data may have been updated while the copy operation was progressing.

**Virtual Concurrent Copy**

Virtual concurrent copy support uses snapshot to provide a concurrent copy-like function when the source device supports snapshot, but does not support concurrent copy.

During virtual concurrent copy, data is “snapped” from the source location to an intermediate location, and the data is gradually copied to the target location using normal I/O methods. The operation is logically complete after the source data is “snapped” to the intermediate location and physically complete after the data is moved to the target media.

---

## Managing Data Movement with DFSMSdss

DFSMSdss can help you move data to replace devices, add capacity, and meet performance requirements. The three general types of data movement are data set, volume, and track movement. See “Chapter 6. Managing Availability with DFSMSdss” on page 29 for more information.

### Moving Data

Using the DFSMSdss COPY command, you can perform data set, volume, and track movement. The COPY command with DELETE causes DFSMSdss to delete the source data set after it successfully copies the data set.

The full-volume COPY command is useful for moving data between like devices. If you are moving volumes to like devices of larger capacity, generally you need a larger VTOC because the larger device can hold more data sets. DFSMSdss rebuilds indexed VTOCs and recognizes larger VTOCs on target volumes (as long as the target VTOC is outside the range of the source volume) when the VTOCs are moved to a like device of larger capacity.

For moving data between unlike devices, you must use the logical data set COPY command for all the data sets on the volume. DFSMSdss fills the tracks as completely as possible instead of just copying track for track. In addition, if the reblockable indicator is set on, the data set is reblocked to a system-determined block size efficient for the device.

### Moving Data in an SMS-Managed Environment

In an SMS-managed environment, ACS routines and VTOC/Data Set Services (VDSS) determine the target volume in an SMS-managed environment.

DFSMSdss moves data sets to different volumes if their storage groups change. However, even if their storage groups do not change, DFSMSdss might move the data sets to a different location on the same volume or to different volumes. Target volumes selected by the user may not be honored.

If a new, empty volume is added to a storage group, data sets moved into that storage group are likely to be placed on that volume.

If a data set's storage class has the guaranteed-space attribute and the user specifies output volumes, the data set is placed on the SMS volumes specified in the volume list if:

- All SMS-managed volumes specified with the OUTDDNAME or OUTDYNAM keyword belong to the same storage group.
- The ACS storage group routine assigns the data set to the storage group that contains the specified SMS volumes.

**Note:** SMS-managed data sets must be cataloged in the standard order of search.

For a more complete discussion of SMS and how to use it, refer to *OS/390 DFSMSdss Storage Administration Reference*.

### Moving Data with Concurrent Copy

Concurrent copy and virtual concurrent copy can be used during copy as well as dump. See “Concurrent Copy” on page 4 for details about concurrent copy and virtual concurrent copy.

## Moving Data with FlashCopy

If conditions are right, DFSMSDss can use the FlashCopy feature of the Enterprise Storage Server (ESS) to quickly move the data from the source location to the target location. The source and target devices must be in the same logical subsystem of the ESS. FlashCopy is much faster than traditional methods of data movement, especially when you are moving large amounts of data.

## Moving Data with Snapshot

When the source and target devices are in the RAMAC Virtual Array (RVA) and the data does not need to be manipulated (such as, reblocked, track packed to unlike), DFSMSDss may be able to use snapshot to quickly move the data from the source location to the target location. This process is referred to as native snapshot support, and it is much faster than traditional methods, especially when large amounts of data are moved.

---

## Converting Data to and from SMS Management

DFSMSDss is the primary tool for converting data to and from SMS management. There are two ways of converting data:

- Conversion of data sets with data movement
- Conversion of volumes without data movement

The following sections briefly describe these two kinds of conversion.

### Conversion of Data Sets with Data Movement

To convert data sets by data movement, use the DFSMSDss COPY or DUMP/RESTORE command. When moving data sets from non-SMS-managed volumes to SMS-managed volumes, DFSMSDss invokes ACS, which may assign class names to the data sets. Alternatively, you can specify the BYPASSACS and STORCLAS keywords with the COPY or RESTORE command to force the data sets to be SMS-managed.

When moving data sets out of SMS management, specify the BYPASSACS and NULLSTORCLAS keywords with the COPY or RESTORE command. DFSMSDss then bypasses ACS and drops the data set's class names. ACS can also make data sets non-SMS-managed; see "Chapter 8. Converting Data to and from SMS Management" on page 95 for more information.

### Conversion of Volumes without Data Movement

To convert volumes to and from SMS management without data movement, you can use the DFSMSDss CONVERTV command. This command lets you:

- **Prepare a volume for conversion.** Using the PREPARE keyword, you can stop new allocations and data set extensions to another volume while still allowing access to the data on the volume.
- **Convert a volume to SMS management.** Using the SMS keyword, you can convert a volume and all its data sets to SMS management.
- **Convert a volume from SMS management.** Using the NONSMS keyword, you can remove a volume and its data sets from SMS management.
- **Simulate conversion.** Using the TEST keyword, you can verify that the volume and its data sets are eligible for conversion and see what class names ACS would assign to the data sets.

See “Chapter 8. Converting Data to and from SMS Management” on page 95 for more information.

---

## Managing Space with DFSMSdss

DFSMSdss has four functions to help you manage your DASD space:

**COMPRESS** Compresses your partitioned data sets by taking unused space and consolidating it at the end of the data set. To make the unused space available for other data sets, you must use the **RELEASE** command. This does not apply to PDSEs.

**RELEASE** Releases the unused space in sequential, partitioned, and extended-format VSAM data sets for use by other data sets.

**DEFRAG** Consolidates the free space on a volume to help prevent out-of-space abends on new allocations.

**DUMP/RESTORE** Deletes unwanted data sets and combines data set extents. (The **COPY** command can also be used to combine data set extents.)

---

## Chapter 2. Requirements for Running DFSMSdss

This chapter describes the requirements for running DFSMSdss.

---

### Operating Environment

DFSMSdss is exclusive to OS/390 and is available only as a component of OS/390.

You can use the Stand-Alone restore program of DFSMSdss outside a system environment. Alternatively, you can run the Stand-Alone restore program on an IBM System/370™ (S/370™) in MVS/ESA™ mode, MVS/XA™ mode, and S/370 mode. It can also be run on an IBM System/390® in S/390® mode or S/370 mode. The available modes are dependent on your CPU type and model. Additionally, the Stand-Alone restore program can be run in a virtual machine under VM in either 370 mode or XA mode.

---

### Storage Requirements

In most cases, you can let DFSMSdss determine the amount of storage it uses for an operation. Sometimes, however, you might want closer control of the amount of storage DFSMSdss uses. Use the storage estimates in this section as a starting point for determining minimum region sizes in which DFSMSdss can run. Table 1 and Table 2 on page 10 show the minimum storage requirements, in bytes, to run each DFSMSdss operation. Table 3 on page 11 shows the minimum storage requirements, in bytes, to restore partitioned and VSAM data sets to unlike devices. The values include the storage required to load the DFSMSdss program into the region.

Storage requirements depend on your operating system configuration and your device and data set characteristics. The storage requirement estimates shown for the COPY, DUMP, and RESTORE commands are only for the full-volume copy, dump, and restore operations; they might vary for a data set operation.

If DFSMSdss determines that the storage requirements are greater than the storage available during processing, DFSMSdss issues error message ADR376E to indicate this. The out-of-storage condition might cause abend 80A during DFSMSdss postprocessing.

If you use buffers above 16 megabytes virtual storage, the buffer size is allocated independently of the region size.

If you use the PARALLEL command to run two or more DFSMSdss tasks concurrently, the total storage required is the sum of the storage required for all functions to be run in parallel. However, because DFSMSdss is reentrant, the DFSMSdss code is not duplicated in storage. Therefore, do not include the DFSMSdss load module size more than once.

Table 1. Minimum Storage Requirements for DFSMSdss Operations with I/O Buffers below 16-Megabyte Virtual

DFSMSdss Command	Storage Requirements
COMPRESS	dsssize + (2 * trksize of largest device) + (number of additional volumes, up to five * copysize)
CONVERTV	dsssize + (buffersize * 5)
COPY (FULL)	dsssize + (trksize * 5)
COPYDUMP	dsssize + (buffersize * 5)
DEFRAG	dsssize + (trksize * 5) + (16KB if VSAM present) + (1KB * number of non-VSAM entries in VVDS)
DUMP (FULL) OPT(1)	dsssize + (buffersize * 7)
DUMP (FULL) OPT(2)	dsssize + (buffersize * 6)
DUMP (FULL) OPT(3)	dsssize + (buffersize * 15)
DUMP (FULL) OPT(4)	dsssize + (buffersize * (3 * trk/cyl))
PRINT	dsssize + (3 * trksize)
RELEASE	dsssize + trksize
RESTORE (FULL)	dsssize + copysize + (buffersize * 6)

Table 2. Minimum Storage Requirements for DFSMSdss Operations with I/O Buffers above 16-Megabyte Virtual

DFSMSdss Command	Storage Below 16MB Virtual	Storage Above 16MB Virtual
COMPRESS	dsssize + (number of additional volumes, up to five * copysize)	2 * trksize of largest device
CONVERTV	dsssize	buffersize * 5
COPY (FULL)	dsssize	trksize * 5
COPYDUMP	dsssize	buffersize * 5
DEFRAG	dsssize + (16KB if VSAM present) + (1KB * number of non-VSAM entries in VVDS)	(trksize * 5) + (152 * number of data set extents) + (144 * number of VSAM components) + (48 * number of non-VSAM data sets)
DUMP (FULL) OPT(1)	dsssize	buffersize * 7
DUMP (FULL) OPT(2)	dsssize	buffersize * 6
DUMP (FULL) OPT(3)	dsssize	buffersize * 15
DUMP (FULL) OPT(4)	dsssize	buffersize * (3 * trk/cyl)
PRINT	dsssize	3 * trksize
RELEASE	dsssize	trksize
RESTORE (FULL)	dsssize	buffersize * 6

Table 3. Minimum Storage Requirements for a Restore to an Unlike Device

Type of Data Set	Storage Requirements
Partitioned Data Sets	$dsssize + (((trksize + 64) * 5) + 8KB)$
VSAM Data Sets	$dsssize + (((trksize + 64) * 5) + (2 * \text{maximum record size}) + (3 * \text{buffspace}))$

**Legend:**

- KB** 1024 bytes.
- dsssize** DFSMSdss load module size, 1432KB.
- copysize** IEBCOPY load module size + IEBCOPY storage requirements below the 16MB line (that is, 1MB minimum, or 2MB if the data set being compressed has more than 1000 members).
- buffersize** 64KB for output to tape unless DFSMSdss is forced to a smaller blocksize. The maximum of the largest trksize used and a blocksize of 32KB.
- buffspace** Buffer space specified in the DEFINE command when the data set was allocated.
- trksize** The size, in bytes, of a track on your DASD volume.
- trks/cyl** The number of tracks per cylinder on your DASD volume.

## Hardware Requirements

You can use DFSMSdss with all IBM DASD magnetic tape devices, system consoles, printers, and card readers supported by Data Facility Product.

**Notes:**

1. VSAM extended addressability requires a cached storage subsystem that has concurrent copy-capable Licensed Internal Code.
2. DFSMSdss does not support virtual input/output (VIO) devices.

## Stand-Alone Restore Program for DFSMSdss

Refer to *OS/390 DFSMSdss Storage Administration Reference* for a list of devices you can use with the Stand-Alone DFSMSdss restore program.

## Volume Formats

You can use DFSMSdss with the following DASD volume formats:

- Volumes with indexed VTOCs
- Volumes with nonindexed VTOCs
- OS/VS minivolumes in a VM environment
- VM-formatted volumes (full or mini) with an OS-compatible VTOC beginning on track zero, record five.

All DASD volumes used by DFSMSdss must be initialized by Device Support Facilities (ICKDSF) and be mounted and online.

**Note:** You cannot use concurrent copy on minivolumes of any format unless they are on an RVA. However, you can use concurrent copy on full VM-format volumes that contain minivolumes to the extent that they are supported by IBM Extended Facilities Product (IXFP) device reporting.

---

## Data Set Organizations

DFSMSDss can copy, dump, and restore data sets of the following types:

- DATABASE 2™ (DB2®)
- Direct
- EXCP (execute channel program)
- Indexed sequential
- Partitioned, including:
  - PDS (partitioned data set)
  - PDSE (partitioned data set extended)
  - HFS (hierarchical file system) data set
- Sequential, including extended-sequential data sets
- VSAM data sets that are cataloged in an ICF catalog, including:
  - ESDS (entry-sequenced data set)
  - KSDS (key-sequenced data set)
  - KSDS with key ranges
  - LDS (linear data set)
  - RRDS (relative record data set)
  - VRRDS (variable relative record data set)
  - Extended-format ESDS, KSDS, LDS, RRDS, and VRRDS, including striped ESDS, KSDS, LDS, RRDS, and VRRDS
  - Extended-addressable VSAM ESDS, KSDS, LDS, RRDS, and VRRDS, including striped ESDS, KSDS, LDS, RRDS, and VRRDS
- Unmovable data set types (PSU, POU, DAU, ABSTR, ISU, and direct with OPTCD=A).

### Notes:

1. The logical data set copy function does not support hierarchical file system (HFS) data sets.
2. Physical data set operations do not support extended-format VSAM data sets. DFSMSDss does not provide conversion between nonextended-format VSAM and extended-format VSAM.
3. DFSMSDss cannot be used to process migrated data sets.

---

## Temporary Data Set Names

DFSMSDss must allocate the following temporary data sets to perform certain functions such as copy and restore. The high-level qualifiers of those data set names can be protected, and your installation must ensure that these temporary data sets can be allocated.

### Message data set—

allocated by DFSMSDss to store messages. This data set lets DFSMSDss print out messages by task rather than intermixing them. This data set is deleted when DFSMSDss completes the operation. System-generated temporary names are used.

### Special DEFrag data set—

allocated by DFSMSDss to contain information about the DASD extents that are being moved. The data set name is in the following format:

```
SYS1.DFDSS.DEFRAG.xxxxxxxx.volser.DUMMY
```

where xxxxxxxx represents 8 bytes of X'FF', and volser is the volume serial number of the volume being defragmented. The data set is deleted when the DEFrag operation completes successfully. If the DEFrag operation is

interrupted (for example, when DFSMSdss is canceled), this data set is left on the volume. You must perform the following operations:

- You must run a new DEFRAG operation.
- You may have to convert an index VTOC (IXFORMAT) volume to nonindexed (OSFORMAT) before rerunning the DEFRAG operation. Otherwise, the volume free-space values may be incorrect.
- The hexadecimal qualifier is used to prevent you from deleting this data.

**Temporary copied data sets—**

allocated by DFSMSdss when a copy is performed and deleted when the copy is completed.

The format of the temporary name depends on the number of qualifiers of the data set being copied:

Number of qualifiers ( <i>n</i> )	Temporary name
1	dsnhlq.chmmsstt
2–5	First <i>n</i> –1 qualifiers.chmmsstt
> 5	First 4 qualifiers.chmmsstt

where *c* is:

- T** Target cluster name
- D** Target data component name
- I** Target index component name
- U** Source cluster name
- E** Source data component name
- J** Source index component name

and *hmmsstt* is the time stamp information in low-order hours digits (*h*), minutes (*mm*), seconds (*ss*), and hundredths of a second (*tt*). In the course of copying data sets, DFSMSdss will rename the source data set using the above conventions. Whenever DFSMSdss renames a data set that is protected by RACF, to a temporary name, a RACF profile must exist for the temporary data set name.

**Temporary copied catalogs—**

allocated by DFSMSdss when copying a catalog. When DFSMSdss copies a catalog, two temporary data sets are used.

First, DFSMSdss allocates a temporary data set into which records are temporarily exported. The export data set name format is:

CATHLQ.EXPORT.Thmmsstt

where,

**CATHLQ**

The first three high-level qualifiers of the catalog being copied.

**hmmsstt**

The time-stamp information in low-order hours digits (*h*), minutes (*mm*), seconds (*ss*), and hundredths of a second (*tt*).

Second, DFSMSdss allocates a temporary catalog. The temporary catalog name format is:

CATHLQ.Thmmsstt

where,

**CATHLQ**

The first four high-level qualifiers of the catalog being copied.

**hmmsstt**

The time-stamp information in low-order hours digits(h), minutes (mm), seconds (ss), and hundredths of a second (tt).

**Dummy data set—**

allocated by DFSMSdss when copying or restoring volumes and an indexed VTOC needs to be rebuilt or the volume free-space values need to be recalculated: The data set name is in the following format:

```
SYS1.VTOCIX.DSS.TEMP.volser
```

where volser is the volume serial number of the volume being restored. Allocation of this data set is never successful because DFSMSdss uses dummy allocation values.

---

## Chapter 3. Logical and Physical Processing and Data Set Filtering

Before you begin using DFSMSdss, you should understand the difference between logical and physical processing and how to use data set filtering to select data sets for processing. The following sections describe these two aspects of DFSMSdss.

---

### Logical and Physical Processing

DFSMSdss can perform two kinds of processing when executing COPY, DUMP, and RESTORE commands:

- *Logical processing* operates against data sets independently of physical device format.
- *Physical processing* moves data at the track-image level and operates against volumes, tracks, and data sets.

Each type of processing offers different capabilities and advantages.

During a restore operation, the data is processed the same way it is dumped because physical and logical dump tapes have different formats. If a data set is dumped logically, it is restored logically; if it is dumped physically, it is restored physically. A data set restore operation from a full-volume dump is a physical data set restore operation.

### Logical Processing

A logical copy, dump, or restore operation treats each data set and its associated information as a logical entity, and processes an entire data set before beginning the next one.

Each data set is moved by tracks from the source device and is potentially written to the target device as a set of data records, allowing data movement between devices with different track and cylinder configurations. Checking of data record consistency is not performed during dump operations.

DFSMSdss performs logical processing if:

- You specify the DATASET keyword with the COPY command. A data set copy is always a logical operation, regardless of how or whether you specify input volumes.
- You specify the DATASET keyword with the DUMP command, and either no input volume is specified, or LOGINDDNAME, LOGINDYNAM, or STORGRP is used to specify input volumes.
- The RESTORE command is performed, and the input volume was created by a logical dump.

Catalogs and VTOCs are used to select data sets for logical processing. If you do not specify input volumes, the catalogs are used to select data sets for copy and dump operations. If you specify input volumes for the COPY or DUMP command with LOGINDDNAME, LOGINDYNAM, or STORGRP, the VTOCs are used *for starting the filtering process*.

**Note:** To copy or dump entire multivolume data sets, you need not specify all the volumes in the LOGINDDNAME or LOGINDYNAM volume list if you also specify the SELECTMULTI keyword with either the FIRST or ANY subkeywords.

## When to Use Logical Processing

Use logical processing for the following situations:

- Data is copied to an unlike device type.  
Logical processing is the only way to move data between unlike device types.
- Data that may need to be restored to an unlike device is dumped.  
Data must be restored the same way it is dumped. This is particularly important to bear in mind when making backups that you plan to retain for a long period of time (such as vital records backups). If a backup is retained for a long period of time, it is possible that the device type it originally resided on will no longer be in use at your site when you want to restore it. This means you will have to restore it to an unlike device, which can be done only if the backup has been made logically.
- Aliases of VSAM user catalogs are to be preserved during copy and restore functions.  
Aliases are not preserved for physical processing.
- Unmovable data sets or data sets with absolute track allocation are moved to different locations.
- Multivolume data sets are processed.
- VSAM and multivolume data sets are to be cataloged as part of DFSMSdss processing.
- Data sets are to be deleted from the source volume after a successful dump or copy operation.
- Both non-VSAM and VSAM data sets are to be renamed after a successful copy or restore operation.
- You want to control the percentage of space allocated on each of the output volumes for copy and restore operations.
- You want to copy and convert a PDS to a PDSE or vice versa.
- You want to copy or restore a data set with an undefined DSORG to an unlike device.
- You want to keep together all parts of a VSAM sphere.

## Physical Processing

Physical processing moves data based on physical track images. Because data movement is carried out at the track level, only target devices with track sizes equal to those of the source device are supported. Physical processing operates on volumes, ranges of tracks, or data sets. For data sets, it relies only on volume information (in the VTOC and VVDS) for data set selection, and processes only that part of a data set residing on the specified input volumes.

### Notes:

1. VSAM data sets are *not* cataloged during physical processing within SMS or non-SMS environments. The CATALOG keyword is ignored for VSAM data sets during physical restore. Use IDCAMS DEFINE RECATALOG to catalog the data sets after the physical restore.
2. The RENAME and RENAMEUNCONDITIONAL keywords are ignored for VSAM data sets during physical restore.

DFSMSDss performs physical processing if:

- You specify the FULL or TRACKS keyword with the COPY or DUMP command. This results in a physical volume or physical tracks operation.

**Attention:** Take care when invoking the TRACKS keyword with the COPY and RESTORE commands. The TRACKS keyword should be used only for a data recovery operation. For example, you can use it to “repair” a bad track in the VTOC or a data set, or to retrieve data from a damaged data set. You cannot use it in place of a full-volume or a logical data set operation. Doing so could destroy a volume or impair data integrity.

- You specify the DATASET keyword on the DUMP command and input volumes with the INDDNAME or INDYNAM parameter. This produces a physical data set dump.
- The RESTORE command is executed and the input volume is created by a physical dump operation.

### When to Use Physical Processing

Use physical processing when:

- Backing up system volumes that you might want to restore with a Stand-Alone DFSMSDss restore operation.

Stand-Alone DFSMSDss restore supports only physical dump tapes.

- Performance is an issue.

Generally, the fastest way—measured by elapsed time—to copy or to dump an entire volume is with a physical full-volume command. This is primarily because minimal catalog searching is necessary for physical processing.

- Substituting one physical volume for another or recovering an entire volume. With a COPY or RESTORE (full-volume or track) command, the volume serial number of the input DASD volume can be copied to the output DASD volume.
- Dealing with I/O errors. Physical processing provides the capability to copy, dump, and restore a specific track or range of tracks.
- Dumping or copying between volumes of the same device type but different capacity.

## Data Integrity Considerations

*Broken* data sets or volumes are ones that do not comply with the defined IBM standard. Such data sets and volumes may cause no problems during normal operations, and DFSMSDss does not usually detect or repair them. However, if DFSMSDss processing relies on the data set or volume structure being correct, the broken data may be unusable.

In a few circumstances DFSMSDss can detect and correct inconsistencies found in data that it is processing. For example, DFSMSDss does consistency checking to verify the reliability of a partitioned data set (PDS) directory before using it. You can also use the CHECKVTOC keyword (on the COPY and DUMP commands) to have DFSMSDss do extra consistency checks on the volume table of contents (VTOC) of a volume before processing data on the volume.

If you are making backups to be used for disaster recovery, you might want to take extra precautions to ensure that your data is valid. You can do this before invoking DFSMSDss or as part of the DFSMSDss invocation (for example, running access method services DIAGNOSE periodically reorganizing VSAM data sets and compressing PDSs, using the DFSMSDss CHECKVTOC keyword, and so forth).

You may choose logical or physical processing depending on what type of abnormal condition (if any) you expect. Neither processing mode provides a significantly higher level of data integrity. They are simply different views of the same data; one mode might detect a condition that the other could miss. For example, the more frequent PDS abnormal conditions do not cause problems during physical processing but could during logical processing. Usually, the DFSMSDss processing mode you choose should mirror as closely as possible the mode in which you typically access your data. For most situations, logical processing is the best choice.

---

## Choosing Data Sets for Processing—Filtering

You can select data sets for DFSMSDss processing by filtering on specified criteria. DFSMSDss can filter on fully qualified or partially qualified data set names (by using the INCLUDE or EXCLUDE keyword) and on various data set characteristics (by using the BY keyword).

You can filter data sets with any of the following commands:

- Logical dump
- Logical restore
- Physical data set dump
- Physical data set restore
- Data set copy
- COMPRESS
- RELEASE

At least one of the INCLUDE, EXCLUDE, or BY parameters must be specified with the above commands.

**Note:** DFSMSDss cannot serialize all of the data sets being considered during filter processing. It is possible that, between the time when DFSMSDss does the filtering and builds the list of data sets to process and the time when DFSMSDss actually processes the data sets, some or all of the data sets may be moved, deleted, or migrated. The status of the moved, deleted, or migrated data sets will therefore have changed by the time they are processed, which may in turn cause the DFSMSDss operation to fail.

The following sections briefly describe what you can filter on and how you might use the available criteria.

### Filtering by Data Set Names

Using the INCLUDE or EXCLUDE keyword, you can filter on fully qualified or partially qualified data set names. A fully qualified data set name is one in which all qualifiers are completely spelled out. For example:

```
(INCLUDE(SYS1.UTIL3.LOAD))
```

A partially qualified data set name is one in which the qualifiers are not completely spelled out. Using asterisks (\*) and percent signs (%), you can select data sets without specifying their fully qualified names.

The single asterisk (\*) is used in place of one qualifier. For example:

```
(INCLUDE(ABC.*.LOAD))
```

This partially qualified name matches ABC.DEF.LOAD and ABC.XYZ.LOAD. The single \* is also used to indicate that only part of a qualifier has been specified. For example, if you want to filter using only the first three characters of the first qualifier of a name, specify it as follows:

```
(INCLUDE(SYS*.**))
```

This partially qualified name matches data sets whose first qualifier was SYS1 and SYS1A. The other qualifiers in the data set name are ignored.

When used with other qualifiers, the double asterisk (\*\*) indicates that one or more leading, trailing, or middle qualifiers do not exist or they do not play a role in the selection process. For example:

```
(INCLUDE(**.LOAD))
```

This partially qualified name selects any data set with LOAD as its last qualifier (such as data sets named LOAD, ABC.LOAD, and ABC.DEF.LOAD).

The percent sign (%) is used as an ignore character. Each % sign represents one character in the name being filtered, and any character in that position is ignored. One or more % signs can be specified in any qualifier. For example:

```
(INCLUDE(SYS1.A%B))
```

This partially qualified name matches SYS1.AZZB and SYS1.AXYB, but not SYS1.AXXXB.

## Filtering by Data Set Characteristics

Using the BY parameter, you can filter on the following data set characteristics:

<b>Keyword</b>	<b>Criteria</b>
<b>ALLOC</b>	Allocation type (cylinder, track, block, absolute track, or movable)
<b>CATLG</b>	Whether a data set is cataloged or not (using the standard catalog search order)
<b>CREDIT</b>	Creation date (absolute or relative)
<b>DATACLAS</b>	Data class for SMS
<b>DSCHA</b>	Whether the data-set-changed flag is on or off
<b>DSORG</b>	Data set organization (SAM, PAM, PDS, PDSE, BDAM, EXCP, HFS, ISAM, or VSAM)
<b>EXPDT</b>	Expiration date (absolute or relative)
<b>EXTNT</b>	Number of extents
<b>FSIZE</b>	Data set size (number of allocated or used tracks)
<b>MGMTCLAS</b>	Management class for SMS
<b>MULTI</b>	Whether the VTOC shows that the data set is single-volume or multivolume <sup>3</sup>
<b>REFDT</b>	Last-referenced date (absolute or relative)

---

3. Single-volume data sets that have been allocated but have never been opened and are not cataloged may be selected as multivolume.

**STORCLAS** Storage class for SMS

You can use any of the following operators with the BY keyword:

<b>Operator</b>	<b>Meaning</b>
<b>EQ</b> or =	Equal to
<b>LT</b> or <	Less than
<b>LE</b> or <=	Less than or equal to
<b>GT</b> or >	Greater than
<b>GE</b> or >=	Greater than or equal to
<b>NE</b> or ≠	Not equal to

When you specify multiple arguments for an NE operation, DFSMSdss selects only those data sets not matching any of the arguments. When you specify multiple arguments for an EQ operation, DFSMSdss selects those data sets matching any of the arguments.

### **Some Examples of Filtering by Data Set Characteristics**

If you use the following specification of the BY keyword, DFSMSdss selects all data sets allocated in cylinders:

```
BY( ALLOC,EQ,CYL )
```

You can specify more than one criterion with the BY keyword. The following example selects all data sets allocated in cylinders and whose management class is MCNAME1:

```
BY(( ALLOC,EQ,CYL ) ( MGMTCLAS,EQ,MCNAME1 ))
```

You can specify multiple arguments for any of the filtering criteria. The following example selects all data sets that have a data class of DCNAME1 or DCNAME2:

```
BY( DATACLAS,EQ,(DCNAME1,DCNAME2) )
```

## **The FILTERDD Keyword**

The FILTERDD keyword must be used if you have more than 255 entries in the INCLUDE, EXCLUDE, or BY filtering lists. The FILTERDD keyword specifies the name of the DD statement that identifies the sequential data set or member of a partitioned data set that contains the filtering criteria to be used. This is in the form of card-image records, in DFSMSdss command syntax, that contain the INCLUDE, EXCLUDE, and BY keywords.

## **Some Uses of Filtering**

You will make the best use of filtering by data set names if you use meaningful naming conventions. Your naming conventions should allow you to identify large groups of data sets that can be treated similarly. With such conventions, you can use data set name filtering to select large groups of data sets against which you can run DFSMSdss functions.

For more information on setting up data set naming conventions, refer to *MVS/ESA SML: Managing Data*.

Suppose you are a storage administrator and you want to do a daily backup of all payroll data sets that have changed since they were last backed up. If the data sets you want to back up have some identifying qualifiers (for example, PAYROLL.FEDTAX), you can select them by coding:

```
//VRPAY    JOB  Accounting Information,MORGAN
//STEP1    EXEC PGM=ADRDSSU,REGION=4000K
//SYSPRINT DD  SYSOUT=*
//DROUT    DD  DSN=PAYROLL.DAY1,DISP=(NEW,CATLG),UNIT=3480,LABEL=(1,SL)
//SYSIN    DD  *
            DUMP DATASET(INCLUDE(PAYROLL.FEDTAX.**)) -
                BY((DSCHA,EQ,YES) (MGMTCLAS,EQ,DAILY))) -
            OUTDD(DROUT)
/*
```

Filtering by data set characteristics also lets you process large groups of data sets. You can use BY criteria to:

- Filter on the data-set-changed-flag to back up only those data sets that have not been backed up since they were last updated.
- Filter to select uncataloged data sets for deletion as a means of enforcing cataloging.
- Filter to select data sets whose expiration date passed for deletion.
- Filter on the last referenced date to archive or delete data sets that have not been referenced for a long period of time (for example, 18 months).
- Filter on data set size to ensure that when you use the COMPRESS and RELEASE commands, you compress and release space only in data sets where the savings may be significant.
- Filter on management class to perform space management (if in an SMS-managed environment).

It is possible to pass DFSMSdss filtering criteria in a data set by using the FILTERDD keyword. If you do this, the data set should have the following characteristics:

- RECFM=F or FB
- LRECL=80
- BLKSIZE=80 for F (or a multiple of 80 for FB).



---

## Chapter 4. Invoking DFSMSDss

You can invoke DFSMSDss with:

- Interactive Storage Management Facility (ISMF)
- Job control language (JCL)
- The application interface

---

### Invoking DFSMSDss with ISMF

You can use the menu-driven panels of ISMF to build job streams for many DFSMSDss space management and backup functions. ISMF supports the DFSMSDss commands COMPRESS, CONVERTV, COPY, DEFRAG, DUMP, RELEASE, and RESTORE.

The information you supply on ISMF panels is used to build and submit job streams like those you generate using JCL and DFSMSDss commands. Using ISMF panels, you do not have to remember DFSMSDss keywords and syntax. Simply fill in the values you want on the panels, and ISMF generates the job stream. You can then either submit the job or save the job stream for later use.

Using ISMF panels, you can build a list of data sets or volumes according to criteria that you provide. The list provides information about each volume or data set (for example, allocated space and percent of unused space). You can use the list to analyze and manage your data and storage more efficiently.

### How to Invoke ISMF

You invoke ISMF by logging on to TSO. If ISMF is installed as an option on the ISPF Master Application Menu or as an option on the ISPF/PDF Primary Option Menu, specify the selection option that corresponds to ISMF. You can use ISMF to perform DFSMSDss functions against one or more data sets or volumes on a list you create. Extensive help screens are available for all the DFSMSDss functions supported by ISMF. Refer to *OS/390 DFSMS: Using the Interactive Storage Management Facility* for more details.

---

### Invoking DFSMSDss with JCL

DFSMSDss is controlled by JCL statements and DFSMSDss commands. You can use the JCL statements to invoke DFSMSDss and to define the data sets used and created by it. The JCL defines the DFSMSDss commands that specify and control tasks. Refer to the chapter about specifying DFSMSDss commands in *OS/390 DFSMSDss Storage Administration Reference* for JCL information and examples.

---

### Invoking DFSMSDss with the Application Interface

This section documents General-Use Programming Interface and Associated Guidance Information provided by DFSMSDss.

You can invoke DFSMSDss from an application program by using the application interface. This allows you, for example, to gather statistical or auditing information and to specify control variables.

The application interface allows you to:

- Fully utilize the invocation capabilities of DFSMSDss when the ATTACH, LINK, or CALL system macro is specified in your application program.
- Optionally, specify a list of parameters to be used by DFSMSDss during the processing caused by that invocation.
- Optionally, interact with DFSMSDss during processing of user installation options after the installation options exit has been called.
- Optionally, interact with DFSMSDss during the processing at convenient points where input/output (I/O) operations are being performed.

**Note:** DFSMSDss runs as an authorized problem program (nonsupervisor state); any program invoking DFSMSDss must also be authorized and in nonsupervisor state.

For more information on the application interface, refer to *OS/390 DFSMSDss Storage Administration Reference*.

## User Interaction Module Exit Functions

When DFSMSDss is invoked from an application program, you can use the user interaction module (UIM) to interact with DFSMSDss at points where I/O operations are being performed. UIM exit functions can be used to:

- Replace, insert, delete, or modify a SYSIN record after DFSMSDss has read it or a SYSPRINT record when DFSMSDss is ready to print it.
- Replace, insert, delete, or modify a write-to-operator message before DFSMSDss writes it.
- Insert a statistics record during a logical dump operation.
- Modify the installation options specified in the ADRUFO control block to override the specified options.
- Bypass password and expiration-date checking, or reject the tape volume and request a scratch tape, when DFSMSDss is ready to open a tape.
- Request a specific volume serial when a nonspecific tape is passed to DFSMSDss.
- Get information about the data set being allocated.
- End a task or processing of individual data sets.
- Bypass authority checking for individual data sets. This includes both RACF and password authorization.
- Bypass serialization checking of individual data sets.
- Show the status of the concurrent copy initialization.

For more information and examples of UIM exit functions, refer to *OS/390 DFSMSDss Storage Administration Reference*.

---

## Chapter 5. Protecting DFSMSdss Functions

This chapter discusses the functions of DFSMSdss to which you can control access by using RACF, an element of the OS/390 Security Server. You can protect DFSMSdss/ISMF functions and some DFSMSdss keywords.

---

### Protecting DFSMSdss/ISMF Functions with the OS/390 Security Server RACF Element

Using the program control feature of the OS/390 Security Server RACF elements Version 1 Release 7 or later, you can set authorization levels for ISMF functions: ISMF itself, each of the ISMF applications, and the individual line operators and commands. In addition, each ISMF function you identify the RACF element for authorization checking is included in the RACF report and logging process. You can also use standard RACF authorization checking to limit access to individual data sets, volumes, or catalogs. Used in conjunction with program control, authorization checking ensures that the appropriate ISMF data and functions are available to users when they need them.

#### ISMF Functions You Might Want to Protect

Program control allows you to determine the ISMF functions to which users have access. The authorization scheme you set up can apply to both individual users and user groups. The ISMF functions you can protect fall into two general categories: line operators and commands.

With program control, you can set up authorization levels for each category. You can also vary the level within a category to suit the needs of your site. Before you set up an authorization structure, consider the following:

- Do you want all users at your site to have access to ISMF?
- Do you want all users to have access to the data set, volume, or profile applications?
- Are there line operators or commands to which you want to limit access?

#### Setting Up the Authorization Structure

RACF program control checks authorization before allowing access to an ISMF function. Protection for each function is based on the authorization level of the load module that contains the function. A user is allowed to execute an ISMF function (for example, the RESTORE list command) when one of the following is true:

- The user is authorized to execute the load module corresponding to the function requested. Authorization is defined as READ level access or greater.
- The user's RACF profile has the OPERATIONS attribute.
- The user's group is authorized to execute the load module.
- The universal access authority (UACC) for the load module is READ or greater. This makes the load module available to anyone who can access ISMF.

## Finding the DFSMSdss/ISMF Module Names

### Programming Interface information

The names of the load modules for DFSMSdss/ISMF are stored in command tables in both the panel library, DGTPLIB, and the load library, DGTLLIB. The load module names are listed in Table 4 and Table 5. The module names are found in the DGTSMMD1 member of the panel library.

Table 4 lists the names for the corresponding line operators. The module names for line operators are found in the DGTTLPD3 member of the load library. Table 5 lists the names for commands. These names are in the DGTCTD2 member of the load library.

Table 4. Module Names for DFSMSdss/ISMF Line Operators

Line Operator	Data Set Application Module Name	Volume Application Module Name
COMPRESS	DGTFCM01	DGTFC01
CONVERTV	—	DGTFCN01
COPY	DGTFCY01	DGTFCV01
DEFRAG	—	DGTDF01
DUMP	DGTDFP01	DGTDFM01
RELEASE	DGTFR01	DGTFRV01
RESTORE	DGTFR01	DGTFR01

Table 5. Module Names for DFSMSdss/ISMF Data Set Application Commands

Command	Module Name
COMPRESS	DGTFCP01
COPY	DGTFCO01
DUMP	DGTFDU01
RELEASE	DGTFRE01
RESTORE	DGTFR00

To view the command table, you need to know the data set names that your site uses for the panel library and the load library. The installation of DFSMSdss/ISMF puts the panel library in SYS1.DGTPLIB and the load library in SYS1.DGTLLIB. However, your site's postinstallation procedures might involve moving the DFSMSdss/ISMF libraries. If they were moved, you can determine the data set name by issuing the TSO LISTALC command and scanning the low-level qualifiers for DGTPLIB and DGTLLIB.

### End of Programming Interface information

## Protecting DFSMSdss/ISMF Modules

The two steps to protect DFSMSdss/ISMF modules are:

1. To define the modules you want to protect, use the RDEFINE command or the ISPF RACF entry panels. When you define the modules to RACF, supply the name of the load module you want to protect, the name of the data set that contains the module, and the volume serial number of the volume that contains

the data set. Each module you identify is added to the profile for the PROGRAM general resource class. You have several options when you define modules:

- If you want to define several modules at the same time, you can use asterisk notation. For example, DGT\* means all the modules beginning with the letters DGT.
  - You can add an access list with user IDs, group names with their associated access authority to the profile, or both.
  - You can define the UACC to give default access to all users or to none.
  - You can use the AUDIT parameter to set up RACF logging or to bypass it.
2. To allow users to execute an application, line operator, or command, use the PERMIT command.

For more information on how to perform these steps and the options you have using program control, refer to *OS/390 SecureWay Security Server RACF Security Administrator's Guide*.

---

## Protecting DFSMSdss Keywords with RACF

In addition to protecting DFSMSdss/ISMF functions, you can also protect certain DFSMSdss keywords. This is done by defining FACILITY class resource profiles and restricting access to those profiles. Table 6 lists these keywords and their associated RACF class profiles. For a given command or parameter, protection occurs when both of the following conditions are met:

- RACF FACILITY class is active
- The indicated FACILITY class profile has been defined

When FACILITY class is active and one of the below profiles is defined, you need READ access authority to it in order to use the indicated command or parameter. Otherwise, anybody can use the indicated command or parameter. If OS/390 Security Server, RACF element, Version 1 Release 7 or later is not installed, or if Facility Class checking is not set up for these keywords, any DFSMSdss user can use them.

*Table 6. RACF Facility Class Profile Names for DFSMSdss Keywords*

<b>Keyword</b>	<b>Profile Name</b>
BYPASSACS with COPY	STGADMIN.ADR.COPY.BYPASSACS
BYPASSACS with RESTORE	STGADMIN.ADR.RESTORE.BYPASSACS
CONCURRENT with COPY	STGADMIN.ADR.COPY.CNCURRNT
CONCURRENT with DUMP	STGADMIN.ADR.DUMP.CNCURRNT
CONVERTV	STGADMIN.ADR.CONVERTV
DEFRAG	STGADMIN.ADR.DEFRAG
DELETECATALOGENTRY with RESTORE	STGADMIN.ADR.RESTORE.DELCATE
IMPORT with RESTORE	STGADMIN.ADR.RESTORE.IMPORT
INCAT(catname) with COPY	STGADMIN.ADR.COPY.INCAT
INCAT(catname) with DUMP	STGADMIN.ADR.DUMP.INCAT
INCAT(catname) with RELEASE	STGADMIN.ADR.RELEASE.INCAT
PROCESS(SYS1) with COPY	STGADMIN.ADR.COPY.PROCESS.SYS
PROCESS(SYS1) with DUMP	STGADMIN.ADR.DUMP.PROCESS.SYS
PROCESS(SYS1) with RELEASE	STGADMIN.ADR.RELEASE.PROCESS.SYS

Table 6. RACF Facility Class Profile Names for DFSMSdss Keywords (continued)

<b>Keyword</b>	<b>Profile Name</b>
TOLERATE(ENQF) with COPY	STGADMIN.ADR.COPY.TOLERATE.ENQF
TOLERATE(ENQF) with DUMP	STGADMIN.ADR.DUMP.TOLERATE.ENQF
TOLERATE(ENQF) with RESTORE	STGADMIN.ADR.RESTORE.TOLERATE.ENQF

You can bypass this type of FACILITY class checking with the DFSMSdss installation options exit routine that your installation may be using. Refer to *OS/390 DFSMS Installation Exits* for details about this routine. Refer to *OS/390 SecureWay Security Server RACF Security Administrator's Guide* for additional information on RACF class profiles.

---

## Chapter 6. Managing Availability with DFSMSdss

One of the major functions of DFSMSdss is the backup and recovery of data. Using the DUMP and RESTORE commands, you can backup and recover data sets and volumes. You can also use the DUMP and RESTORE commands on ranges of tracks. However, this is usually done as a means of diagnosing I/O errors rather than as a means of backing up and recovering data.

---

### Planning an Availability Strategy

In planning your overall availability strategy, you should consider the following types of backup:

**Backup of volumes and data sets**—The general type of backup to guard against users accidentally losing or incorrectly changing their data sets and against losing volumes because of hardware failures.

**Disaster recovery backup**—Backup to protect against the loss of all your data in a major disaster at your site. These backups are stored off site and, in the event of a major disaster, are recovered at another site.

**Vital records backup**—Backup copies of data sets kept to meet externally imposed retention requirements, such as tax records.

**Archival**—Backup of data that is unused for a long period of time. You remove the data from DASD and retain it on tape in case it is needed again.

DFSMSdss is a flexible backup and recovery tool. You can use DFSMSdss by itself to perform all backups listed above or to complement other backup and recovery tools.

**Note:** Physical data set DUMP and RESTORE do not support extended-format VSAM data sets.

### Backup and Recovery

General backup should be done at both the data set and the volume level. To protect against users accidentally deleting or changing their data sets, it is usually more efficient to do incremental backup (logical backup of those data sets that changed since they were last backed up). Incremental backups minimize processing time because you are not backing up every data set. Logical backup lets you restore data sets to unlike devices.

#### Data Set Backup

For data set backup, you need to consider the frequency of backup and the number of versions you want to keep. A number of factors can influence this decision, such as:

- The rate at which the data changes.
- The ease or difficulty of rebuilding the data (for example, it is easier to rebuild an object library than a source library).
- The importance of the data. For data that is extremely important to your business, you might want to keep extra backup versions.

A more complete discussion of the things you must consider when determining frequency of backup and number of versions can be found in *MVS/ESA SML: Managing Data*.

## Volume Backup

Volume backup is necessary to guard against losing a volume, but it need not be done often if you are doing incremental backup on a regular basis. If you lose a volume, you can recover from the latest volume backup, and then recover data sets from incremental backups to return the volume to its status before the failure. This form of recovery is sometimes referred to as forward recovery. In order to perform it, though, you must have a record of all your backups. The DFSMShsm component keeps its own inventory of the data sets it backs up and can perform forward recovery using that inventory. DFSMSdss prints the names of the data sets it dumps and the serial number and data set sequence number of the tape volumes on which the dump begins and ends. You must use this printed record to perform forward recovery with DFSMSdss.

## Backup and Recovery in an SMS-Managed Environment

Two kinds of data exist in an SMS-managed environment: SMS-managed and non-SMS-managed data. DFSMSdss can help you fulfill your availability requirements for both kinds of data.

**SMS-Managed Data:** The DFSMShsm component can perform automatic volume backup (by invoking DFSMSdss) and incremental backup on SMS-managed data. Each data set is assigned a management class that indicates how often DFSMShsm should back it up and how many versions of the backup to keep. Using DFSMShsm this way lets you manage availability at the data set level.

If you do not have the DFSMShsm component installed, you can use DFSMSdss to back up and recover data sets and volumes. By filtering on management class name and the data-set-changed flag, you can perform incremental backup on all the data sets belonging to a particular management class. To facilitate this backup procedure, you can set up a DFSMSdss job to run periodically.

For more details on planning for backup of SMS-managed data, refer to *MVS/ESA SML: Managing Data*.

**Non-SMS-Managed Data:** Typically, non-SMS-managed data is data that SMS does not support or data that is in transition from non-SMS to SMS management. If it is data that SMS does not support, you can probably still use DFSMSdss to back it up and recover it, because DFSMSdss supports many kinds of data that SMS does not. If it is data in transition to SMS management, you can use DFSMShsm or DFSMSdss to back up and recover it until it is placed under SMS management.

## Backup and Recovery in a Non-SMS-Managed Environment

If SMS is not active, you are in a non-SMS-managed environment. For availability management, the data in this environment can be treated much the same as the non-SMS-managed data in an SMS-managed environment. DFSMSdss can be used to back up and recover it at the data set and volume level.

## Disaster Recovery

Disaster recovery backups are made specifically for recovering data and applications following a disaster. Never rely on your regular backup data sets (for instance, DFSMShsm or DFSMSdss incremental backups) for disaster recovery. Disaster recovery backups require some very special considerations that normally do not apply to other types of backups.

### Store at a Remote Site

A basic difference between regular backups and disaster recovery backups is that disaster recovery backups must be transported to a different site. How remote the

recovery site is depends upon the type of disaster you are preparing for (in the case of a fire, the recovery site can be around the corner; in the case of an earthquake or flood, it should be many miles away). The fact that the backups must be taken to another site means that they must be on a portable media: tape.

**Note:** You can also automatically transmit backups to another site.

### Use Logical Data Set Dump

Because the environment at the remote site might differ from your environment, you should ensure that your disaster recovery backups can be restored in a different environment. In general, it is recommended that you use the logical data set DUMP command and filter on the data set name to make disaster recovery backups. Logical data set dump processing allows you to back up only your critical data sets and to restore to unlike devices.

Making logical data set dumps for disaster recovery backup requires a naming convention or some other method to identify your critical data sets. If, for example, you establish the convention of having the letters CRIT as the first four characters in the first qualifier of critical data sets, you can back them up for disaster recovery as follows:

```
DUMP -  
  DATASET(INCLUDE(CRIT*,**)) -  
    BY(MGMTCLAS,EQ,MCNAME)) -  
  OUTDDNAME(TAPE) -  
  COMPRESS
```

If for some reason you must do volume dumps for disaster recovery, you should do logical volume dumps instead of physical volume dumps. That way, you can restore the backups to unlike devices. You can perform logical volume dumps by using DATASET (INCLUDE(\*\*)) and either the LOGINDDNAME or LOGINDYNAM keyword with the DUMP command.

### Back Up Only Critical Data Sets

You should back up only data sets that are critical to your operation. For example:

- Critical application data sets
- RACF inventory data sets
- System data sets
- Catalogs

Because you normally back up only critical data sets for disaster recovery, the amount of data you have to back up is only a small percentage of all your data. To identify those data sets you want backed up for disaster recovery, you should create a unique naming convention.

If you have DFSMSHsm installed on your system, the recommended method of disaster backup is to use aggregate backup and recovery support (ABARS). Refer to *OS/390 DFSMSHsm Storage Administration Guide* for more information on ABARS.

To maintain versions of your disaster recovery backups, you can use generation data group (GDG) dump data sets.

When recovering after a disaster, you may need to use the DELETEDCATALOGENTRY or IMPORT keywords or both. See “Logical Restore of Data Sets with Phantom Catalog Entries” on page 64.

For more information on disaster recovery, refer to *MVS/ESA SML: Managing Data*.

## Vital Records

Vital records are maintained to meet external retention requirements (such as legal requirements).

Like disaster recovery backups, vital records are kept at a remote site and therefore should reside on tape. Vital records are usually an even smaller percentage of all data than disaster recovery backups. Unlike disaster recovery backups, vital records are rarely necessary for normal processing.

Vital records are usually kept for long periods of time. The device they originally resided on may no longer be in use at the time of recovery, and you may need to restore them to unlike devices. Therefore, vital records should be dumped logically so they can be restored to unlike devices. As with disaster recovery, using logical data set DUMP processing requires a naming convention or some other method to identify data sets for dumping.

If, for example, you establish the convention of having the letters VR as the first two characters in the first qualifier of data sets to be backed up for vital records purposes, you can dump them as follows:

```
DUMP -  
  DATASET(INCLUDE(VR*.**) -  
          BY(MGMTCLAS,EQ,MCNAME)) -  
  OUTDDNAME(TAPE) -  
  COMPRESS
```

For more details on vital records, refer to *MVS/ESA SML: Managing Data*.

## Archiving

Archived data sets are data sets created to remove data from active status. This data is placed on alternate storage media because it is not currently being used but may be used in the future. Archived data sets are usually used for long-term retention.

You can use DFSMSdss to archive data sets by periodically filtering on last-referenced date and then dumping and deleting data sets that have not been referenced for long periods of time. This frees space for data that is being accessed more frequently and requires the faster access time of DASD. Because archived data sets might not be recovered for a long time, they should be dumped logically so they can be restored to unlike devices.

For example, the following logical DUMP command results in the archiving of all data sets in management class MCNAME1 that have not been referred to since April 10, 1986:

```
DUMP -  
  DATASET(BY((REFDT LT 86100)(MGMTCLAS EQ MCNAME1))) -  
  OUTDDNAME(TAPE1) -  
  DELETE -  
  COMPRESS -  
  PURGE
```

For more details on archiving, refer to *MVS/ESA SML: Managing Data*.

---

## Backing Up Data Sets

With the DUMP command, you can dump DASD data to a sequential data set, which can be a generation in a generation data group (GDG). The storage medium for the sequential data set can be tape or DASD. The output data set must be a standard format sequential data set and cannot use any extended-format features, such as compression.

DFSMSDss can dump data sets both logically and physically. Data sets are located by searching either the catalog or the VTOC.

You can select data sets for dump processing based on data set names and numerous data attributes, as discussed in “Choosing Data Sets for Processing—Filtering” on page 18. If you want to perform incremental backups with DFSMSDss, you can filter with BY(DSCHA,EQ,1) to dump only data sets that have changed since the last dump was taken. If you also code the RESET keyword, DFSMSDss changes the data-set-changed (DSCHA) indicator after successfully dumping the data set.

### Notes:

1. If you are using DFSMSDss on data sets that DFSMShsm is also backing up, you should not use the RESET keyword because it might cause confusion as to which backup is the most current.
2. DFSMSDss does not permanently record the names of candidate volumes during dump processing.
3. See “Backup with Concurrent Copy” on page 35 for more information about using the RESET keyword.

The data-set-changed indicator and the last-referenced date (REFDT) are supported for:

- Non-VSAM data sets
- VSAM data sets

For more information, refer to *OS/390 DFSMS Installation Exits*.

Temporary data sets that are created and deleted by other jobs running while DFSMSDss is running might be included in the data set list at the beginning of a DFSMSDss job, and might disappear at a later time before DFSMSDss finishes. DFSMSDss can issue a message informing the user what happened only at the time DFSMSDss tries to access the data set. If the user wants to hold all the data sets in a volume for the entire DFSMSDss execution, the user must write an enqueue installation exit to enqueue the volume for the entire job.

When you create backups of data sets with the DUMP command, you can make multiple (up to 255) dump copies with a single DUMP command. This is done by specifying multiple ddnames on the OUTDDNAME parameter. To specify multiple ddnames on the OUTDDNAME parameter, you could code:

```
DUMP -  
  DATASET(INCLUDE(**) -  
          BY(MGMTCLAS,EQ,MCNAME1)) -  
  OUTDDNAME(TAPE1,TAPE2,TAPE3) -  
  COMPRESS
```

This technique can be helpful if you want to create several backup copies to be used for different purposes.

Unless overridden by the installation options exit routine, DFSMSdss continues dumping while at least one output copy does not have an output error. In the event of an abend, however, DFSMSdss ends without completing any backups.

## Logical Data Set Dump

If you specify the DATASET keyword with the DUMP command and do not specify input volumes, DFSMSdss performs a logical data set dump using information in the catalogs to select data sets. For example, the following DUMP command results in a logical data set dump:

```
DUMP -  
  DATASET(INCLUDE(**) -  
          BY(DSCHA,EQ,YES)) -  
  OUTDDNAME(TAPE1) -  
  COMPRESS
```

If you specify the DATASET keyword with either LOGINDDNAME or LOGINDYNAM, DFSMSdss performs a logical data set dump using information in the VTOCs to select data sets. For example, the following DUMP command results in a logical data set dump of all the single-volume data sets on volume 338001:

```
DUMP -  
  DATASET(INCLUDE(**)) -  
  LOGINDYNAM(338001) -  
  OUTDDNAME(TAPE) -  
  COMPRESS
```

The following data sets cannot be processed by logical data set dump or restore operations:

- VSAM data sets not cataloged in an integrated catalog facility catalog
- Page, swap, and SYS1.STGINDEX data sets
- VSAM Volume Data Sets (VVDS)
- Partitioned data sets containing location-dependent information that does not reside in note lists or in the directory

**Note:** DFSMSdss cannot be used to dump data sets with a volume serial of MIGRAT. The recommended method of dumping migrated data sets is to use ABARS.

## Physical Data Set Dump

If you specify DATASET and INDDNAME or INDYNAM, DFSMSdss performs a physical data set dump. For instance, the following DUMP command results in a physical data set dump:

```
DUMP -  
  INDDNAME(DASD1) OUTDDNAME(TAPE) -  
  DATASET(INCLUDE(**)) -  
  COMPRESS -  
  OPTIMIZE(4)
```

When multiple input volumes are specified for a physical data set dump operation, multiple logical files (logical volumes) are created for each physical DASD source volume.

DFSMSdss facilitates backup and recovery procedures for physical data set dumps by printing the names of data sets dumped, and the serial and data set sequence numbers of the backup tape volumes on which the dump of a DASD volume begins and ends.

The following data sets cannot be processed by physical data set dump or restore operations:

- KSDSs with key ranges. Logical processing should be used for this type of data set.
- Extended-format VSAM data sets, including extended-addressable VSAM data sets. Use logical processing for these types of data sets.
- VSAM data sets not cataloged in an integrated catalog facility catalog.
- Page, swap, and SYS1.STGINDEX data sets.

**Note:** When dumping multivolume data sets, take care to ensure that all volumes where the data set resides are dumped at the same time and restored at the same time. Dumping parts of a multivolume data set and then restoring them may leave the entire data set or those parts unusable. In particular, keyed VSAM data sets are easily damaged by such an operation.

## Backup with Concurrent Copy

---

### Programming Interface information

DFSMSdss provides the concurrent copy function to let you backup data but minimize the time that the data is unavailable. The data base or application determines an appropriate time to start a backup (for example, when the data is in a known state and update activity is stopped). DFSMSdss is invoked directly or via the DFSMSdss application program interface (API) to do a concurrent copy of the entire data base. After initialization is complete, DFSMSdss releases any serialization it held on the data sets and prints a message to SYSPRINT and the console that the concurrent copy operation is logically complete. If DFSMSdss was invoked via the API, DFSMSdss informs the caller through a new UIM exit option, Eioption 24. Refer to *OS/390 DFSMSdss Storage Administration Reference* for details. The application can resume normal operation at this time.

---

### End of Programming Interface information

If for any reason data cannot be processed with concurrent copy, (for example, the hardware being used does not support concurrent copy) DFSMSdss uses normal backup methods and does not release the serialization until the backup is completed.

**Notes:**

1. The CONCURRENT keyword applies to all the data being dumped or copied by the function under which it is specified. It cannot be applied to a subset of the data being processed.
2. To improve data integrity, do not update the data during a concurrent copy initialization.
3. If a concurrent copy operation fails after signaling that the concurrent copy initialization was complete (and update activity on the data has resumed), it is not possible to recover the data at the point-in-time at which the concurrent copy operation was started. This is because the data may have been updated while the copy operation was progressing.

4. The CONCURRENT keyword cannot be used with the DELETE or UNCATALOG keywords.
5. The RESET keyword is ignored when CONCURRENT is also specified unless you use a patch to allow it. Refer to the appendix in *OS/390 DFSMSdss Diagnosis Guide* for more information on DFSMSdss patches.
6. VM mini-volumes are supported if you are using RVA devices to the extent that they are supported by IBM Extended Facilities Product (IXFP) device reporting.

If the source device supports snapshot, but does not support concurrent copy, DFSMSdss will use the snapshot function to provide a concurrent copy-like function known as virtual concurrent copy. Refer to “Performance Considerations” on page 44 for more information on concurrent copy and virtual concurrent copy.

### Invocation from an Application Program

Usage of the concurrent copy function can also be controlled through the installation options exit, a product-sensitive programming interface intended for customer use. Refer to *OS/390 DFSMS Installation Exits* for more information.

## A Backup Scenario

As discussed under “Backup and Recovery” on page 29, you should consider using a combination of incremental and volume backup to fulfill your general availability requirements. Some ways to implement this strategy are:

- Dump a full volume at a given interval—perhaps once a week. Use the RESET keyword to reset the data-set-changed flag. To do full-volume dumps of two volumes at once (in parallel, which is most effective if tapes are on separate channels), code the following:

```
PARALLEL
DUMP INDYNAM(111111) OUTDD(TAPE1) RESET OPTIMIZE(1)
DUMP INDYNAM(222222) OUTDD(TAPE2) RESET OPTIMIZE(2)
```

- Dump only changed data sets at a shorter interval—perhaps daily.

```
DUMP LOGINDY((111111),(222222)) OUTDD(TAPE3) RESET -
  OPTIMIZE(3) DATASET(INCLUDE(**) -
    BY(DSCHA,EQ,YES))
```

- Use data set naming conventions to set up a dumping scheme that takes account of the relative importance of the data. For example, include CRIT in the first-level qualifier of all your critical data sets. With this convention in place, you can back up your critical data sets as follows:

```
DUMP LOGINDY((111111),(222222)) OUTDD(TAPE4) RESET -
  OPTIMIZE(4) DATASET(INCLUDE(CRIT*,**) -
    BY(DSCHA,EQ,YES))
```

Other naming conventions can also be used to identify groups of data sets. For instance, you can use department numbers, charge numbers, user initials, or project codes to identify data sets you want to dump together. For more information on naming conventions, refer to *MVS/ESA SML: Managing Data*.

For data set operations, SYSPRINT contains the names of all the data sets that were dumped for each run. You should keep them for reference if you have to restore a data set and you want it to be at the latest level. This prints a listing of all the data sets that might be on the restore tape, and you can now find the latest dumped version of a particular data set.

---

## Backing Up Data Sets with Special Requirements

Some data sets require special processing when they are backed up. The sections below describe how to back up data sets that have special requirements.

### Dumping HFS Data Sets

The following topics present guidelines for backing up an HFS data set with either logical data set dump or physical data set dump.

#### Logical Dump

Back up mounted HFS data sets with logical data set dump. Logical data set dump provides the quiesce serialization mechanism (BPX1QSE) to ensure data integrity. The quiesce ability allows you to dump an HFS data set while it is in use, as long as you run the dump job on the same system that the HFS data set is currently mounted on.

With DFSMSdss prior to Version 1 Release 5, you had to specify the SHARE keyword when you wanted to dump a mounted HFS data set. When an HFS data set was mounted, OpenEdition MVS had a shared SYSDSN ENQ. Before DFSMSdss Version 1 Release 5, DFSMSdss obtained an exclusive SYSDSN ENQ if the SHARE keyword was not specified.

In DFSMSdss Version 1 Release 5, DFSMSdss no longer obtains a SYSDSN ENQ, so the SHARE keyword is not required during logical dump.

For further information on serialization of HFS data sets, refer to the serialization appendix in *OS/390 DFSMSdss Storage Administration Reference*.

#### Physical Dump

Physical dump does not provide the quiesce serialization mechanism, and it is not recommended for backing up mounted HFS data sets. If you do perform a physical dump of an HFS, do not specify the SHARE keyword. The SHARE keyword applies to the SYSDSN ENQ, and therefore does not provide protection against updates during dump.

**Attention:** Exercise caution if you use TOL(ENQF) during a physical dump of HFS data sets. Unlike other types of data sets, if an HFS is updated during a physical dump with TOL(ENQF), a subsequent restore will likely result in an unusable data set.

### Dumping Multivolume Data Sets

An important advantage of DFSMSdss as a backup tool is that it can back up multivolume data sets without having to specify any or all of the input volumes. If you do not specify any input volumes (you are using catalog filtering), multivolume data sets will be automatically processed in their entirety. The catalogs are scanned to select an entire data set; that is, the data set is processed in its entirety from all the volumes it resides on. Logical processing consolidates the extents of the data set in one dump data set for you.

If you specify input volumes (with either the LOGINDDNAME or LOGINDYNAM volume list), a data set is selected based on the following criteria:

- When you either specify SELECTMULTI(ALL) or specify input volumes without specifying the SELECTMULTI keyword, **all** of the volumes that contain a part of a non-VSAM or VSAM cluster must be in the volume list.

For VSAM data sets:

- If SPHERE is not specified, all parts of the base cluster or alternate index must be contained in the volume list.
- If SPHERE is specified, all parts of the base cluster can be contained in the volume list.
- When you specify SELECTMULTI(ANY), **any part** of the non-VSAM data set or VSAM base cluster can be on a volume in the volume list. For VSAM data sets:
  - If SPHERE is not specified, any part of the base cluster or alternate index can be contained in the volume list.
  - If SPHERE is specified, any part of the base cluster can be contained in the volume list.
- When you specify SELECTMULTI(FIRST), the volume list must include the volume that contains the **first part** of either the non-VSAM data set or the primary data component of the base cluster for a VSAM sphere. For VSAM data sets:
  - If SPHERE is not specified, the volume list must include the volume containing the first extent of the data component for the base cluster or the volume containing the first extent of the data component of an alternate index.
  - If SPHERE is specified, the volume list must include the volume containing the first extent of the data component for the base cluster.

The following is an example of the DUMP command with SELECTMULTI specified:

```
DUMP -  
  DATASET(INCLUDE(**)) -  
  SELECTMULTI -  
  LOGINDYNAM(338001) -  
  OUTDDNAME(TAPE) -  
  COMPRESS
```

SELECTMULTI works only for logical data set dumps. If you dump a multivolume data set physically, you must ensure that the segments from all the volumes are dumped together. If you dump a multivolume data set physically, it is dumped from all the volumes that are passed. The output dumped data contains a logical file for each selected volume.

A DFSMSdss logical data set dump operation attempts to ensure that all parts of a multivolume non-VSAM data set exist. In cases where a part of the data set is missing, such as an inadvertent scratching of the VTOC entry on a volume, DFSMSdss issues an error message and discontinues processing the data set.

DFSMSdss cannot process the following non-VSAM data sets because they are missing one or more parts:

- Multivolume data sets whose catalog volume order differs from the VTOC volume order
- Single-volume data sets with the same name that are cataloged as one multivolume data set
- Multivolume data sets whose last volume indicator in the VTOC entry is not set

A multivolume data set standard user label is not supported.

## Dumping Integrated Catalog Facility User Catalogs

Another important use of DFSMSdss as a backup tool is the backing up of integrated catalog facility user catalogs and their aliases (using logical data set dump). The user catalog name must be fully qualified with the INCLUDE keyword on the DUMP command. The LOCK attribute of an integrated catalog facility user catalog is dumped. The LOCK status is preserved if the catalog does not exist at restore time. Otherwise, the LOCK status of the existing catalog is used. Refer to *OS/390 DFSMS: Managing Catalogs* for a description of the LOCK attribute.

The following example shows the JCL used to dump an integrated catalog facility user catalog. RACF access to the catalog is not required if you have RACF DASDVOL update access or if the installation authorization exit routine bypasses authorization checking. Refer to *OS/390 DFSMS Installation Exits* for more information about the installation authorization exit routine.

```
//STEPT006 EXEC PGM=ADRSSU
//SYSPRINT DD SYSOUT=*
//TAPE DD DISP=(NEW,PASS),LABEL=(1,SL)
        VOL=SER=(A00760),DSN=OW22673.DUMP,
        UNIT=3480,DCB=(BLKSIZE=32760)
//SYSIN DD *
        DUMP DS(INC(TEST.CAT.PUBSEXMP)) -
            OUTDDNAME (TAPE)
/*
```

Figure 1 shows printed output produced by the dump.

```
PAGE 0001      5695-DF175  DFSMSDSS V2R10.0 DATA SET SERVICES      1999.211 14:54
  DUMP
    DS(INCL(TEST.CAT.PUBSEXMP)) -
    OUTDDNAME (TAPE)
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP '
ADR109I (R/I)-RI01 (01), 1999.211 14:54:32 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 1999.211 14:54:32 EXECUTION BEGINS
ADR801I (001)-DTDSC(01), DATA SET FILTERING IS COMPLETE. 1 OF 1 DATA SETS WERE SELECTED: 0 FAILED
    SERIALIZATION AND 0 FAILED FOR OTHER REASONS.
ADR454I (001)-DTDSC(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
    CLUSTER NAME TEST.CAT.PUBSEXMP
    CATALOG NAME SYS1.MVSRES.MASTCAT
    COMPONENT NAME TEST.CAT.PUBSEXMP
    COMPONENT NAME TEST.CAT.PUBSEXMP.CATINDEX
ADR006I (001)-STEND(02), 1999.211 14:54:32 EXECUTION ENDS
ADR013I (001)-CLTSK(01), 1999.211 14:54:32 TASK COMPLETED WITH RETURN CODE 0000
ADR012I (SCH)-DSSU (01), 1999.211 14:54:32 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
```

Figure 1. Output from a Dump of an Integrated Catalog Facility User Catalog

## Dumping Non-VSAM Data Sets That Have Aliases

DFSMSdss does not support INCLUDE filtering of non-VSAM data sets using an alias. To include a non-VSAM data set that has an alias for dump processing, you must use the data set's real name, as shown in the VTOC. DFSMSdss does not detect or preserve aliases of non-VSAM data sets. You will need to redefine the aliases after the data set is dumped and restored.

## Dumping VSAM Spheres

Using the SPHERE keyword, you can dump an entire VSAM sphere (base cluster and all associated alternate index clusters and paths). To dump the base cluster and the other components, all you need to specify is the base cluster name.

An example of the DUMP command with the SPHERE keyword is:

```
DUMP -  
    OUTDDNAME(TAPE) -  
    DATASET(INCLUDE(PARTS.VSAM1)) -  
    SPHERE -  
    PSWD(PARTS.VSAM1/MASTUPW1) -  
    COMPRESS
```

**Note:** You should be aware that you cannot restore a sphere unless it is dumped as a sphere with the SPHERE keyword.

## Dumping Indexed VSAM Data Sets

Indexed VSAM data sets (such as key sequenced or variable relative record data sets) can be logically dumped either without regard for the track contents or with validity checking of each track as the tracks are written. If dumped in the latter format, they must be restored on a system that supports the validate function.

The VALIDATE keyword, which is the default, specifies that the index and data track contents are to be validated as the tracks are dumped. Spanned record errors and split errors are detected and reported, but the dump continues. If other errors are detected, a message is issued and the dump stops.

The validate function can be overridden with the NOVALIDATE keyword, which specifies that no validation is done as the tracks are dumped. Some errors may not be detected until the data set is restored.

**Note:** Extended-format VSAM data sets cannot be dumped with the NOVALIDATE keyword.

## Dumping SYS1 System Data Sets

With the PROCESS keyword in conjunction with the SYS1 subkeyword, you can delete data sets with a high-level-qualifier of SYS1, except for a VVDS or a VTOCIX, when performing a logical or a physical data set dump.

To limit the use of PROCESS, you need to set up a RACF class profile. Refer to *OS/390 SecureWay Security Server RACF Security Administrator's Guide* for more information on RACF facility class profile.

## Dumping Data Sets Containing Records Past the Last-Used-Block Pointer

Some data sets on your system may contain records past the last-used-block pointer in the data set's VTOC entry. This could be a result of a data set not being properly closed or an application that accesses data in such a way as to bypass the updating of this field. In this case, special consideration needs to be given to these data sets as DFSMSdss recognizes this block pointer as the end of the used space in the data set and, therefore, the end of the real data.

Using the ALLDATA or ALLEXCP keyword will result in all the allocated space being dumped for applicable data sets. This includes all the data up to the last block pointer as well as all the data to the end of the allocated space. However, whether or not all the data is restored depends on data set characteristics and device characteristics during the restore. For example, if the data set must be reblocked (either because the target is an unlike device, the REBLOCK keyword is specified, or the data set is marked reblockable) only the used space will actually be restored. This limitation is due to the fact that any residual data (that is in the unused portion of the data set) will likely have different characteristics than the real data (that is in the used portion of the data set). This inconsistency would result in data incompatibilities causing the restore to fail, and thereby inhibiting the ability to restore the real data. Because of this, DFSMSDss will only restore the data in the used portion of the data set when the data characteristics must change.

If you require that all of the unused space is restored, then you should ensure that the data set is restored to a like device type and not reblocked or compressed. (Compress is the default for PDS data sets on restore unless you use the NOPACKING keyword.) In this case, the characteristics of the data do not change, and DFSMSDss will restore all the allocated space.

## Backing Up SMS-Managed Data Sets

When backing up data sets in an SMS-managed environment, you need to think about some special conditions in addition to those discussed under “Backing Up Data Sets” on page 33. The following sections discuss how you can back up SMS-managed data sets in an SMS-managed environment.

In most cases, you should let DFSMSHsm back up SMS-managed data sets for you. However, if you do not have DFSMSHsm or if you prefer not to rely on it for all your backup requirements, you can use DFSMSDss to back up your SMS-managed data sets.

### Filter on Class Names

DFSMSDss can select data sets for dump processing based on their storage, management, and data class names. Because management class is the construct that contains a data set’s availability attributes, you might want to filter on it when selecting data sets for dump processing.

If you want to back up data sets in a particular management class, you can filter on the management class name. For example, if you want to perform incremental backup on data sets in management classes MCNAME1 and MCNAME2, specify the DUMP command as follows:

```
DUMP -  
  DATASET(INCLUDE(**) -  
    BY((MGMTCLAS,EQ,(MCNAME1,MCNAME2)) (DSCHA,EQ,YES))) -  
  OUTDDNAME(OUTVOL1)
```

### Class Names Saved

DFSMSDss saves the class names of the data sets it dumps. These names are then used as input to ACS routines when the data set is restored.

---

## Backing Up Volumes

With DFSMSdss, you can back up volumes either logically or physically. If the volume is to be restored to an unlike device, you must dump it logically.

### Logical Volume DUMP

To perform a logical volume dump, you specify DATASET(INCLUDE(\*\*)) with either LOGINDDNAME or LOGINDYNAM. LOGINDDNAME identifies the input volume that contains the data sets to be dumped. LOGINDYNAM specifies that the volumes containing data sets to be dumped be dynamically allocated.

Here is an example of how you specify the DUMP command to perform a logical volume dump:

```
DUMP DATASET(INCLUDE(**)) -  
      LOGINDDNAME(DASD1) -  
      OUTDDNAME(TAPE)
```

**Note:** Certain data sets can be restored only to like devices even though they were dumped logically.

### Physical Volume Dump

To perform a physical volume dump, specify the DUMP command with INDDNAME or INDYNAM and OUTDDNAME. Because FULL is the default keyword for the DUMP command, you need not specify it. Unallocated tracks are not dumped. The following example shows how you can specify the DUMP command to physically back up a volume:

```
DUMP INDDNAME(DASD1) OUTDDNAME(TAPE)
```

## Backing up System Volumes

If you plan to use Stand-Alone restore to restore a volume (with no operating system), the volume must be dumped physically. In addition, when doing a full physical volume dump to back up a system residence volume, you must use JCL to invoke DFSMSdss.

## Backing up VM-Format Volumes

You can use DFSMSdss to back up VM-format volumes that are accessible to your MVS system. The volumes must have OS-compatible VTOCs starting on track zero, record five. DFSMSdss can only retrieve device information from the OS-compatible VTOC; it cannot interpret any VM-specific information on the volume.

Use the CPVOLUME keyword and specify the range of tracks to be backed up with the TRACKS keyword. You can use concurrent copy on VM-format volumes by specifying the CONCURRENT keyword. Because DFSMSdss cannot check access authorization for VM data, CPVOLUME is only allowed with the ADMINISTRATOR keyword.

Exercise caution when using DFSMSdss to back up VM-format volumes, because DFSMSdss does not serialize any VM data in any way. Dumps of VM-format volumes cannot be restored with Stand-Alone Restore.

---

## Dumping Data Efficiently

When backing up data, you can specify both the `OPTIMIZE` and the `COMPRESS` keywords to improve performance and save dump space. The two keywords can be used together.

A selective data set dump operation saves space, but a full-volume dump operation saves time. The same applies to the `COMPRESS` keyword. It saves dump space, but involves some processing overhead. In general, if you are dumping to tape, saving space is probably less of a concern than performance. Usually, saving space is important only when it results in using fewer tapes to store the data because that, in turn, reduces the number of tape mounts necessary to recover the data.

## Space Considerations

Using larger block sizes saves dump space and improves performance by minimizing the number of I/O operations performed during a dump operation.

The default block size for output records written to **tape** is 65 520 bytes (65 520 is also a maximum). You can change this default to 32 760 bytes by using the installation options exit routine. Refer to *OS/390 DFSMS Installation Exits* for more information.

For output records written to **DASD**, the block size is the track length of the output volume for devices whose track length is less than 32KB, or one half the track length for devices whose track length is greater than 32KB. You can select a different block size for tape or DASD by coding `DCB=BLKSIZE=block size` in the corresponding DD statement. The minimum block size you can select is 7892 bytes; the maximum is 32 760 bytes.

**Note:** To have the block size specification included in the tape label, you must specify the `BLKSIZE` parameter in the DD statement for the tape.

You can also save dump space by doing the following:

- Dump only the used space (the default if you do not use keywords `ALLDATA` or `ALLEXCP`), instead of all allocated space, in sequential and partitioned data sets or in data sets with a null `DSORG` field. For VSAM key sequenced data sets, the `VALIDATE` keyword (which is the default) dumps only the used data instead of all allocated space.
- Use the `COMPRESS` keyword.

### Notes:

1. The `COMPRESS` keyword is ignored if it is specified during a logical data set dump for physical sequential extended-format data sets.
  2. If you have a 3480 or 3490 tape drive with the improved data recording capability (IDRC) and you want to use hardware data compaction, you do not need to use the `COMPRESS` keyword with the `DUMP` command. If you want software compression, specify the `COMPRESS` keyword, but you do not need to specify `DCB=TRTCH=COMP` in the JCL. In most cases, hardware data compaction without software data compression gives the best performance. However, you can use hardware compaction and software compression at the same time.
- Perform incremental data set backup instead of volume backup. This reduces the amount of data dumped and decreases processing time.

## Performance Considerations

This section provides tips for improving the performance of copy and dump operations.

### DUMP

Dump to tape, where the larger block size reduces the number of I/O operations.

- Use OPTIMIZE(2), (3), or (4) to read more than one track per read operation. This results in the reading of two tracks, five tracks, or a full cylinder, respectively. The default, OPTIMIZE(1), reads one track at a time. OPTIMIZE(2), (3), or (4) results in less elapsed time and fewer I/O operations on the DASD device whenever the load on the tape channel is low enough and the tape speed is high enough to keep pace with the data being read from the DASD volume.
- Use the PARALLEL feature to dump multiple DASD volumes simultaneously.

### Concurrent Copy

To get an exact copy of your data at a specific time, do not update it during the concurrent copy initialization.

Concurrent copy initialization includes the time DFSMSdss spends filtering data sets. Therefore, the more precisely you specify the data sets to be processed, the sooner the initialization is completed and the sooner you can update your data again. Here are some ways to reduce initialization time:

- Keep data to be dumped by one DUMP command cataloged in one catalog, if possible.
- Do not specify the DYNALLOC keyword if you do not need dynamic allocation for your data sets.
- Specify fully or almost fully qualified data set names. This reduces the time that DFSMSdss searches catalogs for the data sets to be processed.
- Specify smaller groups of data sets to be processed together on a DFSMSdss operation.
- Minimize the use of wildcards with the INCLUDE keyword.
- Minimize the use of sophisticated BY filtering to determine the data to be processed.
- Ensure that DFSMSdss can obtain serialization on all data sets being processed.
- Specify WAIT(0,0) to prevent DFSMSdss from waiting for serialization when it cannot be obtained. Refer to *OS/390 DFSMSdss Diagnosis Guide*.
- Do not specify the NOTIFYCONCURRENT keyword if you do not need notification of each data set included in the concurrent copy session.
- Do not specify the SPHERE keyword if you are not processing VSAM spheres.
- Use the ADMINISTRATOR keyword or DASDVOL RACF protection (where applicable) to bypass authorization checks for each data set being processed.
- Ensure that the volumes containing the VTOC and catalog entries for the data sets to be processed have caching enabled. Also ensure that the catalogs involved are enabled for the in-storage cache (ISC) or the catalog data space cache (CDSC).
- Ensure that data sets being processed have not been migrated by DFSMSHsm.

Concurrent copy uses storage in the control unit cache and in the processor. Here are some ways to minimize the storage needed:

- Limit the amount of data included in the concurrent copy operation.
- Use concurrent copy during periods of low update activity (as most backups are currently done today).

- Concentrate the update activity in a subset of the data being processed by concurrent copy.

### **Concurrent Copy Storage Requirements**

The concurrent copy support for the 3990 Model 6 Storage Control uses data spaces to contain track image copies of the data being processed by the DFSMSDss. MVS data spaces are backed by expanded storage and local paging spaces. The amount of expanded storage and local paging space required for concurrent copy usage is dependent on a number of variables. Based on simulations and test scenarios, a typical data space size is about 10% of the amount of data being dumped or copied with concurrent copy.

If your data space size exceeds this nominal value, you may need to consider the following planning guidelines for determining how much expanded storage or local paging space may be required for the following concurrent copy functions:

- Full volume and tracks copy; and full volume, tracks and physical data set dump operations:

All volumes are processed on a track-by-track basis by DFSMSDss. The data space requirements can vary from 0% for a volume that has no updates during the DFSMSDss operation to 100% if the entire volume is updated before DFSMSDss can process it. For example, a 3390-3 that is 80% full (2671 cylinders) may require up to 2671 cylinders of data space storage if the volume is completely rewritten before DFSMSDss can process it. An example of this situation would be that a volume contains many VSAM data sets and a reorganization is done for all of the VSAM data sets on the volume while the concurrent copy job is being run for the volume.

- Logical data set copy and dump processing of non-VSAM data sets and nonindexed VSAM data sets (for example, VSAM ESDS), logical data set copy of indexed VSAM data sets (for example, VSAM KSDS), and logical data set dump of indexed VSAM data sets processed with NOVALIDATE are described as follows:

These data sets are processed on a track-by-track basis by DFSMSDss. The data space is used to contain updates for tracks that have not yet been processed by DFSMSDss. The data space requirements can vary from 0% for a data set that has no updates during the DFSMSDss operation to 100% if the entire data set is updated before DFSMSDss can process it. For example, a 50-cylinder data set may require up to 50 cylinders of data space storage if the data set is completely rewritten before DFSMSDss can process it.

- Logical data set dump of indexed VSAM data set (for example, VSAM KSDS) processed with VALIDATE is described below:

These data sets are processed with numerous accesses to sequence set information in the index component and track-by-track accesses to the data component. In all cases, update activity to either the data component or the index component maintains a copy of the updated track in the data space until the track is either processed by DFSMSDss or the dump operation is ended for all data sets.

Index component tracks that do not contain sequence set information and data component tracks that are beyond the high used relative byte address are included in the concurrent copy operation but are never read by DFSMSDss. If those tracks are updated, they will remain in the data space for the duration of the dump operation for all data sets. If the data set has the sequence set information imbedded in the data component (using the IMBED attribute), no additional (nonupdated) tracks are maintained in the data space. If the data set has the sequence set information in the index component, then all index

component tracks containing sequence set information will be maintained in the data space (whether they were updated or not) for the duration of the dump processing for the data set. For example, if the index for a VSAM data set is 20 cylinders and the data is 2500 cylinders, plan paging space of 20 cylinders for the index component.

Based on the update activity during the dump operation, plan to use a paging space of between 0 and 2500 cylinders for the data. The most data space is used when doing a complete reorganization while dumping the VSAM data set. This requires 2520 cylinders of space. If only 10% of the data will change during the operation, you will need 20 cylinders for the index and 250 cylinders for the data or 270 cylinders of paging space.

In using concurrent copy against aggregate groups, determine the data space storage requirements based on the expected update rate to the data sets during the dump operations. Failure to allocate sufficient local paging space may result in system failures due to insufficient paging storage.

**Note:** All storage requirements will be in addition to the working set of storage required by *all* other applications active (including all other concurrent copy operations) during the execution of the DFSMSdss concurrent copy operation.

### **Virtual Concurrent Copy Working Space**

To perform a snapshot copy, space is required on one or more volumes in the same RAMAC Virtual Array (RVA) subsystem as the source data set. Ensure that space is available before using the concurrent keyword on DFSMSdss commands referring to data sets on an RVA subsystem.

Allocation of some number of data sets to be used as working space by virtual concurrent copy is necessary. The naming convention for these data sets is:

`SYS1.ANTMAIN.Ssysname.SNAPnnnn`

Variable *sysname* is the system identifier and *nnnn* is a four-digit decimal number in the value range 0001–9999. If the system identifier is eight characters, 'S' replaces the first character.

A numerically sequential catalog search is done for each data set, starting with `SYS1.ANTMAIN.Ssysname.SNAP0001` until a catalog locate error is encountered, indicating that the data set was not found. The working space data sets must be cataloged. Data sets with this naming convention beyond the data set which was not found will not be used as working space. Data sets must be allocated as physical sequential and must not be extended-format. The data sets may be SMS-managed or non-SMS-managed. If the catalog search for a working space data set indicates that the data set is multivolume, the data set will not be used as a working space data set.

System Data Mover (SDM) does not extend a working space data set. If secondary space allocation is desired, the user must extend the data set by filling it with data prior to starting the DFSMSdss job. SDM holds an enqueue for the data set while the working space is being used during a snapshot operation; it then releases the enqueue after all usage of the data set is completed. A working space data set may be reallocated or extended when SDM does not have the data set enqueued. On subsequent use of the reallocated or extended working space data set by SDM, the new size of the data set will be used by SDM. Additional working space data sets may be added after ANTMAIN has completed the initialization process.

SDM uses these data sets the first time an out-of-working-space condition is encountered during a snapshot operation. When this condition occurs, SDM refreshes the list of working space data sets by performing a catalog search that starts with SYS1.ANTMAIN.Ssysname.SNAP0001.

The LRECL and block size may be any valid combination. The tracks within the data set are used as the target of snapshot operations, and you should not try to access them using normal data access methods.

**Note:** It is recommended that the working space data sets be protected by security, such as RACF, to ensure that sensitive data is not made available to unauthorized users.

Data sets must be allocated on some volume in each RVA subsystem which will be used for virtual concurrent copy. If more than one device type is defined on the RVA subsystem, a working data set must be allocated on each device type that contains a data set that will be processed using DFSMSdss snapshot support.

If each system and each device type for DFSMSdss jobs (with concurrent copy specified) runs simultaneously from more than one system and accesses data on the same RVA subsystem, it requires at least one work data set each. For example, DFSMSdss must allocate three working space data sets to process data on an RVA subsystem from three MVS systems, on devices of each device type containing data processed with concurrent copy.

The total size of all work data sets allocated on each RVA subsystem should be equal to or exceed the largest total amount of data to be processed in a single DFSMSdss COPY or DUMP command on that RVA. If there is insufficient space, the concurrent copy initialization for one or more data sets in the job fails.

### **Read DASD I/O Pacing**

You can tune the performance of a system by pacing the DFSMSdss read DASD I/O operations. Pacing reduces the channel utilization and lets other I/O (for example, from the data base application) be processed in a more timely fashion. The pacing is done by waiting a specified amount of time before issuing each channel program that reads from DASD.

**Note:** The additional wait time does not apply to error recovery channel programs or concurrent copy I/O. The System Data Mover dynamically controls pacing for concurrent copy I/O.

**Invocation from a Customer Program:** The value of the READIOPACING parameter can also be controlled through the installation options exit, a product-sensitive programming interface intended for customer use. Refer to *OS/390 DFSMS Installation Exits* for more information.

## **Shared DASD Considerations**

Shared DASD presents volume and data set serialization problems not encountered in nonshared DASD environments. Care should be taken when you enlist data set operations if programs operating in another processor might be accessing the data sets at the same time.

A data set can be dumped from one processor while being processed from another. The dumped version may be partially updated on JES2 systems. This same exposure is present on a full dump operation.

---

## Restoring Data Sets

With the RESTORE command, you can restore data to DASD volumes from DFSMSdss-produced dump volumes, which are identified with the INDDNAME keyword.

The restore function is logical or physical, depending upon the dump volume. If the dump volume was made physically, a physical restore is made. If it was made logically, a logical restore is made. If the data was compressed when it was dumped, it is automatically expanded to its original form during the restore operation.

Using the ALLDATA or ALLEXCP keyword during a dump affects target data set allocation during a restore. Only used space is dumped for both a physical and logical data set dump unless the ALLDATA or ALLEXCP keyword is specified as part of the DUMP command.

When ALLDATA or ALLEXCP is specified, the total allocated space is dumped. During a physical data set restore, the target data set is allocated with the same amount of space as the source data set. During a logical data set restore (without the ALLDATA or ALLEXCP keyword), the target data set is allocated according to the amount of space used in the data set, thereby releasing unused space. If logical data set processing is used and the target data set must preserve the total allocation of the source, the ALLDATA or ALLEXCP keyword should be specified during the dump.

When ALLDATA or ALLEXCP is specified for an extended-sequential data set, data beyond the last-used-block pointer is not retained. The target data set is allocated with the same amount of space as the source data set during a logical restore or copy operation.

As with a data set DUMP command, you can use filtering to select data sets for restore processing. DFSMSdss reads the entire dump data set once during a restore regardless of how much data is actually being restored. This will result in multiple tape mounts if the dump data set is on multiple tapes. See “Choosing Data Sets for Processing—Filtering” on page 18, for more information on filtering.

**Note:** Fully qualified names are required to restore the following data sets:

- VVDS
- VTOCIX
- SYS1.STGINDEX
- Integrated catalog facility catalogs
- OS catalog
- VSAM read-only data sets (temporarily exported with the INHIBITSOURCE parameter)

## Logical Data Set Restore

A logical data set restore is performed if you are restoring from a volume created with a logical dump operation and if you specify the DATASET keyword. For instance, the following RESTORE command generates a logical data set restore if the volume was created with a logical dump operation.

```
RESTORE INDDNAME(TAPE) -  
        DATASET(INCLUDE(USER1.OLDDS)) -  
        REPLACE
```

**Note:** DFSMSDss logical restore processing cannot be used to process partitioned data sets containing location-dependent information that does not reside in note lists or the directory.

### Output Volume Selection

In most cases, specifying output volumes is optional for a logical data set RESTORE command. Output volume specification is required only if the data set:

- Exists and is to be restored to a volume that is different from the current location
- Does not exist and is to be restored to a volume that is different from the source volume.

Specify output volumes with the OUTDDNAME or OUTDYNAM keywords. An example of a logical data set RESTORE command with OUTDDNAME is:

```
RESTORE -  
      INDDNAME(TAPE) OUTDDNAME(DASD1) -  
      DATASET(INCLUDE(**))
```

When not specified, the volume on which the source data set currently resides is found from the catalog and is dynamically allocated. To do this, you must include the REPLACE keyword. This is particularly useful on a data set restore operation from a data-set-selection-by-catalog dump because you need not know where the data sets resided at dump time.

You can specify multiple output DASD volumes on a logical data set RESTORE command. This is required when all the data sets to be restored cannot fit on a single volume. An example of a logical data set restore operation with a spill volume specified is:

```
RESTORE -  
      INDDNAME(TAPE) OUTDYNAM((338001),(338002)) -  
      DATASET(INCLUDE(PARTS.**)) -  
      PCTU(80)
```

Note the use of the PERCENTUTILIZED (PCTU) keyword in the above example. With PERCENTUTILIZED, you can set a limit on the amount of space DFSMSDss can fill on the volume. When this limit is reached, subsequent data sets are allocated to other volumes. In the above example, PERCENTUTILIZED is used to specify that only 80% of the first target volume is to be filled. This leaves 20% free space for the data sets to extend, if necessary.

PERCENTUTILIZED is ignored for SMS-managed volumes.

**Note:** User data-set labels on DASD volumes are supported during a data set restore operation. However, either the data set on both the source and target volumes must have these labels, or neither must have them.

### Restoring Preallocated Target Data Sets

In some instances, you might want to control the placement of a data set on a volume when you restore it. Some data sets (such as data sets allocated by absolute track) have location-dependent data and must be preallocated. Others (such as catalogs) should be placed for performance reasons. For details on restoring such data sets, see “Restoring Indexed Sequential, Unmovable, Direct, and Absolute Track Data Sets” on page 55 and “Restoring Integrated Catalog Facility Catalogs” on page 54.

In order to use a preallocated data set, the REPLACE keyword must be specified and the preallocated target data set name must be identical to the source data set name.

If a target data set is preallocated, it is scratched and reallocated if it is not large enough to contain the dumped data set. VSAM preallocated target data sets are also scratched and reallocated when:

- Any of the following source and target data set attributes do not match:
  - CI size
  - Record length
  - IMBED (only KSDS and key range data sets)
  - Key length (only KSDS and key range data sets)
  - REPLICATE (only KSDS and key range data sets)
  - SPANNED
- The preallocated target is multivolume and the space of the target data set on the first volume is not large enough to contain all of the dumped data.
- The data set was not defined as reusable and the high-used relative byte address (RBA) of a target VSAM KSDS is not 0.

During logical restore, a compression is performed when partitioned data sets are restored to both like and unlike devices. If the partitioned data set is being restored to an unlike device, the device-dependent information (such as TTR pointers and note lists) is in a usable form after the restore. DFSMSDss is unable to resolve device-dependent information for all other data set types being restored to unlike devices.

The NOPACKING keyword is effective only for partitioned data sets. If NOPACKING is specified for preallocated partitioned data sets, the preallocated target must reside on the same or a like device. Processing is stopped for the data set if the target resides on an unlike device. The target is not deleted and reallocated.

### **Cataloging Data Sets During Logical Restore Processing**

When you restore a data set, you might need to catalog it in the standard order of search or recatalog it in its original catalog. The CATALOG keyword catalogs the data set in the standard order of search. The RECATALOG(\*) keyword catalogs it in the same catalog that points to the source data set.

When a data set is restored as an SMS-managed data set, it is cataloged using the standard order of search. The RECATALOG keyword is ignored.

Examples of the CATALOG and RECATALOG keywords in a logical data set RESTORE command follow:

```
RESTORE -  
  INDDNAME(TAPE) -  
  DATASET(INCLUDE(USER1.**)) -  
  CATALOG
```

```
RESTORE -  
  INDDNAME(TAPE) -  
  DATASET(INCLUDE(USER1.**)) -  
  RECATALOG(*)
```

When a VSAM KSDS or key range data set is being restored to an unlike device, the data set must be cataloged in the standard order of search.

## Renaming Data Sets during Logical Restore Processing

In addition to cataloging data sets when they are restored, you can rename restored data sets by using the RENAME keyword. For instance, you can code the following to rename a data set you are restoring:

```
RESTORE -  
  INDDNAME(TAPE) -  
  DATASET(INCLUDE(USER2.OLDDS)) -  
  RENAME(*.OLDDS,*.NEWDS)
```

**Note:** The RENAME keyword works only if the data set exists on the output DASD with the old name. If you really want to unconditionally rename a data set, use RENAMEU. Both VSAM and non-VSAM data sets can be renamed. The rules for renaming VSAM clusters are the same as for non-VSAM data sets. You can rename only clusters. DFSMSdss assigns a new name for the components of VSAM clusters. SMS considerations require DFSMSdss to ensure that VSAM component names resolve to the same catalog as the cluster name. DFSMSdss uses the cluster name as a guide to determine the component names. This applies equally to SMS and non-SMS data sets.

## Expiration Date Handling during Logical Restore

For preallocated targets, the expiration date of the preallocated target is preserved. For non-preallocated targets, the expiration date is dependent upon whether the data set is VSAM or non-VSAM, whether the source data set is SMS-managed or not SMS-managed, and whether the target data set is SMS-managed or not SMS-managed. SMS also ensures that the expiration date conforms with the target's management class retention period.

### Allocating to SMS

VSAM data sets: The expiration date from the source catalog entry is used to set the target's expiration date in both the catalog and the VTOC. For an indexed VSAM data set, the expiration date in the VTOC for the index component will be zero.

Non-VSAM data sets: The expiration date from the source VTOC is used to set the target expiration date in both the catalog and VTOC. If the expiration date violates the target's management class retention period, SMS will modify the date to conform with the management class.

### Allocating to non-SMS

VSAM data sets: The expiration date from the source catalog entry is used to set the target's expiration date in the catalog. The target expiration date in the VTOC will be 99365. For an indexed VSAM data set, the expiration date in the VTOC for the index component will also be 99365.

Non-VSAM data sets: The expiration date from the source VTOC is used to set the expiration date in the target VTOC. If the target is cataloged, the expiration date in the catalog is set to the date from the source VTOC if the source data set is SMS-managed. If the target is cataloged, and the source data set is not SMS-managed, the expiration date in the catalog is not set.

## Physical Data Set Restore

A physical data set restore is done if you are restoring from a dump volume created by physical dump processing and you specify the DATASET keyword. If the dump

volumes resulted from a physical data set dump operation, you must do a physical data set restore or a tracks restore operation. A tracks restore operation can consist of a subset of the dump data.

**Notes:**

1. When you perform a physical restore of many data sets, there is an initial delay while DFSMSdss allocates the target data sets.
2. To prevent index components from being restored inadvertently, you must specify the fully qualified name of the cluster.

On a physical data set restore operation, data sets from one or more logical volumes can be restored to a single DASD volume. If you want to restore data sets from specific source DASD volumes, use the LOGICALVOLUME keyword to specify the volume serial numbers of the source DASD volumes you want to restore. For example:

```
RESTORE -  
      INDDNAME(TAPE) OUTDDNAME(DASD1) -  
      DATASET(INCLUDE(**)) LOGICALVOLUME(111111) -  
      REPLACE
```

The following data sets cannot be processed by physical data set DUMP or RESTORE operations:

- VSAM data sets not cataloged in an integrated catalog facility catalog.
- Extended-format VSAM data sets. Use logical processing for these types of data sets.
- Page, swap, and SYS1.STGINDEX data sets.

**Note:** Data from a specific volume can be restored only to a DASD volume of like device type.

### Output Volume Selection

For a physical data set RESTORE command, you must specify an output volume with the OUTDDNAME or OUTDYNAM keyword. A physical data set restore operation restores only to the first volume in a list passed in the OUTDDNAME or OUTDYNAM parameter.

### Cataloging Data Sets During Physical Restore Processing

If you specify CATALOG on a physical data set restore operation, DFSMSdss creates catalog entries for single-volume, non-VSAM data sets that were allocated by DFSMSdss. The cataloging is done immediately after successful allocation of a data set. Failure in cataloging does not prevent the data set from being restored. A data set that was allocated and cataloged but encountered errors during the restore operation is neither uncataloged nor scratched by DFSMSdss. You must not specify the RECATALOG keyword for physical restore.

The catalog that DFSMSdss uses to catalog a data set is determined as follows:

- The catalog specified by a STEPCAT or JOBCAT (but not both) DD statement.
- If the STEPCAT or JOBCAT DD statement specifies concatenated catalogs, only the first catalog in the concatenation is used.
- If no STEPCAT or JOBCAT DD statement is provided and the first qualifier of the data set name is an alias for a user catalog, the catalog pointed to is used for that data set.
- Otherwise, the master catalog is used.

DFSMSdss does not catalog VSAM data sets during physical restore processing. If the CATALOG keyword is specified, it is ignored when processing VSAM data sets. You should use the IDCAMS DEFINE RECATALOG command to catalog VSAM data sets that were allocated by DFSMSdss (not preallocated). To recatalog and later access the VSAM data set, the volume serial numbers for the target and source volumes must match and the data set must be cataloged in the same catalog from which it was dumped. The volume serial number and the catalog name are printed in message ADR4181 during restore.

**Note:** To catalog multivolume non-VSAM data sets, use the IDCAMS DEFINE NONVSAM command.

---

## Restoring Data Sets with Special Requirements

Some data sets have special requirements for being restored. The sections that follow describe some of the special cases you might encounter when you restore data sets.

### Restoring Multivolume Data Sets and Restoring Data Sets Using Multiple Target Volumes (Spill Volumes)

Multivolume data sets from a logical data set dump tape can be restored either to a single volume or to multiple volumes. When they are not preallocated and the specified output volumes are different from the input volumes, multivolume data sets are restored to a single volume, space permitting. When multiple target volumes are specified, DFSMSdss selects target volumes as follows:

- If a target volume that has the same volume serial as the source volume is available and has adequate space, it is chosen.
- If a volume of the *same* device type is available, and if it has adequate space, it is selected.
- A volume of a *like* device type is selected if it has adequate space.
- A volume of an *unlike* device type is selected if it has adequate space.

If you are restoring a multivolume data set from a physical dump, be sure the segments from all volumes are restored with successive RESTORE commands. Restoring a portion of a multivolume non-VSAM data set to a preallocated data set is allowed only if the volume sequence numbers of the source and target data sets are the same.

A VSAM data set that has its index component defined on more than one volume (that is, a multivolume KSDS defined with the imbed attribute) should always be processed logically. If it must be processed physically, it should be treated as an absolute track allocation data set, and its extents restored to their original location. This can be accomplished by performing either a full-volume restore, or a tracks restore of the relevant tracks. If this procedure is not done, the index may become unusable.

During logical restores of VSAM data sets whose data and index components are on different source volumes, DFSMSdss preserves the volume spread if enough target volumes of like device types are specified.

**Note:** DFSMSdss preserves the volume spread by placing the data and index components on separate devices only if all of the following are true:

- The source data and index components reside on separate devices.

- The target data set is preallocated with the data and index components on separate devices.
- DFSMSdss does not need to scratch and reallocate the preallocated target data set.

See “Restoring to Preallocated Targets” on page 56 for details on when DFSMSdss scratches and reallocates target data sets.

## Restoring Integrated Catalog Facility Catalogs

Integrated catalog facility user catalogs can be restored only to the same volumes (the same volume serial and the same device type) from which they were dumped. The component names of the source and target user catalog must be the same. In addition, you must specify the fully qualified name to restore a catalog.

Restore from a logical dump is generally the best way to restore a catalog because it restores user catalog aliases if they are present in the logical dump data set. For logical restore operations, user catalog aliases are restored as follows:

- If DFSMSdss allocated the user catalog, aliases are restored if the catalog is successfully restored.
- If the target catalog was preallocated and is not empty, aliases are not restored.
- If the target catalog was preallocated and is empty, aliases are restored.

A physical restore operation does not restore aliases, and physically dumped catalogs cannot be restored if they are open. In addition, if the entries in the catalog during the dump operation do not match the entries during the physical restore operation, some of the data sets may become inaccessible.

An integrated catalog facility user catalog can be restored dynamically. Catalog recovery jobs should be modified to include the IDCAMS ALTER LOCK command to lock the existing catalog before the DFSMSdss restore operation. After the recovery is complete, unlock the catalog using IDCAMS ALTER UNLOCK. The LOCK attribute on the dump tape is used if the catalog does not exist. For information on the LOCK attribute and the access authority, refer to *OS/390 DFSMS: Managing Catalogs*.

The following example shows the JCL used to restore an integrated catalog facility user catalog. If the master catalog is RACF-protected, RACF access to it is required, unless you have DASDVOL update access or the installation authorization exit routine bypasses authorization checking. Refer to *OS/390 DFSMS Installation Exits* for more information about the installation authorization exit routine.

```
//STEPT007 EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//TAPE DD DISP=OLD,LABEL=(1,SL)
        VOL=SER=(A00760),DSN=OW22673.DUMP,
        UNIT=3480
//SYSIN DD *
        RESTORE DS(INCL(TEST.CAT.PUBSEXP)) -
        OUTDYNAM ((D9S060)) -
        REPLACE -
        INDDNAME (TAPE)
/*
```

Figure 2 on page 55 shows the printed output produced by the RESTORE command.

```

PAGE 0001      5695-DF175 DFSMSDSS V2R10.0 DATA SET SERVICES      1999.211 14:54
RESTORE
  DS(INCL(TEST.CAT.PUBSEXMP)) -
  OUTDYNAM (D9S060)          -
  REPLACE                     -
  INDDNAME (TAPE)
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'RESTORE '
ADR109I (R/I)-RI01 (01), 1999.211 14:54:46 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 1999.211 14:54:46 EXECUTION BEGINS
ADR780I (001)-TDDS (01), THE INPUT DUMP DATA SET BEING PROCESSED IS IN LOGICAL DATA SET FORMAT AND WAS
CREATED BY DFSMSDSS VERSION 2 RELEASE 10 MODIFICATION LEVEL 0
ADR442I (001)-FRLB0(01), DATA SET TEST.CAT.PUBSEXMP PREALLOCATED, IN CATALOG SYS1.MVSRES.MASTCAT, ON
VOLUME(S): D9S060
ADR489I (001)-TDLOG(02), CLUSTER TEST.CAT.PUBSEXMP WAS RESTORED
      CATALOG      SYS1.MVSRES.MASTCAT
      COMPONENT    TEST.CAT.PUBSEXMP
      COMPONENT    TEST.CAT.PUBSEXMP.CATINDEX
ADR454I (001)-TDLOG(01), THE FOLLOWING DATA SETS WERE SUCCESSFULLY PROCESSED
      TEST.CAT.PUBSEXMP
ADR006I (001)-STEND(02), 1999.211 14:54:47 EXECUTION ENDS
ADR013I (001)-CLTSK(01), 1999.211 14:54:47 TASK COMPLETED WITH RETURN CODE 0000
ADR012I (SCH)-DSSU (01), 1999.211 14:54:47 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000

```

Figure 2. Output from Restore of Integrated Catalog Facility User Catalog

## Restoring Non-VSAM Data Sets That Have Aliases

DFSMSdss does not support INCLUDE filtering of non-VSAM data sets using an alias. To include a non-VSAM data set which has an alias for restore processing, you must use the data set's real name, as shown in the VTOC. DFSMSdss does not detect nor preserve aliases of non-VSAM data sets. You will need to redefine the aliases after the data set is restored.

## Restoring Indexed Sequential, Unmovable, Direct, and Absolute Track Data Sets

One important use of DFSMSdss is restoring data sets that contain device-dependent information. In some cases, such data sets can be restored without preallocating the target data sets. In other cases, however, you must preallocate the target in order to restore the data set.

### Restoring without Preallocated Targets

You can use a logical data set restore operation to restore an indexed sequential data set without preallocating it if the tracks on which it is to be restored are not occupied and if the volume has an indexed VTOC. If an indexed sequential or unmovable data set is not preallocated, DFSMSdss tries to allocate the data set on the same relative tracks that it was dumped from. If this allocation fails, the unmovable data sets are allocated to any available location if the FORCE keyword is specified. The FORCE keyword does not apply to indexed sequential data sets.

When you specify the FORCE keyword and some of the data sets have truly location-dependent data, you should specify their names in the EXCLUDE parameter to prevent DFSMSdss from restoring them. Subsequently, you must either restore the data set onto a scratch volume or free up the area of the DASD where these data sets were located on the source volume and rerun the restore operation. When restoring a direct, undefined data set and the target data set is not preallocated, DFSMSdss allocates the data set. The allocation may result in a new

data set with a different configuration from the data set that was dumped (for example, fewer volumes). This situation may cause problems when processing the restored data set.

### **Restoring to Preallocated Targets**

If you are restoring any of these types (indexed sequential, unmovable, direct, or absolute track) of data sets by preallocating them, the size and location of the extents on the dump volume and on the restore volume should match. Restoring a data set to a larger preallocated data set can cause problems because of the extraneous data beyond the end of the original dumped data. If the preallocated data set is too small, DFSMSDss deletes it and reallocates a new data set. The allocation may fail, or it may result in a new data set with a different configuration (for example, fewer volumes). The latter situation can cause problems processing the data set.

For unmovable data sets (those allocated as ABSTR, PSU, POU, or DAU), a preallocated data set is restored if the extents match and the REPLACE keyword is specified. Even if the extents do not match, the data set is restored if you specify both the REPLACE and FORCE keywords. When an unmovable data set cannot be restored, the extents of the data set on the source volume are listed so that you can take action to restore it.

**Note:** Indexed sequential data sets cannot be restored to unlike devices.

### **Restoring Direct Access Data Sets**

When DFSMSDss restores direct data sets, several processing options can be used. Direct data sets can be organized by relative block address or by track-track record (TTR).

Relative block addressable direct access data sets can be processed block by block to like and unlike target devices if the block size fits on the target track. When the data sets are processed block by block, DFSMSDss updates the block reference count of dummy records contained in the relative block addressed direct access data sets. To process block by block, the direct access data sets must have neither a variable record format nor a standard user label.

TTR direct access data sets may become unusable if they are processed block by block. TTR and relative block addressable data sets can be processed track by track to like and unlike target devices whose track capacity is equal to or greater than the source. Block by block processing is more efficient because track by track processing to an unlike device of larger track capacity can leave some unused space on each track of the target data set.

Several DFSMSDss keywords implement the BDAM processing options (refer to *OS/390 DFSMSDss Storage Administration Reference* for details on their use):

#### **AUTORELBLOCKADDRESS**

If the data set is accessed with Optional Services Code (OPTCD) indicating relative block addressing,<sup>4</sup> it is processed as if it were specified in the RELBLOCKADDRESS subkeyword list, and processing is block by block. If your installation has many relative block address direct access data sets, you may wish to consider

---

4. Refer to *OS/390 DFSMS Macro Instructions for Data Sets* for macro instructions for non-VSAM data sets.

the DFSMSdss installation options exit to turn on the AUTORELBLOCKADDRESS function (refer to *OS/390 DFSMS Installation Exits*).

#### **RELBLOCKADDRESS**

If the data set is specified in the subkeyword list, the data set is processed block by block.

#### **TTRADDRESS**

If the data set is specified in the subkeyword list, the data set is processed track by track.

#### **FORCE**

If the track capacity of the receiving volume is smaller than the source, FORCE may be required for variable or undefined length TTR-organized direct access data sets. These data sets may be unusable after restore and, if possible, should be restored to a like device. Use RELBLOCKADDRESS to restore relative block address direct access data sets to unlike devices.

**Note:** If you do not specify a keyword, data is moved to the target track by track.

## **Restoring an Undefined DSORG Data Set**

The PROCESS(UNDEFINEDSORG) keyword permits logical data set restore of an undefined DSORG data set to an unlike device of larger track capacity. The restore yields a usable data set; however, some unused space might remain on each track of the target data set. It may not always be possible to restore all undefined DSORG data sets to an unlike device type, even when the unlike device type has a track capacity greater than or equal to the source device. For example, if the source device is a 3380, the output device is a 3390, and the data set's block size is less than 277 bytes, a track on the target cannot contain as much data as a track on the source, and message ADR366W (invalid track format) is issued.

**Note:** A data set with an undefined DSORG or with a block size of 0 cannot be restored to a device of smaller track capacity than the source.

## **Restoring a VSAM Sphere**

With DFSMSdss, you can restore an entire VSAM sphere (base cluster and all associated alternate index clusters and paths). The SPHERE keyword causes DFSMSdss to restore the entire VSAM sphere. If the dump was also taken with the SPHERE keyword specified, you need to specify the SPHERE keyword and the base cluster name to restore the base cluster and the other components.

When you restore a sphere to a preallocated target, all components (base clusters, alternate indexes, and paths) of the sphere must be preallocated. DFSMSdss does not restore a sphere if only some parts of the sphere are preallocated.

An example of the RESTORE command with the SPHERE keyword is:

```
RESTORE -  
  INDDNAME(TAPE) -  
  DATASET(INCLUDE(PARTS.VSAM1)) -  
  SPHERE -  
  REPLACE -  
  PSWD(PARTS.VSAM1/MASTUPW1)
```

## Restrictions for Restore Processing

- You can restore a sphere only if all parts of the sphere resolve to the same catalog.
- You may not need to rename all the parts in the VSAM sphere as you would with the copy function.
- Multiple path names to an alternate index are not supported. Only the last path name listed in the catalog is preserved.
- When restoring a sphere with one or more alternate indexes missing from the dump tape, DFSMSDss issues a message to indicate that the sphere was incompletely restored.

## Restoring a Preallocated VSAM Cluster

Restoring a VSAM cluster that has been preallocated is allowed if the following are the same on the source and the destination volumes:

- The number of components on the volume
- The beginning relative byte address (RBA)
- The component names
- Catalog names

The size of the cluster must be equal to or greater than that on the source volume. (Only the tracks that were dumped are restored.)

You should ensure that the control interval size, allocation unit, and secondary allocation quantity are the same as in the initial definition.

## Restoring the VVDS and the VTOCIX

To restore a VVDS or VTOCIX data set, you must specify the fully qualified data set name. The VVDS and the VTOCIX data set cannot be restored with other data sets in the same RESTORE command. VVDS and VTOCIX data sets should not be restored by data set as a normal recovery procedure. The VTOCIX data set is an extension of the VTOC and can be rebuilt using IBM's Device Support Facilities program (ICKDSF).

The VVDS is an extension of the VTOC and of the catalogs for the VSAM data sets on the volume. If it is restored by a data set restore operation, it is possible that some of these data sets can become unusable because of a mismatch between the catalog, the VVDS, and the VTOC. If this occurs, run the diagnose function of access method services to determine the extent of the problem and to take appropriate corrective action.

DFSMSDss/VVDS Manager does not support dumping a multiple-extent VVDS and restoring the VVDS to a nonpreallocated VVDS. DFSMSDss can restore to a nonpreallocated VVDS only when the source VVDS resides on one extent.

The user can consolidate the VVDS extents by doing the following:

- DFSMSDss dump the multiple-extent VVDS
- IDCAMS delete the multiple-extent VVDS
- Preallocate a single-extent VVDS
- DFSMSDss restore the multiple-extent VVDS into the preallocated, single-extent target VVDS.

## Restoring a PDSE

DFSMSDss lets you restore a PDSE.

## Restoring a Damaged PDS

During a logical restore, a PDS is monitored by DFSMSdss for conditions that are not normal. The following conditions are detected and reported:

- Missing high-key entry in the PDS directory
- Missing directory EOF
- Invalid member start TTR:
  - TTR points before directory EOF
  - TTR points after end of data set
- Missing member EOF. Each member of a partitioned data set is normally terminated by an EOF record.
- Invalid note or note list TTR:
  - Note pointing before the start of member data
  - Note pointing after the member EOF
  - Note pointing past the last valid record on a track
  - Note pointing to record 0 of a track

DFSMSdss notes all of these conditions with a message.

During compression, DFSMSdss repairs all missing high-key directory entries, missing directory EOFs, and missing member EOFs.

Invalid start TTRs prevent DFSMSdss from compressing data for that member. DFSMSdss translates all valid note and note list TTRs during compression.

Use the NOPACKING keyword to restore damaged partitioned data sets to same or like device target volumes. This results in an exact track-for-track image of the source data set. Obviously, no compression is performed in this case. During physical restore operations, DFSMSdss uses only track-level I/O. Therefore, no compression takes place against the PDS.

## Restoring Data Sets in an SMS-Managed Environment

Use the RESTORE command to recover data sets in an SMS-managed environment. If the data set was dumped logically, it is recovered logically. If it was dumped physically, it is recovered physically.

As discussed earlier, an SMS-managed environment can contain both SMS-managed and non-SMS-managed data. The following sections discuss how you can use the RESTORE command to recover these data sets.

### Note:

#### Programming Interface information

---

The following sections discuss variables available to automatic class selection (ACS) routines during DFSMSdss processing. This information is provided for guidance purposes only. It is not associated with any interface provided by DFSMSdss. For details on writing ACS routines, refer to *OS/390 DFSMSdfp Storage Administration Reference*.

#### End of Programming Interface information

---

## Converting Non-VSAM Data Sets to Multivolume

### Programming Interface information

---

The number of volumes allocated for certain VSAM and non-VSAM data sets can be changed with VOLCOUNT keyword options. The output data set must be SMS-managed. Single-volume data sets can be converted to multivolume; multivolume data sets can be converted to single-volume; or the number of volumes allocated for multivolume data sets can be changed. Allocation depends on which VOLCOUNT keyword is selected, and on whether output volumes are specified.

**Note:** TTR-BDAM and unmovable data sets cannot be converted to multivolume with the VOLCOUNT keyword. If an existing multivolume TTR-BDAM or unmovable data set is encountered, a DADSM error occurs. Partitioned data sets (PDS and PDSE) cannot be made multivolume by the VOLCOUNT keyword. If an existing multivolume PDS or PDSE data set is encountered, it is converted to single-volume.

### End of Programming Interface information

---

## Restoring SMS-Managed Data Sets

When you use the RESTORE command in an SMS-managed environment, automatic class selection (ACS) routines are invoked. ACS routines are written for each installation by the installation's own storage administrator. Refer to "Appendix. ACS Routine Information" on page 115 for more information.

When you use the RESTORE command, you are in the ACS RECOVER environment. See "ACS Variables Available during RESTORE and CONVERTV Processing" on page 116 for a list of variables available to ACS routines during restore processing.

DFSMSdss passes a data set's classes at the time of the dump to ACS as input, and the ACS routines can assign or override these input classes.

VSAM alternate indexes do not have SMS constructs of their own; they use the same constructs as the base cluster. When restoring alternate indexes as independent clusters (because you did not specify the SPHERE keyword on the DUMP and RESTORE commands), DFSMSdss passes null classes to ACS. If you want DFSMSdss to pass the base cluster's classes to ACS, you must invoke sphere processing by specifying the SPHERE keyword on the DUMP and RESTORE commands.

If the source data set is not SMS-managed and has no class names, DFSMSdss passes null classes to ACS. If the source data set is SMS-managed and you do not specify otherwise, DFSMSdss passes ACS the source data set's classes. If you specify what you want passed to ACS with the STORCLAS, MGMTCLAS, NULLSTORCLAS, or NULLMGMTCLAS keywords, DFSMSdss passes ACS what you specify. In all cases, the ACS routines ultimately decide the classes assigned to the data set.

You can, however, force the storage class and management class you specify to be assigned to a data set by using the BYPASSACS keyword with the RESTORE command.

The following RESTORE command results in ACS routines determining the target classes, using the source classes as input:

```
RESTORE -  
    INDDNAME(TAPE) -  
    DATASET(INCLUDE(USER12.**))
```

If you preallocate a data set and specify the REPLACE keyword, the preallocated data set's classes are used.

For details about using the RESTORE command to convert data to and from SMS management, see "Chapter 8. Converting Data to and from SMS Management" on page 95.

### Changing Storage Class with Restore

In some cases, you might want to pass ACS a storage class that is different from that of the source data set. You can specify the RESTORE command with the STORCLAS keyword to pass ACS a storage class name as follows:

```
RESTORE -  
    INDDNAME(TAPE) -  
    DATASET(INCLUDE(USER12.**)) -  
    STORCLAS(SCNAME1)
```

However, using STORCLAS does not guarantee that the data set is assigned the storage class you specify. It means only that the storage class you specified is passed to the ACS routines. Depending on how your installation's ACS routines are written, the storage class you specify can be ignored, assigned to the data set, or used in combination with other input variables to determine a new storage class for the data set.

RACF checks if the RESOWNER field of a given data set is authorized to define the data set with the given STORCLAS. Ensure that the RESOWNER field of the data set has the proper authority to use the indicated storage class.

To make certain that the storage class you specify is assigned to the data set, you can use the BYPASSACS keyword as follows:

```
RESTORE -  
    INDDNAME(TAPE) -  
    DATASET(INCLUDE(USER12.**)) -  
    STORCLAS(SCNAME1) -  
    BYPASSACS(**)
```

In this case, ACS is not invoked, and therefore the data set is assigned whatever you have specified with STORCLAS or, if you do not use STORCLAS, the source data set's storage class. You should ensure that the storage class you specify is valid, or you will get an error.

To limit the use of BYPASSACS, an installation can set up a RACF class profile.

You can use the NULLSTORCLAS keyword in conjunction with the BYPASSACS keyword to make a data set non-SMS-managed. For example, the following specification of the RESTORE command causes the specified data sets not to be SMS-managed:

```
RESTORE -
  INDDNAME(TAPE) -
  DATASET(INCLUDE(USER12.**)) -
  NULLSTORCLAS -
  BYPASSACS(**)
```

### Changing Management Class with Restore Processing

In addition to influencing a data set's storage class when you restore it, you can also give ACS input for assigning or overriding the data set's management class. By specifying MGMTCLAS, you can pass a management class name to ACS and, as with STORCLAS, ACS can ignore it, assign it to the data set, or use it in combination with other input variables to determine the data set's management class. By specifying NULLMGMTCLAS, you can pass a null management class to ACS, which may or may not assign a management class.

An example of the RESTORE command with the MGMTCLAS keyword is:

```
RESTORE -
  INDDNAME(TAPE) -
  DATASET(INCLUDE(USER12.**)) -
  MGMTCLAS(MCNAME1)
```

As with STORCLAS, RACF checks if the RESOWNER field of a given data set is authorized to define the data set with the given MGMTCLAS. Ensure that the RESOWNER field of the data set has the correct authority to use the indicated management class.

Just as you can with STORCLAS, you can use MGMTCLAS with BYPASSACS to ensure that the data set is assigned the management class you specify. For instance:

```
RESTORE -
  INDDNAME(TAPE) -
  DATASET(INCLUDE(USER12.**)) -
  MGMTCLAS(MCNAME1) -
  BYPASSACS(**)
```

You should ensure that the management class you specify with MGMTCLAS is valid, or you will get an error. Remember that BYPASSACS skips both the STORCLAS and MGMTCLAS ACS routines.

To limit the use of BYPASSACS, an installation can set up a RACF class profile.

When you influence or assign the management class of a data set, you also need to be careful that the data set resides in a storage group capable of providing for the management class attributes associated with the management class you specify. For instance, if a data set has a management class that makes it eligible for migration, it needs to reside in a storage group on which DFSMSHsm does migration. Otherwise, the data set will never migrate. For this reason, you might have to change the storage class along with the management class to ensure that the data set resides on volumes that can accommodate its management class.

However, if you are having to continually override your installation's ACS routines, you should see your storage administrator about changes to the ACS routines that would make it possible to let SMS do its job.

## Restoring SMS-Managed Data Sets Physically

In general, it is recommended that you use logical data set restore processing in an SMS-managed environment. If you use physical data set restore processing, you should be aware of the special rules for volume and SMS construct selection.

When restoring a non-SMS-managed user catalog on an SMS-managed volume or an SMS-managed user catalog on a non-SMS-managed volume, physical restore does **not** convert the catalog. Instead, DFSMSdss physical restore ensures that the user catalog looks exactly like the source catalog (SMS or non-SMS-managed) and then places the output volume in INITIAL status.

DFSMSdss physical data set restore processing is sensitive to the number of logical volumes in a dump data set. A DFSMSdss physical dump tape can contain multiple logical volumes. Because a physical dump operates at the track-image level, every volume from which data was dumped is on the tape in the form of a logical volume.

The following example shows how a dump tape can contain more than one logical volume:

```
DUMP -  
  DATASET(INCLUDE(**)) -  
  INDYNAM((338001),(338002)) -  
  OUTDD(TAPE) -  
  COMPRESS
```

If data sets are dumped from both volumes, two logical volumes are on the dump tape.

During physical data set restore processing, the SMS class selection is similar to logical data set restore processing. The source data set's SMS classes (if any) are used as input to the ACS routines. You can influence the classes selected for the target data set by using the STORCLAS, MGMTCLAS, NULLSTORCLAS, NULLMGMTCLAS, and BYPASSACS keywords.

The major difference in physical data set restore (as opposed to logical data set restore processing) is that all the data will be restored to the first volume you specify in the OUTDDNAME or OUTDYNAM keyword.

**Note:** If the specified target volume is SMS-managed, no non-SMS-managed data sets are restored; conversely, if the specified target volume is not SMS-managed, no SMS-managed data sets are restored.

## Restoring GDG Data Sets

For generation data group (GDG) data sets, filtering on generations is supported. Generation names in relative generation number, dsn(n), can be specified in the INCLUDE and EXCLUDE parameters. The GDG base must be defined (cataloged) before restoring GDG data sets. Otherwise, messages indicating that catalog errors have occurred may be issued during the restore.

**Note:** When using JOBCAT or STEPCAT during the restore, ensure that the GDG base is defined in the catalog in which the GDG data sets that are being restored will be cataloged. If there is no GDG base in the JOBCAT or STEPCAT, but there is a GDG base in another catalog in the search order, the GDG data sets will be restored to the GDG base that exists outside the JOBCAT or STEPCAT. *Failure to take this precaution can result in the GDG*

*data sets being restored to the wrong GDG base, which can then, in turn, result in current GDG data sets in that base being rolled off or possibly even scratched.*

### **Restoring SMS-Managed GDG Data Sets**

SMS-managed GDG data sets can be in any one of the following states:

- ACTIVE
- DEFERRED
- ROLLED-OFF

When restoring a GDG data set to SMS-managed storage, DFSMSdss does one of the following:

- Preallocated restore retains the status of the preallocated generation data set (GDS).
- Restore function places the GDS in DEFERRED status, if the TGTGDS keyword is not specified. DFSMSdss leaves the GDS in DEFERRED status to enable you to (1) roll it back as an ACTIVE generation or (2) leave it as DEFERRED.
- If the TGTGDS keyword is specified, the appropriate status is assigned to the data set as long as the requested target status does not violate rules of the generation data group. The default status of logical and physical data set restore operation is DEFERRED.

### **Restoring Non-SMS-Managed Data Sets**

To restore a data set to a non-SMS-managed target volume, you can use the NULLSTORCLAS and BYPASSACS keywords on the RESTORE command. If you use these keywords, the data set is placed on a non-SMS-managed volume, regardless of whether or not the source data set was SMS-managed.

Extended-physical-sequential data sets, VSAM extended-format data sets, HFS files, SAM compressed extended function, and data sets with DFM attributes cannot be restored to non-SMS-managed target volumes during a physical or logical data set restore.

### **Logical Restore of Data Sets with Phantom Catalog Entries**

During a disaster recovery, the logical restore of data sets may be unsuccessful because of *phantom catalog entries*; that is, the target data set names are cataloged but the target data sets do not exist. This condition can occur if you have:

- Scratched the target volumes and have not deleted the catalog entries for the corresponding data sets
- Restored the catalogs before restoring the data sets
- Restored the target data sets to different volumes from the offline source volumes, and the target data sets have not been renamed

DFSMSdss provides two parameters to support disaster recovery operations: DELETEDCATALOGENTRY and IMPORT.

#### **DELETEDCATALOGENTRY Keyword**

DELETEDCATALOGENTRY tells DFSMSdss to do a DELETE NOSCRATCH operation for any phantom catalog entry for a target data set being restored.

**Attention:** DELETECATALOGENTRY should be used with extreme care. **Do not** use it if:

- Any volumes on the restoring system are varied offline. If you do, DFSMSDss does a DELETE NOSCRATCH for any data set being restored that exists on the varied offline volume. Then, when the volume is varied online, you will have two data sets: a cataloged, restored data set and an uncataloged original data set on the volume that was varied offline.

Also note that if the volumes are varied offline, catalog messages will be issued for each cataloged data set informing you that the volume is offline and requesting that you reply with 'CANCEL' or a device name.

- The restoring system is sharing catalogs with another system but not sharing the data set volumes. If you do, DFSMSDss does a DELETE NOSCRATCH for any data set that is cataloged in the shared catalog on the other system but that is on a volume not available to the restoring system. After the restore, you may have two data sets: a cataloged, restored data set and an uncataloged original data set on a volume in the other system.

### **IMPORT Keyword**

IMPORT specifies that you are restoring data sets that were dumped from a system other than the one into which the restore is being done. Because the data sets to be restored are new to the system, the usual source data set authorization checks are not done. If you are authorized to read the input dump data set containing the data sets being restored, you have the authority to read any data set being restored. DFSMSDss continues to ensure that you are authorized to create a new target data set or replace an existing one.

---

## **Restoring Volumes**

You can recover a volume or ranges of tracks from a full-volume dump operation. If the dump volumes resulted from a full dump operation, you can do a full or a tracks restore (that is, ranges of tracks) or a data set restore operation. If the dump volumes resulted from a tracks dump operation (that is, ranges of tracks), you must do a tracks RESTORE command, which can consist of a subset of the dump data.

An example of a full-volume restore operation is:

```
RESTORE -  
    INDDNAME(TAPE) -  
    OUTDDNAME(DASD1) -  
    PURGE
```

With the restore operation, you can copy the volume serial number to the output DASD with the COPYVOLID keyword. For example:

```
RESTORE -  
    INDDNAME(TAPE) -  
    OUTDDNAME(DASD1) -  
    COPYVOLID -  
    PURGE
```

### **Notes:**

1. COPYVOLID is required if you are restoring an SMS-managed volume, unless the source and target volume serial numbers match.
2. Data set restore of VSAM extended-addressable data sets from a physical volume dump is not supported.

You must consider several factors when restoring volumes in an SMS environment. Before you start to restore a full volume, you must ensure that the status of the target volume is synchronized with its environment. For example, if the target volume is a non-SMS-managed volume, the volume must not be defined in a storage group. Conversely, if the target volume is an SMS-managed volume, the volume must be defined in a storage group. Finally, if the target volume is SMS-managed, then SMS must be active for the full-volume restore operation.

If you are using Record Level Sharing (RLS), be careful when restoring volumes with the FULL or TRACKS keywords. If the target volume has data sets on it that have retained locks or data in the coupling facility associated with them, a full-volume or tracks restore can result in data integrity problems.

## Specifying Output Volumes

For a full or tracks restore, you must specify an output volume by using the OUTDDNAME or OUTDYNAM keywords.

The device type of the source volume used in the dump operation and the device type of the target volume used in the restore operation must be the same. However, the following exceptions are possible:

- Data from a smaller-capacity IBM 3380 model can be restored to a larger-capacity IBM 3380 model.
- Data from a smaller-capacity IBM 3390 model can be restored to a larger-capacity IBM 3390 model.
- Data from a minivolume or a virtual volume can be restored to a real volume of like device type, and vice versa, device capacity permitting.
- Data can be restored from a larger-capacity IBM 3380, 3390, or 9345 model to a smaller-capacity IBM 3380, 3390, or 9345 model, if you are restoring specific track ranges using the TRACKS keyword and if the range of data to be processed falls within the capacity of the output device.
- Data from a smaller-capacity IBM 9345 can be restored to a larger-capacity IBM 9345.

**Note:** When you perform a full-volume restore to a DASD that is shared between multiple systems, the DASD should be offline to all systems except the one performing the restore.

When doing a full-volume restore operation to DASD, DFSMSdss automatically corrects the free-space information on the volume and rebuilds, as necessary, the VTOC index. DFSMSdss does this whenever it copies data to a larger-capacity DASD from a tape dumped from a smaller-capacity DASD or whenever both of the volumes, including volumes of equal capacity, contain a VTOC index. DFSMSdss allocates a large (more than 65 535 tracks) dummy data set to recalculate the free-space information; ignore any IEC614I messages that DFSMSdss generates as part of this procedure.

During a full-volume restore operation, other jobs may be enqueued on the output volume. If so, DFSMSdss cannot enqueue on the output volume to perform the full-volume restore operation. To determine if the output volume is allocated before performing the full-volume restore operation, issue the following operator command:

```
D U,DASD,ALLOC,cuu,1
```

This command displays the specified volume and the names of the jobs enqueued on it.

If, for example, the catalog has enqueued on the output volume, you can take the following steps:

1. Use the following catalog modify command to display a list of all open catalogs:

```
MODIFY CATALOG,LIST
```

2. Use the following catalog modify command to make CAS unallocate the catalog:

```
F CATALOG,UNALLOCATE(catname)
```

If no other allocated catalogs are on the volume and the volume is not allocated by any other users, you can proceed with your full-volume restore.

For more information on CAS allocation, refer to *OS/390 DFSMS: Managing Catalogs*.

### **Processing RACF-Protected Data Sets**

On a physical restore operation, DFSMSdss does not delete the profiles of RACF-protected data sets on the volume before a full restore operation. After a full restore, RACF profiles are not built for RACF-indicated data sets on the restored volume. If RACF data set profiles do not exist for these data sets, these data sets are inaccessible until RACF profiles are built for them.

If you use the COPYVOLID keyword to change the volume serial number or if the volume serial for the dump volume and the restored volume are different, DFSMSdss does not build profiles for the RACF-protected data sets on the restored volume or for the RACF DASDVOL for the RACF-protected DASD volume.

The protection status of data sets that are restored through a full restore is unpredictable if:

- RACF profiles (generic or discrete) of the data sets were changed between dump and restore functions.
- The dump was produced on a system (that supports RACF generic profiles) other than the one used for the restore.

### **Recovering System Volumes**

You can use the Stand-Alone restore program of DFSMSdss to perform either a full or a tracks restore from the first data set of DFSMSdss-produced dump tapes, without the use of a host system environment. Using Stand-Alone restore you can recover system volumes in order to bring up the host environment.

You can also use Stand-Alone restore in a VM environment. Stand-Alone restore operates in ESA/370 mode, System/390 mode, System/370 XA mode, or System/370 mode. Refer to *OS/390 DFSMSdss Storage Administration Reference* for instructions on how to perform the restore.

## **Recovering VM-Format Volumes**

You can use DFSMSdss to recover VM-format volumes that are accessible to your MVS system. The volumes must have OS-compatible VTOCs starting on track zero, record five. DFSMSdss can only retrieve device information from the OS-compatible VTOC, and cannot interpret any VM-specific information on the volume.

Use the CPVOLUME keyword and specify the range of tracks to be restored with the TRACKS keyword. Because DFSMSdss cannot check access authorization for VM data, CPVOLUME is only allowed with the ADMINISTRATOR keyword.

Exercise caution when using DFSMSdss to recover VM-format volumes, because DFSMSdss does not serialize any VM data in any way. If you restore OS-format volumes to VM-format volumes or VM-format volumes to OS-format volumes, you must restore all of the volume's tracks. Failure to restore all of the tracks may render the volume unusable.

---

## Chapter 7. Managing Data Movement with DFSMSdss

Data movement is necessary when you are doing the following tasks:

### Replacing devices

When you remove devices to be replaced with other ones, you must move the data off the devices you are removing.

### Adding devices

If you add new devices at your site, you must move data onto them to take advantage of the added capacity.

### Maintaining devices

When you are servicing a volume, you might need to move data off the volume so users can continue to access the data.

### Tuning performance

If a volume is performing poorly, it might be because data sets on the volume are being frequently accessed and causing an I/O bottleneck. In this case, you might move the data sets to another volume that is better able to handle it (either because it is less full or because it is cached).

You can use the DFSMSdss COPY command to move data between volumes.

**Note:** The logical data set copy function does not support hierarchical file system (HFS) data sets.

---

## Preparing for Data Movement

Before moving your data, determine the amount of space the data requires. You can determine this by building a data set or volume list with ISMF. A data set list indicates how much space is allocated for each data set and how much space it actually uses. A volume list indicates how much free space is on each volume in the list. You can use this information to calculate how much space the data to be moved requires and to ensure that enough free space exists on the target volumes. This calculation is especially important when combining multiple devices onto one larger-capacity device.

**Note:** In an SMS-managed environment, this calculation is unnecessary if enough DASD space is provided because the system finds the necessary free space and places the data for you.

Ensure that enough free space exists to contain the data, and back up the data before moving it to guard against its loss during the movement. You can use the DFSMSdss DUMP command to back up volumes or data sets. See Chapter 6. Managing Availability with DFSMSdss and refer to *OS/390 DFSMSdss Storage Administration Reference* for more information on using the DUMP command to back up data.

## Evaluating the Use of Logical and Physical Copy

As previously stated, you can use the COPY command to perform the actual data movement. However, you must determine whether to use logical or physical copy function to move the data. The physical copy function gives you better performance, but the logical copy function allows you to move data to unlike devices.

Because the physical volume copy operation enqueues on the volume, do it only at a time of low activity. The logical data set copy operation enqueues at the data set level, and, as a result, it is less critical that you perform the copy operation at a time of low activity. However, seriously consider doing so because DFSMSdss cannot move data sets that are in use. In addition, because logical processing takes more time than physical processing, do it at a time of low activity, especially if you are moving a large number of data sets.

**Note:** It is best not to specify TOLERATE(ENQFAILURE) when moving data with the COPY command. Moving data while it is being updated can result in the loss of the updates.

After DFSMSdss has finished processing, you can verify that the data has moved by looking at the ISMF data set or volume list.

## Controlling What DFSMSdss Copies

DFSMSdss copies only used space for sequential or partitioned data sets and data sets with null DSORG fields (X'0000'), unless overridden by ALLDATA or ALLEXCP. Use the ALLDATA(\*) and ALLEXCP parameters to process allocated space if:

- You are not sure of the data set organization (DSORG) of a data set on a volume when doing a full-volume copy operation.
- There are sequential, partitioned, or data sets with null DSORG fields (X'0000') that are not accessed using SAM or PAM.

**Note:** The COPY command requires temporary work space. Ensure that public or storage volumes are available. Some temporary data sets are allocated to nonspecific devices by referring to SYSDA or SYSALLDA generic groups. If DFSMSdss is to function, these allocations must be allowed by the installation. Allocation validation exits must not restrict DFSMSdss allocations.

For temporary data sets allocated to nonspecific devices, DFSMSdss provides no unit type. SYSDA, SYSALLDA, or whatever is specified in the default allocation table is used. In an SMS-managed environment the default unit specified in the SMS base configuration table is taken even for non-SMS-managed temporary data sets.

---

## Moving Data Sets

Using the COPY command with the DATASET keyword, you can copy one or more data sets from one DASD volume to another of like or unlike device types. If you specify the DELETE keyword with the COPY command, the data set on the source volume is deleted after it has been successfully copied to the target volume. In this way, you can perform a data set move.

**Note:** Concurrent copy operation fails and a message is issued if the DELETE keyword is specified.

## Specifying Input Volumes

The COPY DATASET command does not require that you specify input volumes. If you do not specify input volumes, data sets are selected from all the data sets cataloged in the standard search order.

If you specify input volumes (with either the LOGINDDNAME or LOGINDYNAM volume list), a data set is selected based on the following criteria:

- When you either specify SELECTMULTI(ALL) or specify input volumes without specifying the SELECTMULTI keyword, **all** of the volumes that contain a part of the non-VSAM data set or VSAM base cluster must be in the volume list. For VSAM data sets:
  - If SPHERE is not specified, all parts of the base cluster or alternate index must be contained in the volume list.
  - If SPHERE is specified, all parts of the base cluster must be contained in the volume list.
- When you specify SELECTMULTI(ANY), **any part** of the non-VSAM data set or VSAM base cluster can be on a volume in the volume list. For VSAM data sets:
  - If SPHERE is not specified, any part of the base cluster or alternate index can be contained in the volume list.
  - If SPHERE is specified, any part of the base cluster can be contained in the volume list.
- When you specify SELECTMULTI(FIRST), the volume list must include the volume that contains the **first part** of either the non-VSAM data set or the primary data component of the base cluster for a VSAM sphere. For VSAM data sets:
  - If SPHERE is not specified, the volume list must include the volume containing the first extent of the data component for the base cluster or the volume containing the first extent of the data component of an alternate index.
  - If SPHERE is specified, the volume list must include the volume containing the first extent of the data component for the base cluster.

**Note:** DFSMSdss, when processing input volumes, filters first based on the VTOC, and second, based on catalog filters, if specified.

If a data set is found on more than one specified input volume and the volume sequence numbers match, DFSMSdss cannot determine which data set is to be selected for processing. You do not need to specify a SELECTMULTI option when you build a list of input volumes with STORGRP. The volume list will contain all of the volumes in a storage group.

## Selecting Output Volumes

Specifying output volumes is required for the COPY DATASET command in a non-SMS-managed environment. (For a discussion of SMS considerations for moving data, see “Moving SMS-Managed Data Sets” on page 85.)

You can specify multiple target volumes with the OUTDDNAME or OUTDYNAM keywords. This allows you to specify *spill* volumes. These spill volumes are used if the data sets you are moving require more space than is available on your first choice of volume.

If the output volume has unexpired data sets, you can either not process the data sets or write over them.

DFSMSdss now distinguishes between non-SMS and SMS volumes specified in the OUTDDNAME or OUTDYNAM keywords. For non-SMS allocations, only the volumes that are non-SMS are considered for allocation. Similarly, only SMS volumes are considered for SMS allocations.

The distinction between SMS and non-SMS is also used when determining the volume count for a multivolume allocation. Where volume count is determined from the number of specified volumes, only those volumes eligible for the type of allocation (SMS volume for SMS allocation or non-SMS volume for non-SMS allocation), processing proceeds with a null volume list.

There are several reasons for distinguishing between SMS and non-SMS volumes:

- Non-SMS volumes cannot be used for SMS allocations
- Specifying non-SMS volumes interferes with SMS guaranteed-space allocation
- Reducing volume count problems
- Improving the ability of DFSMSdss to process both non-SMS and SMS allocations in a single operation

## Renaming Data Sets

You can rename data sets with the RENAMEU keyword of the COPY command. For VSAM data sets, you can rename only clusters. DFSMSdss derives the new names for the components of VSAM clusters as follows:

- The old cluster name can have the following format:

```
HLQ1.DSNDB%.HLQ3.HLQ4.I0001.A*
```

where % is any single character, and \* is any string of 0 to 7 characters. In this case, DFSMSdss generates the target component name as follows:

- If a qualifier from the source cluster name is identical to the corresponding qualifier of the source component name, the corresponding qualifier from the target cluster name is used in the target component name. Otherwise, the qualifier from the source component name is used in the target component name.
  - The sixth qualifier of the new component name is set to AD for a data component or to AI for an index component whenever:
    - The new target component name exceeds 44 characters.
    - The new cluster name and new component name are identical.
    - The old component name and new component name are identical.
  - If the new target component name is directed by the standard order of search to a different catalog than the new cluster name, then the target component name is regenerated using the first five qualifiers of the new cluster name and appending a sixth qualifier of either AD for the data component or AI for the index component.
- The component names are derived by using the naming conventions of the system that entails the creation of qualifiers based on the system date and clock values. The naming convention is of particular interest in the following situations:
    - When the old cluster name and the new cluster name both have “cluster” as the last qualifier, and the old component names match the cluster name up to the last qualifier, then the new component names will adhere to the old component naming convention.

For example, assume that the old cluster name is SYS1.IODF00.CLUSTER and the old data component name is SYS1.IODF00. When you specify RENAMEU(SYS1,SYS2), DFSMSdss sets the new cluster name to SYS2.IODF00.CLUSTER and the new data component name to SYS2.IODF00.
    - When the old cluster name and the new cluster name both have “cluster” as the last qualifier, and the old component names *do not match* the cluster

name up to the last qualifier, then DFSMSdss will generate new component names using the new cluster name. DFSMSdss replaces the last “cluster” qualifier with “data” or “index”.

For example, assume that the old cluster name is SYS1.IODF00.CLUSTER and the old data component name is SYS1.CHAK05. When you specify RENAMEU(SYS1,SYS2), the new cluster name will be SYS2.IODF00.CLUSTER and the new data component name will be SYS2.IODF00.DATA.

- When the cluster name is less than or equal to 42 characters and the last qualifier is not “cluster”, component names are created by adding a single character, “D” or “I”, to the new cluster name.

For example, say that the old cluster name is SYS1.IODF00.DATASET and the old data component name is SYS1.CDNN02. When you specify RENAMEU(SYS1,SYS2), the new cluster name will be SYS2.IODF00.DATASET and the new component name will be SYS2.IODF00.DATASET.D.

## Expiration Date Handling

When you copy a data set, the expiration date of the target data set is dependent upon whether:

- The data set is VSAM or non-VSAM.
- The source data set is SMS or non-SMS-managed.
- The target data set is SMS or non-SMS-managed.
- The source data set is cataloged or not cataloged.
- The SMS target’s expiration date matches the target’s management class.

### SMS to SMS

The catalog expiration date and the expiration date in the Volume Table of Contents (VTOC) will have the same value as that of the source data set. For an indexed VSAM data set, the expiration date in the VTOC for the index component will be zero. If the expiration date is different than the target’s management class, SMS will modify the expiration date to match the target’s management class.

### SMS to Non-SMS

The expiration date handling is dependent upon whether the data set is VSAM or non-VSAM:

- VSAM data set: The catalog expiration date will be the same as that of the source data set. In the VTOC, the expiration date is set to 99365. For an indexed VSAM data set, the expiration date in the VTOC for the index component will also be 99365.
- Non-VSAM data set: The catalog expiration date and the expiration date in the VTOC will have the same value as that of the source data set.

### Non-SMS to SMS

The expiration date handling is dependent upon whether the data set is VSAM or non-VSAM and whether the source data set is cataloged or not cataloged:

- VSAM data set: The catalog expiration date and the expiration date in the VTOC will have the same value as the catalog expiration date of the source data set. For an indexed VSAM data set, the expiration date in the VTOC for the index component will be zero. If the expiration date violates the target’s management class, SMS will change the date to conform with the management class.
- Non-VSAM data set: The catalog expiration date and the expiration date in the VTOC will have the same value as the VTOC expiration date of the source data

set. If the expiration date violates the target's management class, SMS will change the date to conform with the management class.

### Non-SMS to Non-SMS

The expiration date handling is dependent upon whether the data set is VSAM or non-VSAM and whether the source data set is cataloged or not cataloged:

- VSAM data set: The catalog expiration date is the same as that of the source data set. In the VTOC, the expiration date is set to 99365. For an indexed VSAM data set, the expiration date in the VTOC for the index component will also be 99365.
- Non-VSAM data set: The catalog expiration date of the source data set is used for the catalog expiration date of the target data set. The expiration date in the VTOC of the source data set is used for the VTOC expiration date of the target data set.

## Defining RACF Profiles

For information on defining RACF profiles, refer to *OS/390 DFSMSdss Storage Administration Reference*.

## Moving Data Sets with Utilities

In some cases, DFSMSdss invokes a utility to move a data set. Table 7 shows when DFSMSdss invokes a utility for a data set copy operation.

When you move a data set and a utility is used, the data set must be cataloged in the standard search order.

When DFSMSdss invokes IEBCOPY to copy a LOADMOD, message IEC507D is issued requesting operator authorization to overwrite an unexpired area when the source data set has an incorrect RLD count and an unexpired date.

When DFSMSdss invokes IEHMOVE to copy data sets, IEHMOVE has DD statement requirements that DFSMSdss cannot always satisfy. To avoid potential abnormal ends, do one or both of the following:

- Specify the source and target volumes as PRIVATE.
- Ensure that the source and target volumes are not in the list of default volumes for dynamic allocation.

When DFSMSdss invokes IDCAMS to copy a KSDS, the data set is automatically reorganized to optimize it for VSAM processing. A large KSDS may require extensive reorganization that could result in greater processing time for the copy operation. If IDCAMS was selected because multiple output volumes were specified, performance may be improved by specifying a single output volume for the data set.

Table 7. Data Mover Selection Matrix for Data Set Copy

Data Set Type	Like Devices	Unlike Devices
Sequential	DFSMSdss	DFSMSdss
Partitioned (not PDSE)	DFSMSdss (1, 2)	DFSMSdss (1, 2)
Partitioned (not PDSE) load modules	DFSMSdss (3)	IEBCOPY
Partitioned data set extended (PDSE)	DFSMSdss (4)	DFSMSdss (4)
Direct nonrelative block address mode	DFSMSdss	DFSMSdss (5)
Direct relative block address mode (6)	DFSMSdss	DFSMSdss

Table 7. Data Mover Selection Matrix for Data Set Copy (continued)

Data Set Type	Like Devices	Unlike Devices
Indexed sequential to same track. Target volume uses VTOC index. Target space available.	DFSMSDss	N/A
Indexed sequential (all other cases)	IEBISAM	IEBISAM
ESDS	DFSMSDss (7)	DFSMSDss (7, 8)
RRDS	DFSMSDss (7)	IDCAMS (REPRO)
LDS	DFSMSDss (7)	IDCAMS (REPRO)
KSDS or VRRDS	DFSMSDss (9)	IDCAMS (REPRO)
Key range data set	DFSMSDss (10)	IDCAMS (REPRO)
Extended-format VSAM	DFSMSDss (7)	IDCAMS (REPRO)
Integrated catalog facility user catalogs	IDCAMS (EXPORT/IMPORT)	IDCAMS (EXPORT/IMPORT)
Undefined DSORG	DFSMSDss	DFSMSDss
<p><b>Notes:</b></p> <ol style="list-style-type: none"> <li>All partitioned data sets that are not load modules are compressed during a copy to a like or unlike device.</li> <li>DFSMSDss calls IGWFAMS when you are converting a PDS to a PDSE.</li> <li>If copying partitioned load modules with REBLOCK, DFSMSDss calls IEBCOPY to copy the data set to a like device.</li> <li>DFSMSDss calls IGWFAMS when the requirements for either snapshot or concurrent copy are not met. DFSMSDss calls IGWFAMS when you are converting a PDSE to a PDS.</li> <li>The source data set is not copied if the target data set is preallocated or if the target device has a smaller track capacity than the source.</li> <li>Specify the DFSMSDss RELBLOCKADDRESS parameter.</li> <li>DFSMSDss calls IDCAMS if the target CISIZE, CASIZE, physical record size, or physical block size of the target is different from that of the source.</li> <li>DFSMSDss calls IDCAMS if the calculated number of blocks per control area is different from the calculated number of usable blocks per control area.</li> <li>DFSMSDss calls IDCAMS if any of the following is true: <ul style="list-style-type: none"> <li>The CISIZE, CASIZE, physical record size, physical block size, imbed, or span attributes of the target are different from that of the source.</li> <li>The target data set is SMS and has an imbedded index or has key ranges, and the target volume count is greater than one. Refer to the VOLCOUNT keyword in <i>OS/390 DFSMSDss Storage Administration Reference</i> to determine the volume count.</li> <li>The target data set is non-SMS, the source component or components span multiple volumes, and there is not enough space on one target volume to contain the entire data set.</li> </ul> </li> <li>DFSMSDss calls IDCAMS if the source and target CASIZE, physical record size, or physical block size are different; if the components span multiple volumes; for a KSDS with IMBED and either the source HURBA=HARBA or it has extended indexes.</li> </ol>		

## Moving Data Sets with Concurrent Copy

### Programming Interface information

The DFSMSDss concurrent copy function lets you move data but minimizes the time that the data is unavailable. The user determines an appropriate time to move the data (for example, when the data is in a known state and update activity is

stopped). DFSMSDss is invoked directly or via the DFSMSDss application program interface (API) to do a concurrent copy of the data. After initialization is complete, DFSMSDss releases any serialization it held on the data sets and prints a message both to SYSPRINT and the console that the concurrent copy operation is logically complete. If DFSMSDss was invoked via the API, DFSMSDss informs the caller through a new UIM exit option, Eioption 24 (refer to *OS/390 DFSMSDss Storage Administration Reference* for details). The application can resume normal operation at this time.

## End of Programming Interface information

---

If for any reason data cannot be processed with concurrent copy (for example, the hardware being used does not support concurrent copy), DFSMSDss uses methods of data movement and does not release the serialization until the copy is completed.

If the source device supports snapshot, but does not support concurrent copy, DFSMSDss uses the snapshot function to provide a concurrent copy-like function known as virtual concurrent copy. For information on virtual concurrent copy, refer to “Concurrent Copy” on page 4.

### Notes:

1. If DFSMSDss invokes a utility such as IDCAMS REPRO or IEBCOPY for a data set copy operation, concurrent copy is not done for those data sets.
2. The CONCURRENT keyword applies to all the data being dumped or copied by the function under which it is specified. It cannot be applied to a subset of the data being processed.
3. To improve data integrity, do not update the data during a concurrent copy initialization.
4. If a concurrent copy operation fails after signaling that the concurrent copy initialization is complete (and update activity on the data has resumed), it is not possible to recover the data at the point-in-time at which the concurrent copy operation has started. This is because the data may have been updated while the copy operation was progressing.
5. The CONCURRENT keyword cannot be used with the DELETE or UNCATALOG keywords.
6. VM minivolumes are supported if you are using RVA devices to the extent that they are supported by IBM Extended Facilities Product (IXFP) device reporting.

## Moving Data Sets with Snapshot

When the source and target devices are in the RAMAC Virtual Array (RVA) and the data does not need to be manipulated (such as, reblocked, track packed to unlike), DFSMSDss may be able to use snapshot to quickly move the data from the source location to the target location. This process is referred to as native snapshot support, and it is much faster than traditional methods, especially when large amounts of data are moved.

In order to use native snapshot, the following requirements must be met:

- The source and target device types must be the same.
- The source and target devices must be in the same RAMAC Virtual Array (RVA).
- There must *not* be any required data manipulation. The following types of processing require data manipulation:

- Reblocking — Reblocking occurs when the REBLOCK keyword is specified or when the VTOC indicates that the data set is capable of being reblocked.
  - PDS compression — DFSMSdss, by default, compresses a PDS data set during copy. You can specify the NOPACKING keyword to prevent DFSMSdss from compressing the PDS, thereby allowing the use of snapshot.
  - Changing stripe counts — For a striped VSAM or a striped extended-sequential data set, the source stripe count must be the same as the target stripe count.
  - An individual stripe that extends to more than one volume — A single-striped extended-sequential data set is ineligible if either the source data set or the target data set is multivolume.
  - PDS or PDSE conversion — You cannot specify the CONVERTV keyword with partitioned data sets.
  - Block-by-block processing of direct access data sets — Block-by-block processing occurs when you specify the RELBLOCKADDRESS or the AUTORELBLOCKADDRESS keyword.
- A utility must not be required to move the data.

If the source data is in an RVA, DFSMSdss will attempt to allocate the target data set on the same device type in the same RVA, thus increasing the probability that native snapshot can copy the data. If the source data set is multivolume and not contained entirely in one partition of one RVA subsystem, it is not possible to allocate the target so that native snapshot can be used. These data sets will be allocated to whatever volumes are available, irrespective of their snapshot capability.

DFSMSdss automatically performs native snapshot if possible (even if concurrent copy keyword is specified).

---

## Moving Data Sets with Special Requirements

Some data sets require special treatment when they are moved. The following sections discuss some considerations for moving these special data sets.

### Moving Undefined DSORG and Empty non-VSAM Data Sets

To copy a data set with an undefined DSORG, ensure that the following conditions are met:

- The PROCESS(UNDEFINEDSORG) keyword is specified.
- The selected target volume is either of the same device type as the source volume, or a device type with equal or greater track capacity.

To copy an empty non-VSAM data set, ensure that the following conditions are met:

- An EOF record exists in the first track of the source data set.
- If the target data set is to be SMS-managed, the selected target SMS volume must either be of the same device type as the source data set, or a device type with equal or greater track capacity.

**Note:** It may not be possible to move all undefined DSORG data sets to an unlike device type, even when the unlike device type has a track capacity greater than or equal to the source device. For example, if the source device is a 3380, the output device is a 3390, and the data set's block size is less than 277 bytes, a track on the target cannot contain as much data as a track on the source, and message ADR366W (invalid track format) is issued.

## Moving System Data Sets

Some system data sets do not require movement, either because they are allocated during system generation or because they are built at IPL time. Other system data sets, however, can be moved by DFSMSdss for various reasons.

Unless excluded, system data sets are copied. However, they generally remain open while the system is running and cannot be scratched or uncataloged because the DELETE and UNCATALOG options apply only to data sets not in use.

Frequently, system data sets are prefixed with a high-level qualifier of SYS1. The PROCESS(SYS1) keyword can be used for a data set copy operation of a SYS1 data set to move it to a preallocated target or to copy it with the DELETE option. PROCESS(SYS1) does not apply to VTOCIX or VVDS.

To limit the use of PROCESS, you need to set up a RACF facility class profile. Refer to *OS/390 SecureWay Security Server RACF Security Administrator's Guide* for more information on RACF facility class profiles.

**Note:** The PROCESS(SYS1) option does not lift the restrictions on the processing of volume VVDSs or VTOC indexes.

When the PROCESS(SYS1) keyword is not specified, you cannot move system data sets the way you normally move data sets with DFSMSdss. In order for DFSMSdss to move system data sets, you must do one of the following:

- Dump the data sets, and then restore them to a different volume.
- Copy the data sets to a different volume and then catalog them in a different catalog.

**Note:** When a data set copy operation is used to copy the following data sets, space is defined for the target data set but no data is copied:

- Model DSCBs
- Page and swap data sets
- SYS1.STGINDEX.

## Moving Catalogs

When you copy an integrated catalog facility user catalog, the DELETE keyword must be specified, but an input volume and the RENAMEU keyword must *not* be specified. You must specify the fully qualified name of the user catalog in the INCLUDE parameter. In any processor in the complex, there should be no other jobs executing that access the user catalog being moved; otherwise, the copy operation might fail or the copied catalog might contain errors.

You need RACF access if the catalog is RACF-protected.

User catalog aliases are automatically redefined after the copy. The LOCK attribute of an integrated catalog facility user catalog is preserved during the copy operation. Refer to *OS/390 DFSMS: Managing Catalogs* for a description of the LOCK attribute and the correct access authority.

**Note:** DFSMSdss cannot be used to move an active VSAM master catalog, integrated catalog facility tape volume catalogs (VOLCATALOG), the VVDS, or the VTOCIX.

## Moving Non-VSAM Data Sets That Have Aliases

DFSMSdss does not support INCLUDE filtering of non-VSAM data sets using an alias. To include a non-VSAM data set which has an alias for copy processing, you must use the data set's real name, as shown in the VTOC. In most cases DFSMSdss does not detect or preserve aliases of non-VSAM data sets. However, during logical data set copy with the DELETE keyword specified and the RENAMEU keyword not specified, if the data set is SMS-managed and remains SMS-managed during the copy, any aliases associated with the data set are preserved. In all other cases, you must redefine the aliases after the data set is moved.

## Moving Multivolume Data Sets

If you are specifying input volumes with the LOGINDDNAME or LOGINDYNAM keywords and you are moving multivolume data sets, use the SELECTMULTI keyword on the COPY command. SELECTMULTI allows you to move multivolume data sets in their entirety, even if you do not specify all the volumes on which the data set resides.

If you specify input volumes or candidate space input volumes (with either the LOGINDDNAME or LOGINDYNAM volume list), a data set is selected based on the following criteria:

- When you either specify SELECTMULTI(ALL) or specify input volumes without specifying the SELECTMULTI keyword, **all** of the volumes that contain a part of the non-VSAM data set or VSAM cluster must be in the volume list. For VSAM data sets:
  - If SPHERE is not specified, all parts of the base cluster or alternate index must be contained in the volume list.
  - If SPHERE is specified, all parts of the base cluster must be contained in the volume list.
- When you specify SELECTMULTI(ANY), **any part** of the non-VSAM data set or VSAM base cluster can be on a volume in the volume list. For VSAM data sets:
  - If SPHERE is not specified, any part of the base cluster or alternate index can be contained in the volume list.
  - If SPHERE is specified, any part of the base cluster can be contained in the volume list.
- When you specify SELECTMULTI(FIRST), the volume list must include the volume that contains the **first part** of either the data set or the primary data component of the base cluster for a VSAM sphere. For VSAM data sets:
  - If SPHERE is not specified, the volume list must include the volume containing the first extent of the data component for the base cluster or the volume containing the first extent of the data component of an alternate index.
  - If SPHERE is specified, the volume list must include the volume containing the first extent of the data component for the base cluster.

A multivolume data set can be copied to a single volume or to multiple volumes. For a multivolume data set with a standard user label, only the standard user label on the first volume is copied to the target volumes.

If you do not specify any input volumes, you can move multivolume data sets without any special keywords.

A DFSMSdss logical data set copy operation attempts to ensure that all parts of a multivolume non-VSAM data set exist. In cases where a part of the data set is

missing, such as an inadvertent scratching of the VTOC entry on a volume, DFSMSDss issues an error message and discontinues processing the data set.

DFSMSDss cannot process the following non-VSAM data sets because they are missing one or more parts:

- Multivolume data sets whose catalog volume order differs from the VTOC volume order
- Single-volume data sets with the same name that are cataloged as one multivolume data set
- Multivolume data sets whose last volume indicator in the VTOC is not set

**Notes:**

When copying or restoring multivolume data sets, be aware of the following:

1. DFSMSDss does not preserve candidate volumes. However, for SMS-managed data sets, if you copy and do not specify any output volumes, DFSMSDss preserves the source volume count. If you copy and do specify the output volumes, DFSMSDss sets the volume count to the number of output volumes specified.
2. DFSMSDss does not ensure that the copied or restored data set is on the same number of volumes as the original data set, nor does DFSMSDss ensure that the copied or restored data set extents are the same as the original data set. Instead, DFSMSDss tries to allocate the new data set on as few volumes as possible. This may result in the copied or restored data set becoming a single-volume data set.
3. In addition, DFSMSDss tries to allocate each volume so that all data is contained in a single primary allocation of contiguous space with few, if any, of the secondary allocations being used.

## Converting VSAM and Non-VSAM Data Sets to Multivolume

The number of volumes allocated for certain VSAM and non-VSAM data sets can be changed with VOLCOUNT keyword options. The output data set must be SMS-managed. Single-volume data sets can be converted to multivolume, multivolume data sets can be converted to single-volume, or the number of volumes allocated for multivolume data sets can be changed. Allocation depends on which VOLCOUNT keyword is selected, and on whether output volumes are specified.

**Note:** TTR-BDAM and unmovable data sets cannot be converted to multivolume with the VOLCOUNT keyword. If an existing multivolume TTR-BDAM or unmovable data set is encountered, a DADSM error occurs. Partitioned data sets (PDS and PDSE) cannot be made multivolume with the VOLCOUNT keyword. If DFSMSDss encounters an existing multivolume PDS or PDSE data set, it converts the data set to single-volume.

## Moving VSAM Data Sets

When you move a VSAM data set and the REPLACE keyword is not specified, you must specify DELETE, RENAMEU, or RECATALOG (to a catalog different from the source catalog). If the REPLACE keyword is specified and a preallocated target is not found, DELETE, RENAMEU, or RECATALOG must be specified for the data set to be processed.

For VSAM data sets cataloged in an integrated catalog facility catalog that will be copied using the IDCAMS utility, a preallocated target data set will be renamed

using a DFSMSdss-generated temporary name. This allows dynamic allocation and IDCAMS REPRO to work, because both are currently undirected in catalog usage.

VSAM data sets cataloged in an integrated catalog facility catalog with alternate index and path associations do not use a preallocated target if the DELETE keyword is specified. No search is made for existing data sets in this case. An integrated catalog facility alternate index cannot use a preallocated target. No search is made for existing data sets when copying an alternate index.

For VSAM components that are larger than one cylinder, DFSMSdss will recognize only an integral number of cylinders of free space on a target volume. Also, the required space for a VSAM data set must be contiguous.

You can move the base cluster, all associated alternate index clusters, and paths by using the SPHERE keyword with the COPY command.

### **Restrictions for the COPY command**

The following information covers restrictions when using the COPY command:

- DFSMSdss must be able to invoke IDCAMS to copy an extended-format VSAM data set.
- When performing a logical copy operation of an extended-format data set, the target data set allocation must be consistent with the source data set allocation as follows:
  - If the source is extended-format VSAM, then the target must be extended-format VSAM.
  - If the source is extended-addressable VSAM, then the target must be extended-addressable VSAM.
  - If the source is a compressed VSAM KSDS, then the target must be a compressed VSAM KSDS.
  - If the source is an alternate index for an extended-format KSDS, then the target must be an alternate index for an extended-format KSDS.
  - If the source is an alternate index for a compressed extended-format KSDS, then the target must be an alternate index for a compressed extended-format KSDS.
  - The target control interval size must be equal to the source.
- You can copy a sphere only if all the parts of the sphere resolve to the same catalog.
- Multiple path names to an alternate index are not supported. Only the last path name listed in the catalog is preserved.
- To copy a sphere logically without the DELETE or RECAT keywords, you must rename every data set in the sphere. This includes all paths, all alternate indexes, and the base cluster. If the target sphere is to be SMS-managed, the data sets must be renamed even if the RECATALOG keyword is specified because the RECATALOG keyword is ignored for SMS-managed data sets.

If you do not use the SPHERE keyword and the base cluster has associated alternate index clusters, only the base cluster is moved as follows:

- If you specify DELETE, only the base cluster is moved, but the alternate index cluster continues to be related to the base cluster.
- If you do not specify DELETE, a second copy of the base cluster is created, and the alternate index cluster continues to be related to the original base cluster.

To move an alternate index cluster, specify DELETE on the COPY command. Only the alternate index cluster is moved, and it continues to relate to its base cluster. An alternate index cannot be moved by itself outside the environment of the base cluster. If the base cluster is not SMS-managed, the alternate index cannot be moved to an SMS-managed volume. If the base cluster is SMS-managed, the alternate index cannot be moved to a volume residing in another storage group.

For an empty VSAM data set (zero data relative block address or zero record count), the data set is defined on the target volume but is not copied. Message ADR474W is issued for the data set.

**Note:** DFSMSdss does not preserve candidate volumes during copy processing.

## Moving a PDSE

The COPY command can be used to move a PDSE. The CONVERT keyword, along with the PDSE and PDS subkeywords, can be used with the COPY command to convert a PDS to a PDSE and vice versa.

## Moving a Damaged PDS

DFSMSdss monitors partitioned data sets during compression for conditions that are not normal. The following conditions are detected and reported:

- Missing high key entry in the PDS directory
- Missing directory EOF
- Invalid member start TTR
  - TTR points before directory EOF
  - TTR points after end of data set
- Missing member EOF (each member of a partitioned data set is normally ended by an EOF record)
- Invalid note or note list TTR
  - Note pointing before the start of member data
  - Note pointing after the member EOF
  - Note pointing past the last valid record on a track
  - Note pointing to record 0 of a track

DFSMSdss notes all these conditions with a message.

During compression, DFSMSdss repairs:

- Missing high key directory entry
- Missing directory EOF
- Missing member EOFs

Invalid start TTRs prevent DFSMSdss from compressing data for that member. DFSMSdss translates all valid note and note list TTRs during compression.

You can move damaged partitioned data sets to same or like device target volumes by using the NOPACKING keyword. This results in an exact track-for-track image of the source data set. Obviously, no compression is performed in this case.

## Moving Unmovable Data Sets

When copying unmovable data sets to like devices, DFSMSdss places them at the same track locations on the target volume under the following conditions:

- The target volume has an indexed VTOC.

- The space where the unmovable data would be placed is available.

If any of these conditions do not exist, you must specify the FORCE keyword to move the data set. FORCE enables DFSMSDss to treat the unmovable data set as movable and to move it to an unlike device. Because DFSMSDss places the data set in any available location when FORCE is specified, use FORCE with caution.

If some data sets have CCHHR (cylinder, cylinder, head, head, record) location-dependent data and you are using FORCE, exclude these data sets with the EXCLUDE keyword to prevent DFSMSDss from moving location-dependent data sets.

Another way to position data sets in a specific location on a volume is to allocate all space on the target volume except where you plan to place the unmovable data sets. Then move the unmovable data sets with FORCE and afterwards scratch the dummy space allocation.

## Moving Data Sets to Unlike Devices

DFSMSDss sets the secondary space to zero when processing data sets defined with the contiguous space attribute and zero secondary allocation. This action, which prevents DFSMSDss from creating an unusable data set, may result in ABEND D37-04 due to underallocation of the data set. Should this occur, the user must preallocate the target with adequate space to allow successful copy processing.

## Moving Indexed Sequential Data Sets

When an indexed sequential data set is being moved, the target data set has the prime, index, and overflow areas combined into one allocated area, except when the target data set can be allocated to the same location as the source data set. You must move an indexed sequential data set to a volume with an indexed VTOC. To guarantee a successful allocation, ensure that there is no other activity on this volume.

Indexed sequential data sets that have been allocated with a primary of zero and are empty (there are no extents) are not valid and will not be processed. Also, the VTOC entry of an indexed sequential data set must contain a valid pointer to the other index information.

For a single-volume indexed sequential data set to be selected, either the last-volume indicator must be on or the data set must be cataloged.

## Moving Direct Access Data Sets

When DFSMSDss restores direct data sets, several processing options can be used. Direct data sets can be organized by relative block address or by track-track record (TTR).

Relative block addressable direct access data sets can be processed block by block to like and unlike target devices if the block size fits on the target track. When the data sets are processed block by block, DFSMSDss updates the block reference count of dummy records contained in the relative block addressed direct access data sets. To process block by block, the direct access data sets must have neither a variable record format nor a standard user label.

TTR direct access data sets may become unusable if they are processed block by block. TTR and relative block addressable data sets can be processed track by track to like and unlike target devices whose track capacity is equal to or greater than the source. Block by block processing is more efficient because track by track processing to an unlike device of larger track capacity can leave some unused space on each track of the target data set.

Several DFSMSdss keywords implement the processing options (refer to *OS/390 DFSMSdss Storage Administration Reference* for details on their use):

#### **AUTORELBLOCKADDRESS**

If the data set is accessed with OPTCD indicating relative block addressing, it is processed as if it were specified in the RELBLOCKADDRESS subkeyword list, and processing is block by block. For additional information, refer to *OS/390 DFSMS Macro Instructions for Data Sets* for macro instructions on non-VSAM data sets. If your installation has many relative block address direct access data sets, you may wish to consider the DFSMSdss installation options exit to turn on AUTORELBLOCKADDRESS (refer to *OS/390 DFSMSdss Storage Administration Reference*).

#### **RELBLOCKADDRESS**

If the data set is specified in the subkeyword list, the data set is processed block by block.

#### **TTRADDRESS**

If the data set is specified in the subkeyword list, the data set is processed track by track.

#### **FORCE**

If the track capacity of the receiving volume is smaller than the source, FORCE may be required for variable or undefined length TTR-organized direct access data sets. These data sets may be unusable after restore and, if possible, should be restored to a like device. Use RELBLOCKADDRESS to restore relative block address direct access data sets to unlike devices.

**Note:** If you do not specify a keyword, data is moved to the target track by track.

## **Moving GDG Data Sets**

For generation data group (GDG) data sets, filtering on generations is supported. You can specify generation names in relative generation number, dsn(n), with the INCLUDE and EXCLUDE keywords. During a copy operation, if you catalog the GDGs in a different catalog or you rename them, you must predefine the target GDG base name because the source GDG base name is unusable.

### **Moving Generation Data Sets to SMS-Managed Volumes**

An SMS-managed generation data set (GDS) can be in one of three states:

- ACTIVE
- DEFERRED
- ROLLED-OFF

When copying a GDS to an SMS-managed volume and the data set is not preallocated, DFSMSdss allocates the target GDS as follows:

- If DELETE is specified and RENAMEU is not specified, the target GDS is allocated with the same state as the source GDS.

- If the TGTGDS keyword is specified, the appropriate status is assigned to the data set. The requested target status must not violate rules of the generation data group.
- When the source is an SMS-managed GDS and the target has the same name (that is, DELETE without RENAME), the target status is the same as the source status.
- When the source is a non-SMS-managed GDS and the target has the same name (that is, DELETE without RENAME), the default target status is ACTIVE when the source is cataloged. When the source is not cataloged, the default target status is DEFERRED.
- In all other cases, the default target status is DEFERRED.
- You can use the TGTGDS keyword to alter the target status except when the source is an SMS-managed GDS and the target has the same name.

Table 8 describes the default situation for DFSMSdss to allocate the SMS-managed GDG data set (MOVE refers to COPY command with the DELETE keyword specified):

*Table 8. Default Situation for DFSMSdss to Allocate the SMS-Managed GDG Data Set*

Target Environment	Source Environment	Source Status	DFSMSdss Function	TGTGDS Default
SMS	Non-SMS	Cataloged	COPY	DEFERRED
			MOVE	ACTIVE
		Not Cataloged	COPY	DEFERRED
			MOVE	DEFERRED
	SMS	ACTIVE	COPY	DEFERRED
			MOVE	ACTIVE
		DEFERRED	COPY	DEFERRED
			MOVE	DEFERRED
		ROLLED-OFF	COPY	DEFERRED
			MOVE	ROLLED-OFF

If the data set is preallocated, the state of the target GDS is not altered.

### **Moving Generation Data Sets to Non-SMS-Managed Volumes**

A non-SMS-managed generation data set (GDS) can be in one of two states:

- Cataloged
- Not cataloged

When you copy a GDS to a non-SMS-managed volume, the state of the GDS is determined only by the CATALOG or RECATALOG keywords.

## **Moving SMS-Managed Data Sets**

### **Programming Interface information**

As with the RESTORE command, COPY invokes the Automatic Class Selection (ACS) routines, which in turn assign or override a data set's classes.

When you use the COPY command, you are in the ACS ALLOC environment. The storage class ACS routine is executed first. If the storage class assigned is not null,

the management class ACS routine and then the storage group ACS routine are executed. (See “ACS Variables Available during Copy Function” on page 115 for a list of variables available to ACS routines during copy processing.)

If you do not specify otherwise, DFSMSdss passes the source data set's class names as input to ACS. If you want to specify storage and management class names to be passed to ACS, you can use the STORCLAS and MGMTCLAS keywords. You can use the NULLSTORCLAS and NULLMGMTCLAS keywords to pass null storage and management classes to the ACS routines.

VSAM alternate indexes do not have SMS constructs of their own; they use the same constructs as the base cluster. When copying or moving alternate indexes as independent clusters (because you did not specify the SPHERE keyword on the COPY command), DFSMSdss passes null classes to ACS. If you want DFSMSdss to pass the base cluster's classes to ACS, you must invoke sphere processing by specifying the SPHERE keyword on the COPY command.

If you do not want a data set to be SMS-managed, specify the BYPASSACS and NULLSTORCLAS keywords.

All of these keywords work the same for the COPY command as they do for the RESTORE command (see “Changing Storage Class with Restore” on page 61 and “Changing Management Class with Restore Processing” on page 62).

└─ **End of Programming Interface information** \_\_\_\_\_

## Selecting Target Volumes

└─ **Programming Interface information** \_\_\_\_\_

In an SMS-managed environment, you generally allow the system to place data sets for you. If for some reason you want to control the placement of the data sets (for example, because of performance problems or because you want to put data sets on some new, empty volumes you have just added to a storage group), you must take special steps.

If you use OUTDDNAME or OUTDYNAM to specify a volume list, the volume serial numbers are passed as input to the ACS routines. Depending on how your ACS routines are written, this input might or might not be used in determining where to place the data set.

One way to guarantee that data sets go to particular volumes is to write your storage group ACS routine such that data sets are moved to the volumes you select.

Alternatively, if a data set's storage class has the guaranteed-space attribute, the data set is placed on the user-specified volumes if the volumes reside in the same storage group and ACS selects that storage group for the data set. By using BYPASSACS and STORCLAS keywords, you can ensure that the storage group selected contains the volumes you specify with OUTDDNAME or OUTDYNAM. However, for this procedure to work, your storage group ACS routine must use storage class to determine the storage group for a data set. This allows you to determine which storage class to specify with the STORCLAS keyword to ensure that the storage group containing the volumes specified with OUTDDNAME or

OUTDYNAM is selected.

└─ End of Programming Interface information \_\_\_\_\_

## Changing Storage Class with Copy

└─ Programming Interface information \_\_\_\_\_

You can use the `STORCLAS` keyword to specify a storage class name for `DFSMSDSS` to pass to ACS. You can specify the `NULLSTORCLAS` keyword if you want `DFSMSDSS` to pass a null storage class to ACS.

**Note:** RACF checks if the `RESOWNER` of a given data set is authorized to define the data set with the specified `STORCLAS`. Ensure that the `RESOWNER` of the data set has the correct authority to use the indicated storage class.

Using `STORCLAS` does not guarantee that the data set is assigned the storage class you specify. To ensure that the storage class you specify is assigned to the data set, you must specify `BYPASSACS`. In this case, using `BYPASSACS` causes the storage class and management class ACS routines to be bypassed, so the data set is assigned whatever you have specified with `STORCLAS` or, if you do not use `STORCLAS`, whatever the source data set's storage class is. Ensure that the storage class you specify with `STORCLAS` is valid, or you will get an error.

You can also use `STORCLAS` and `BYPASSACS` to move data sets into a newly defined storage class. For example, suppose you want to combine all your storage classes except two into one new, large storage class. You can code the following:

```
COPY -  
  DATASET(INCLUDE(**) -  
          BY(STORCLAS,NE,(SCNAME1,SCNAME2))) -  
  STORCLAS(SCNAME3) -  
  BYPASSACS(**) -  
  DELETE
```

If you specify `NULLSTORCLAS` and `BYPASSACS` together, the target data set becomes non-SMS-managed.

└─ End of Programming Interface information \_\_\_\_\_

## Changing Management Class with Copy

└─ Programming Interface information \_\_\_\_\_

In addition to influencing a data set's storage class with the copy command, you can also give ACS input for assigning or overriding the data set's management class. By specifying `MGMTCLAS`, you can pass a management class name to ACS and, as with `STORCLAS`, ACS ignores it, assigns it to the data set, or uses it in combination with other things to determine the data set's management class. By specifying `NULLMGMTCLAS`, you can pass null management class to ACS, which might or might not assign a management class to the data set.

**Note:** RACF checks if the `RESOWNER` of a given data set is authorized to define the data set with the specified `MGMTCLAS`. Ensure that the `RESOWNER` of the data set has the correct authority to use the indicated management class.

Also, just as with STORCLAS, you can use MGMTCLAS with BYPASSACS to ensure that the data set is assigned the management class you specify. Ensure that the management class you specify with MGMTCLAS is valid, or you will get an error. You must be authorized to use BYPASSACS and the management class you specify with MGMTCLAS.

└ End of Programming Interface information \_\_\_\_\_

## Moving Non-SMS-Managed Data Sets

If the data set you are moving is to be non-SMS-managed, use the NULLSTORCLAS and BYPASSACS keywords on the COPY command. By using these keywords, you can make an SMS-managed data set non-SMS-managed. Using NULLSTORCLAS and BYPASSACS also prevents a non-SMS-managed data set from becoming SMS-managed.

## Moving to Preallocated Data Sets

In some cases, you might want to copy data sets to preallocated targets. However, integrated catalog facility catalogs, and system data sets that are named SYS1.\* cannot be copied to preallocated data sets unless the PROCESS(SYS1) keyword is specified.

### Rules for Moving to Preallocated Target Data Sets

To use a preallocated data set, you must specify the REPLACE keyword, and the preallocated data set name must be identical to the source data set name. You cannot, however, copy a data set to a preallocated target data set within an SMS environment because SMS does not support duplicate data set names.

The rules for moving VSAM and non-VSAM data sets to preallocated data sets follow.

**VSAM Preallocation:** An existing data set qualifies as a preallocated target for a data set copy operation if the cluster name matches and the complete cluster is available on target volumes.

The preallocated data set is usable if any of the following conditions that apply to the data set being processed are met:

- The user is authorized to update the target data set.
- The cluster types match.
- The number of components match.
- The key length and offset match.
- The KEYRANGES match.
- None of the components are multivolume.
- Sufficient space is available for each component.
- Key sequential data sets (KSDS) are reusable or empty.
- Key range data sets are empty.
- The data set is cataloged in the standard order of search, if required for the copy operation.
- The data set has no alternate indexes or paths defined over it (except for a single path defined directly over the base cluster).

**Non-VSAM Preallocation:** An existing data set qualifies as a preallocated target for a data set copy operation if the data set names match, the complete data set is available on target volumes, and:

- For single-volume target qualification, the data set organization is partitioned *or* the data set's volume sequence number in the VTOC is 1 and the last volume flag is on.
- For multivolume target or single-volume target with the last volume flag off, the data set is cataloged in the standard order of search. All volume serial numbers returned by a locate operation on the data set are in the output volume list. (Candidate volumes are acceptable.)

The preallocated data set is usable if all of the following conditions that apply to the data set being processed are met:

- The user is authorized to update the target data set.
- The DSORG matches.
- For direct access data sets, the target does not exist if the copy operation is done using the IEHMOVE utility. If the RELBLOCKADDRESS keyword is specified for the data set, preallocated targets are allowed.
- For unmovable data sets, extents match exactly when you copy to a like device without specifying the FORCE keyword.
- For movable data sets or unmovable data sets with the FORCE keyword, the amount of allocated space in the target data set is greater than or equal to the amount of allocated space in the source data set.
- For partitioned data sets, the target directory can contain all source members and aliases.
- For preallocated standard user label data sets, the target has more than one extent when the source data set has more than one extent.

If a VSAM or non-VSAM preallocated data set is determined to be unusable, message ADR439E is issued, and the copy operation is stopped only for that data set. No attempt is made to clear or alter the target data set if:

- The source data set is empty.
- The DSORG is not supported.
- The target is preallocated but not empty.

Message ADR363E is issued to inform the user.

### **Specifying Multiple Target Volumes**

When multiple target volumes and the REPLACE keyword are specified, more than one existing data set may qualify as a preallocated target. The first existing data set that qualifies as a preallocated target when you use the OUTDDNAME/OUTDYNAM list order is used as the target data set. For non-VSAM data sets that require catalog verification, the catalog standard order of search determines the data set used as the preallocated target.

The device-selection criteria used for the data set copy operation (same, like, then unlike device preference) is not observed if a preallocated data set target is used.

### **How Keywords Work with Preallocated Targets**

When you use preallocated data sets with the COPY command, some keywords have a different effect and others have no effect at all.

**ALLEXCP and ALLDATA:** If ALLEXCP or ALLDATA is specified and the target is a like device, the data in the source data set is moved to the target. When ALLDATA or ALLEXCP is specified for an extended-sequential data set, data beyond the last-used-block pointer is not retained.

**CATALOG and RECATALOG:** Data set copy operation cannot change the catalog or the catalog status (cataloged or uncataloged) of the preallocated target data set. As a result, the CATALOG and RECATALOG keywords have no effect on preallocated target data sets. (Similarly, passwords and expiration dates of preallocated data sets cannot be changed.)

**NOPACKING:** The NOPACKING keyword is effective only for partitioned data sets. If NOPACKING is specified for preallocated partitioned data sets, the preallocated target must reside on the same or a like device. Processing is stopped for the data set if the target resides on an unlike device. The target is not deleted and reallocated.

**PERCENTUTILIZED:** The PERCENTUTILIZED keyword has no effect when the target data set is preallocated.

**PROCESS(SYS1):** Data set copy operation permits moving SYS1 data sets to a preallocated target.

**REBLOCK:** If a data set qualifies for reblocking when REBLOCK is specified (sequential and partitioned only) and a preallocated target is used, the target block size is overwritten with one of the following values:

- The source data set block size
- A DFSMSdss-selected block size
- A user-selected block size passed by the installation reblock exit
- A system-determined block size

The block size used is determined by the installation reblock exit return code and the reblockable indicator for the data set VTOC entry.

If REBLOCK is not specified, the target BLKSIZE of a non-VSAM data set is overwritten with the source BLKSIZE.

If a partitioned data set is specified with both NOPACKING and REBLOCK keywords, the data set is not reblocked.

**RENAMEUNCONDITIONAL:** RENAMEUNCONDITIONAL has no effect on preallocated target data sets.

**Note:** Data set copy operation does not scratch and reallocate preallocated data sets if the target data set is too small.

---

## Moving Volumes

You can move volumes logically or physically with DFSMSdss.

As with moving data sets, if the output volume has unexpired data sets, you can stop the copy operation or write over the unexpired data sets.

## Logical Volume Copy Operation

To move a volume logically, use the DATASET keyword, specify input volumes with LOGINDDNAME, LOGINDYNAM, INDDNAME, INDYNAM, or STORGRP, and use INCLUDE(\*\*). This method of moving volumes allows you to move data between unlike devices.

Some data sets require special processing when you move them (see “Moving Data Sets with Special Requirements” on page 77). For example:

- Unmovable data sets
- Multivolume data sets
- Integrated catalog facility catalogs
- Data sets beginning with SYS1
- Data sets used by device-dependent application programs

If you use the COPY DATASET command to move a volume and the volume contains such data sets, you must move them in the correct sequence to achieve the expected results.

You may want to process unmovable data sets first, so you can place them at the same track location on the target device. Move user catalogs only when acquiesced. In addition, do not move catalogs together with the data sets cataloged in them.

**Note:** Some data sets are not eligible for movement by DFSMSdss (for example, VSAM data sets not cataloged in integrated catalog facility catalogs). Others might require special parameters (for example, unmovable data sets).

## Physical Volume Copy Operation

If you do not specify DATASET or TRACKS on the COPY command, the COPY command defaults to FULL and moves the volume physically. You must also specify INDDNAME or INDYNAM to indicate the source volume and OUTDDNAME or OUTDYNAM to indicate the target volume. Full-volume copy can move data only between like devices of equal or greater capacity (for example, from a double capacity 3380 model to a double or triple capacity 3380 model).

**With full-volume copy**, you can physically move volumes only between like devices. However, you can move data:

- From a smaller-capacity IBM 3380 to a larger-capacity IBM 3380
- From a smaller-capacity IBM 3390 to a larger-capacity IBM 3390
- From a smaller-capacity IBM 9345 to a larger-capacity IBM 9345
- From a minivolume or virtual volume to a real volume of like device type, and vice versa, device capacity permitting

**With tracks copy**, you can move data:

- From a larger-capacity IBM 3380 to a smaller-capacity IBM 3380, if the range of data to be processed falls within the capacity of the output device
- From a larger-capacity IBM 3390 to a smaller-capacity IBM 3390, if the range of data to be processed falls within the capacity of the output device
- From a larger-capacity IBM 9345 to a smaller-capacity IBM 9345, if the range of data to be processed falls within the capacity of the output device

When you perform a full-volume copy operation to a DASD that is shared between multiple systems, the DASD should be offline to all systems except the one performing the copy.

When you use the physical volume COPY command, you can specify the COPYVOLID keyword. If you specify the COPYVOLID keyword, the volume serial number of the source volume is copied to the target volume. This ensures that RACF profiles and catalog entries for the data sets on the volume have the correct volume serial number.

**Note:** Changing the volume serial number of a volume causes the operating system to demount the target volume at the end of the copy operation. To use the target volume, you must demount the source volume and mount the target volume.

If you are using record level sharing (RLS), be careful when copying volumes with the FULL or TRACKS keywords. If the target volume has data sets on it that have retained locks or data in the coupling facility associated with them, a full-volume or tracks copy can result in data integrity problems.

## Moving Volumes with FlashCopy

DFSMSDss attempts to use FlashCopy during full volume copy if the source and target devices both support FlashCopy, and the volumes are in the same logical subsystem of an Enterprise Storage Server (ESS). FlashCopy is much faster than traditional methods of data movement, especially when moving large amounts of data.

If possible, DFSMSDss automatically uses FlashCopy, even if you specify the CONCURRENT keyword. If FlashCopy is attempted but is unsuccessful, DFSMSDss will retry the operation using concurrent copy if CONCURRENT is specified. If CONCURRENT is not specified, DFSMSDss uses traditional data movement methods.

For operations where DFSMSDss does not attempt to use FlashCopy, the DEBUG=FRMSG parameter of the EXEC statement can help determine why DFSMSDss cannot use FlashCopy. DEBUG=FRMSG directs DFSMSDss to issue an informational message with a reason code that indicates why it did not attempt to use FlashCopy.

## Moving Volumes with Snapshot

When the source and target devices are in the same RAMAC Virtual Array (RVA) during a physical full volume operation, DFSMSDss may be able to use snapshot to quickly move the data from the source location to the target location. Snapshot is much faster than traditional methods of data movement, especially when you are moving large amounts of data.

DFSMSDss automatically uses snapshot if possible, even if you specify the CONCURRENT keyword. If snapshot is attempted but is unsuccessful, DFSMSDss will retry the operation using virtual concurrent copy if CONCURRENT is specified. If CONCURRENT is not specified, DFSMSDss uses traditional data movement methods.

For operations where DFSMSDss does not attempt to use snapshot, the DEBUG=FRMSG parameter of the EXEC statement can help determine why DFSMSDss cannot use snapshot. DEBUG=FRMSG directs DFSMSDss to issue an informational message with a reason code that indicates why it did not attempt to use snapshot.

## VTOC Considerations

When moving volumes, ensure that the VTOC on the target device is large enough to hold entries for all the data sets to be placed on the target device. The following two sections describe how the size of the target VTOC is affected by DFSMSDss processing.

You can also use the EXTVTOC or NEWVTOC functions of ICKDSF to extend or reallocate the VTOC on a volume if it is not large enough.

**Note:** When doing a full-volume copy operation to DASD, DFSMSDss automatically corrects the free-space information on the volume and rebuilds, as necessary, the VTOC index. DFSMSDss does this whenever the copy is to a larger-capacity DASD from a smaller-capacity DASD or whenever both of the volumes, including volumes of equal capacity, contain a VTOC index. DFSMSDss allocates a large (more than 65 535 tracks) dummy data set in order to cause the free-space information to be recalculated; ignore any IEC614I messages that DFSMSDss generates as part of this procedure.

## Moving Volumes to Like Devices of Equal Capacity

When the source and target devices are of equal capacity, you can use logical or physical copy processing. If you use logical processing, the source VTOC is not copied to the target device. In this case, use ICKDSF to initialize the target device with an appropriately sized VTOC, then perform the logical data set copy operation.

If you use physical processing, the source VTOC is copied to the target device and DFSMSDss automatically rebuilds the VTOC index (if present on the source and target devices). If you determine that the source VTOC is not large enough for the target device, use ICKDSF to initialize the target device with an appropriately sized VTOC, then use logical data set copy to move the volume.

## Moving Volumes to Like Devices of Greater Capacity

When the target device is of greater capacity than the source (for example, if you are moving from a 3390 Model 2 to a 3390 Model 3), you can use logical or physical copy processing. If you use logical processing, the source VTOC is not copied to the target device. In this case, use ICKDSF to initialize the target device with an appropriately sized VTOC, then perform the logical data set copy operation.

If you use physical processing, the source VTOC is copied to the target device if the target VTOC is within the range of the source device (for example, if you are copying a 3390 Model 2 to a 3390 Model 3 and the VTOC on the 3390 Model 3 starts at or before cylinder 2226). In this case, DFSMSDss automatically rebuilds the free-space information in the target VTOC or the indexed VTOC (if present) on the target device to account for the larger size. If you determine that the source VTOC is not large enough, do *one* of the following two things:

- Use ICKDSF to initialize the target device with an appropriately sized VTOC, then use logical data set copy to move the volume.
- Use ICKDSF to initialize the target device with an appropriately sized VTOC that is outside the range of the source device (for example, if you are copying a 3390 Model 2 to a 3390 Model 3, put the VTOC on the 3390 Model 3 at or after cylinder 2227), and then use full volume copy to move the volume. In this case, the size and location of the target VTOC are preserved and DFSMSDss automatically rebuilds the free-space information in the target VTOC or the indexed VTOC (if present).

## Moving Volumes to Unlike Devices

When moving data between unlike devices, you must use logical processing. If you specify DATASET with the COPY command, DFSMSDss does a logical copy operation. To copy all the data on a volume logically, you also need to specify input volumes with LOGINDDNAME, LOGINDYNAM, INDDNAME, or INDYNAM. LOGINDDNAME or LOGINDYNAM is required if you specify SELECTMULTI.

## Moving VM-Format Volumes

You can use DFSMSDss to move VM-format volumes that are accessible to your MVS system. The volumes must have OS-compatible VTOCs starting on track zero, record five. DFSMSDss can only retrieve device information from the OS-compatible VTOC, and cannot interpret any VM-specific information on the volume.

Use the CPVOLUME keyword and specify the range of tracks to be copied with the TRACKS keyword. You can use concurrent copy to move the volume by specifying the CONCURRENT keyword. Because DFSMSDss cannot check access authorization for VM data, CPVOLUME is only allowed with the ADMINISTRATOR keyword.

Exercise caution when using DFSMSDss to copy VM-format volumes because DFSMSDss does not serialize any VM data in any way. You cannot copy VM-format volumes to OS-format volumes, nor can you copy OS-format volumes to OS-format volumes.

---

## Chapter 8. Converting Data to and from SMS Management

DFSMSdss is the primary tool for converting data to and from SMS management. Conversion can be done with or without data movement.

This chapter is organized as follows:

- **“Evaluating Conversion to SMS Management”** discusses the advantages and disadvantages of the two types of conversion.
- **“Conversion by Data Movement” on page 96** describes how to use the COPY and DUMP/RESTORE commands to convert data sets *to* SMS management.
- **“Conversion without Data Movement” on page 97** describes how to use the CONVERTV command to convert volumes *to* SMS management.
- **“Special Data Set Requirements for Conversion to SMS” on page 100** describes some of the data sets that have special requirements for conversion *to* SMS management.
- **“Converting from SMS Management without Data Movement” on page 102** describes how to use the CONVERTV command to convert volumes *from* SMS management.
- **“Special Data Set Requirements for Conversion from SMS” on page 102** describes some of the data sets that have special requirements for conversion *from* SMS management.

---

### Evaluating Conversion to SMS Management

When you convert data to SMS management, the first thing to consider is whether to convert data sets with or without data movement. If you have SMS-managed volumes with sufficient free space, you can convert data sets by simply moving them from non-SMS-managed volumes to SMS-managed volumes. The same is also true if you are converting data from SMS-management. Converting data sets to SMS management by data movement is often preferable because it allows the system to place the data sets for you. This ensures that the data sets are placed on volumes in storage groups that can meet the availability and performance requirements of the data set.

If, however, you do not have sufficient free space on your SMS-managed volumes to convert by data movement, you might have to convert data sets without data movement. The drawback to this method of conversion is that it does not allow the system to place data sets for you. You must ensure that the storage group in which you place the volume can meet the availability and performance requirements of the data sets.

Regardless of how you convert to SMS management, you must determine the eligibility for conversion of your data sets and volumes prior to conversion.

### Data Sets Ineligible for Conversion to SMS

The following data sets cannot be converted to SMS management:

- Absolute track allocation data sets
- Direct with OPTCD=A
- GDS with candidate volumes
- Indexed sequential data sets
- Model DSCBs
- SYS1 storage index data sets (SYS1.STGINDEX)
- Indirectly cataloged data sets

- Uncataloged, multivolume data sets
- VSAM data sets not cataloged in an integrated catalog facility catalog
- VVDS/VTOCIX
- Unmovable data sets

**Notes:**

1. Using the CONVERTV command with the SMS and TEST keywords identifies ineligible data sets without actually converting any data.
2. VVDS/VTOCIX data sets can be SMS-managed, but DFSMSdss cannot be used to convert them, except when using the CONVERTV command to convert the volume that they are on.

## Data Sets Ineligible for Conversion from SMS

The following data sets cannot be converted from SMS management:

- Extended-sequential data sets
- Extended-format VSAM data sets
- Indirectly cataloged data sets
- VSAM data sets that have record level sharing (RLS) information associated with them

**Notes:**

1. Using the CONVERTV command with the NONSMS and TEST keywords identifies ineligible data sets without actually converting them.
2. VVDS/VTOCIX data sets can be non-SMS-managed, but DFSMSdss cannot be used to convert them, except when using the CONVERTV command to convert the volume that they are on.

## Volume Eligibility for Conversion to SMS

A volume is eligible for conversion if it:

- Is a DASD volume
- Is permanently mounted and accessible online
- Has an indexed VTOC
- Is defined in an SMS storage group in an active configuration

---

## Conversion by Data Movement

By using the logical data set COPY or DUMP/RESTORE command, you can move data sets between non-SMS-managed and SMS-managed volumes. When moving data sets to SMS-managed volumes, COPY and RESTORE commands invoke ACS to assign classes to the data sets. This type of conversion to SMS allows the data sets to be placed on the most appropriate SMS-managed volume.

## Converting to SMS Management by Data Movement

When moving data sets to SMS-managed volumes, you can use the COPY or RESTORE command. You can specify storage and management class names with the STORCLAS and MGMTCLAS keywords. You can also specify output volumes with OUTDDNAME and OUTDYNAM. DFSMSdss passes the class names and volume serial numbers to ACS, which might use them in determining the classes and placement of the data set.

This method of converting data sets to SMS management is similar to moving data sets in an SMS-managed environment as described in “Moving SMS-Managed Data Sets” on page 85.

If you use the COPY or RESTORE command on a data set that is ineligible for SMS management and if a non-SMS-managed volume has been specified in the output volume list, DFSMSdss puts it on a non-SMS-managed volume. However, if you specify STORCLAS and BYPASSACS with the COPY or RESTORE command for a data set that is ineligible for SMS management, the copy or restore operation fails.

JOB CAT and STEP CAT are not allowed when you process SMS-managed data sets. For data sets cataloged outside the standard order of search, use the INCAT keyword on the COPY or DUMP command to identify what catalog to search. Use the SELECTMULTI keyword on the COPY or DUMP command to convert multivolume data sets. This allows you to specify only the volume with the primary component on the LOGINDD or LOGINDY parameter. You can use the SPHERE keyword on the COPY DUMP/RESTORE command to convert entire VSAM spheres (if you use SPHERE on the RESTORE command, you must specify it on the corresponding dump as well).

## Conversion from SMS Management by Data Movement

To take a data set out of SMS management with the COPY or DUMP/RESTORE command, you should specify the BYPASSACS and NULLSTORCLAS keywords. This forces DFSMSdss to make the data set non-SMS-managed.

---

## Conversion without Data Movement

Conversion without data movement is divided into two phases: conversion of data sets and conversion of volumes. Convert data sets and the volumes they reside on without moving data by using the DFSMSdss CONVERTV command. You should set up RACF facility class authorization to limit the people who can use the CONVERTV command. When you use the CONVERTV command to perform conversion, it attempts to convert all the data sets on the volume. After all the data sets are processed, the volume is placed in one of the following three states:

- **CONVERTED**—the volume and its data sets are converted to SMS management. A volume can be placed in this state with the CONVERTV command and the SMS keyword.
- **INITIAL**—new allocations cannot be made to the volume and, although users can access their data sets, the data sets cannot be extended to other volumes. A volume may be placed in this state because you have used the CONVERTV command with the PREPARE keyword to reduce activity to the volume prior to conversion. A volume may also be placed in this state if you are attempting to convert it but it contains data sets that are not eligible for conversion.
- **NONSMS**—the volume and its data sets were taken out of the CONVERTED or the INITIAL state and are non-SMS-managed. A volume can be placed in this state with the CONVERTV command and the NONSMS keyword.

## Simulating Conversion

Before you convert a volume to SMS management, you should simulate the conversion to ensure that all the data sets on the volume are eligible for conversion to SMS. In addition, simulating conversion shows you the classes ACS would assign to the data sets eligible for conversion.

You can simulate conversion by using the CONVERTV command with the SMS and TEST keywords. If the volume is ineligible for conversion, the data sets on the volume are still examined to determine their eligibility for conversion (provided the volume is permanently mounted and online).

When you use CONVERTV SMS TEST, you are in the ACS CONVERT environment. Only the storage class and management class ACS routines are executed. (See “ACS Variables Available during RESTORE and CONVERTV Processing” on page 116 for a list of variables available to ACS routines during CONVERTV processing.)

Simulated conversion creates a report that identifies data sets ineligible for conversion. For a sample of this report, see “SMS Report” on page 100. Note that this report indicates the management class and storage class that would be assigned to each data set. Careful analysis of this report allows you to determine if your ACS routines will assign appropriate classes to the data sets before doing the actual conversion.

Move data sets unsupported by SMS off the volume prior to actual conversion. Other data sets (for example, uncataloged data sets) can be made eligible for conversion by taking some action (for example, using the CATALOG keyword to catalog uncataloged data sets).

If you have ineligible data sets on a volume and you run the CONVERTV function with SMS, DFSMSdss still converts the eligible data sets on the volume. It then puts the volume in the INITIAL state. You must then take action to make the ineligible data sets eligible for conversion or move them off the volume. Once all the ineligible data sets are dealt with, you can run CONVERTV processing again to complete the conversion.

## Preparing a Volume for Conversion

Before you convert a volume to SMS management, you should reduce the amount of activity to the volume being converted. The CONVERTV command with the SMS keyword automatically places the volume in a state of reduced activity before doing the actual conversion. You might, however, want to reduce activity without doing the actual conversion (for example, if you want to simulate conversion). Do this by specifying the PREPARE keyword on the CONVERTV command.

Specifying PREPARE prevents data sets from extending and new allocations from being made on the volume. However, users can still access the data on the volume from either the SMS system or a system sharing the volume.

When you use PREPARE, a report is generated that tells you the volumes that have been placed in the INITIAL state. If any of the volumes are ineligible to be placed in the INITIAL state, the report also lists them and the reason they were ineligible (for example, they did not have an indexed VTOC or were offline).

If you use the TEST keyword with PREPARE, you still get the report indicating which volumes would and would not be placed in the INITIAL state, but the PREPARE is not actually performed. You can then take some action to make those volumes eligible or simply not run PREPARE against those volumes.

The CONVERTV command with the NONSMS keyword reverses the effect of PREPARE and takes a volume out of the INITIAL state.

---

## Converting to SMS Management without Data Movement

To convert data to SMS management, use the CONVERTV command with the SMS keyword. (Because SMS is the default for the CONVERTV command, you can simply specify CONVERTV.) Of course, the volume and all its data sets must be eligible for conversion to successfully run CONVERTV with SMS.

If the volume is eligible for conversion, the INITIAL indicator on the volume is set. This means the volume is in the same state as when you specify the CONVERTV command with the PREPARE keyword. When a volume has its INITIAL indicator set on, DFSMSdss begins processing the data sets on the volume.

If a data set is eligible for conversion, ACS is called to assign SMS classes to the data set. When you use the CONVERTV command with SMS, you are in the ACS CONVERT environment. The storage class ACS routine is executed first. If the storage class assigned is not null, the management class ACS routine is executed. (See “ACS Variables Available during RESTORE and CONVERTV Processing” on page 116 for a list of variables available to ACS routines during CONVERTV processing.)

RACF checks if the RESOWNER of a given data set is authorized to define the data set with the given STORCLAS, MGMTCLAS, or both. Ensure that the RESOWNER has the correct authority.

If no errors occur, the catalog entry for the data set is updated to include the classes. For VSAM data sets, the catalog entry is updated to indicate that it is SMS-managed. For non-VSAM data sets, a catalog entry is added that indicates the data set is SMS-managed. After the catalog updates and additions are successfully made, the data set's VTOC entry is updated to indicate it is SMS-managed.

If a VSAM data set has the guaranteed-space attribute, a check is done to verify the eligibility of its candidate volumes. If this check fails, the data set is not converted to SMS management. Non-VSAM data sets have candidate volumes in their catalog entries made nonspecific.

When DFSMSdss encounters a data set that is not eligible for conversion, it does not process the data set, but it continues to process other data sets on the volume. The only time conversion of data sets stops is when an error prevents ACS from returning class information for any data set.

DFSMSdss does not mark a volume as SMS-managed until all the data sets on the volume are SMS-managed. If a volume contains data sets that are ineligible for conversion, you must take some action to make them eligible or move them off the volume. You can then resubmit the CONVERTV command to convert any data sets not already converted and mark the volume as an SMS-managed volume.

On subsequent invocations of CONVERTV processing, DFSMSdss processes only those data sets not yet converted unless you specify the REDETERMINE keyword. If REDETERMINE is specified, DFSMSdss processes data sets already converted if their SMS management class or SMS storage class do not match those returned by the current ACS routines and data sets not yet converted. You may want to do this if your ACS routines changed since the last time you ran the CONVERTV operation on the volume.

## SMS Report

Figure 3 shows a sample report generated by DFSMSDss during CONVERTV SMS processing.

```
PAGE 0001      5695-DF175  DFSMSDSS V2R10.0 DATA SET SERVICES      1999.211 14:55
  CONVERTV      -
    SMS         -
    DYNAM(D9S060)
ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'CONVERTV'
ADR109I (R/I)-RI01 (01), 1999.211 14:55:22 INITIAL SCAN OF USER CONTROL STATEMENTS COMPLETED.
ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
ADR006I (001)-STEND(01), 1999.211 14:55:22 EXECUTION BEGINS
ADR860I (001)-KVSMS(01), PROCESSING BEGINS ON VOLUME D9S060
ADR873I (001)-KVSMS(01), VOLUME D9S060 IN STORAGE GROUP XFMT9SSG IS ELIGIBLE FOR CONVERSION TO SMS MANAGEMENT
ADR877I (001)-KVSMS(01), THE FOLLOWING DATA SETS ON VOLUME D9S060 WERE SUCCESSFULLY PROCESSED
      PUBSEXMP.ESDS.S01          CATALOG: TEST.CAT.PUBSEXMP
      STORCLAS: XFMT9SSC        MGMTCLAS: NONE
      PUBSEXMP.KSDS.S01          CATALOG: TEST.CAT.PUBSEXMP
      STORCLAS: XFMT9SSC        MGMTCLAS: NONE
      TEST.CAT.PUBSEXMP          CATALOG: TEST.CAT.PUBSEXMP
      STORCLAS: XFMT9SSC        MGMTCLAS: NONE
      PUBSEXMP.SAM.S01           CATALOG: TEST.CAT.PUBSEXMP
      STORCLAS: XFMT9SSC        MGMTCLAS: NONE
      PUBSEXMP.PDS.S01           CATALOG: TEST.CAT.PUBSEXMP
      STORCLAS: XFMT9SSC        MGMTCLAS: NONE
ADR885I (001)-KVSMS(01), VOLUME D9S060 HAS BEEN SUCCESSFULLY CONVERTED TO SMS MANAGEMENT

PAGE 0002      5695-DF175  DFSMSDSS V2R10.0 DATA SET SERVICES      1999.211 14:55
ADR892I (001)-KVRPT(01), THE STATUS OF EACH VOLUME IS AS FOLLOWS
      VOLUME          FINAL STATUS          REASON FOR FAILURE
      -----
      D9S060 - CONVERTED      SMS
ADR006I (001)-STEND(02), 1999.211 14:55:23 EXECUTION ENDS
ADR013I (001)-CLTSK(01), 1999.211 14:55:23 TASK COMPLETED WITH RETURN CODE 0000
ADR012I (SCH)-DSSU (01), 1999.211 14:55:23 DFSMSDSS PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
```

Figure 3. SMS Report

---

## Special Data Set Requirements for Conversion to SMS

Some data sets have special requirements for conversion to SMS management. The sections below describe the special considerations for converting these data sets to SMS management.

### VSAM Sphere Eligibility

A VSAM sphere is considered to be a single data set by the CONVERTV command. As a result, either all the data sets of the sphere are converted or none of them are.

If any of the following parts are ineligible for conversion, then all the clusters that compose the sphere are ineligible for conversion:

- Components of a base cluster
- Alternate indexes related to the base cluster
- Alternate index components
- Paths relating alternate indexes to the base cluster

You must direct all parts of a VSAM sphere (the base cluster, base cluster components, alternate indexes, alternate index components, and paths) to the same catalog by using an alias. If they are not directed to the same catalog, the sphere

cannot be converted to SMS management. To correct this problem you can either rename the data sets in the sphere, or add or delete catalog aliases, and rerun the CONVERTV command.

## Multivolume Data Sets

If you do not specify SELECTMULTI, all volumes must be included in DDNAME or DYNAM volume lists.

If you specify input volumes (with either the DDNAME or DYNAM volume list), a data set is selected based on the following criteria:

- When you either specify SELECTMULTI(ALL) or specify input volumes without specifying the SELECTMULTI keyword, **all** of the volumes that contain a part of the non-VSAM data set or VSAM base cluster must be in the volume list.
- When you specify SELECTMULTI(ANY), **any part** of the non-VSAM data set or VSAM base cluster can be on a volume in the volume list.
- When you specify SELECTMULTI(FIRST), the volume list must include the volume that contains either the **first part** of the non-VSAM data set or the primary data component of the base cluster for a VSAM sphere.

Multivolume data sets are not eligible for conversion if any part of the data set resides on volumes that:

- Do not have indexed VTOCs
- Are not defined in an SMS storage group
- Are defined to a different storage group
- Are not permanently mounted and online

If the previous requirements are satisfied, DFSMSdss verifies that all the volumes on which the data set resides:

- Are permanently mounted and online
- Have indexed VTOCs
- Are defined to the same storage group

If all these criteria are met, the data set is converted to SMS management.

### Notes:

1. If SELECTMULTI(FIRST) or SELECTMULTI(ANY) is specified, volumes not specified in the DDNAME or DYNAM volume lists are put in the INITIAL state following a successful conversion of the data set to SMS (unless the volume is already in the INITIAL or SMS state.)
2. If SELECTMULTI is not specified or if SELECTMULTI(ALL) is specified, volumes not specified in the DDNAME or DYNAM volume lists are *not* put in the INITIAL state.

DFSMSdss cannot determine whether or not a volume being converted is a candidate volume for one or more data sets in the system. If such a volume is converted, DFSMSdss cannot ensure consistent conversion for all of the volumes of the data set (or sets) for which the volume is a candidate. This can result in a data set having both SMS-managed and non-SMS-managed volumes in its volume list, which can cause the data set to become unusable.

To avoid this situation when performing CONVERTV operations, if you specify any volume of a multivolume data set in the list of volumes to be converted, ensure that you also include at least one of the primary volumes of the data set. This allows DFSMSdss to ensure that all of the volumes of the data set are converted consistently.

## GDG Data Sets

Generation data groups (GDGs) require special consideration while being cataloged or uncataloged during SMS conversion. Uncataloged GDGs are converted to SMS management, but are left uncataloged. Messages ADR877I and ADR879I indicate NOT CATALOGED for the catalog name in the data set name lists for SMS processing.

## Temporary Data Sets

Data set VTOC entries of temporary data sets are updated to indicate uncataloged SMS status.

## VTOC and VVDS

Data set VTOC entries for the VTOC, VTOC index, and VVDS are updated to SMS management.

---

## Converting from SMS Management without Data Movement

If you want to take volumes out of SMS management, you can use the CONVERTV command with the NONSMS keyword. All volumes and most data sets (see note 2 below) are eligible for NONSMS processing. After you execute this command, the volume indicators that designate the volume as an SMS-managed volume are turned off. The active SMS configuration should be updated to remove the volume from its storage group, otherwise data set allocations to the volume will fail. Thereafter, only non-SMS-managed data sets can be allocated to the volume.

As with the SMS keyword, you can specify the TEST keyword with NONSMS. No conversion is actually done, but a report is generated that identifies the data sets that are and are not eligible for conversion from SMS management. The report also indicates whether the volume as a whole is eligible for conversion from SMS management.

To convert a data set from SMS management, the data set's classes are deleted from its catalog entry. Nonspecific volumes also are deleted from the catalog entry. For a VSAM data set, the SMS-related items are deleted from the catalog entry. For a non-VSAM data set, the catalog entries are updated to remove the SMS information. After the catalog and VVDS updates and deletions are made, the VTOC entry is updated to be non-SMS-managed.

**Note:** You cannot specify the CATALOG and REDETERMINE keywords with NONSMS.

---

## Special Data Set Requirements for Conversion from SMS

When being converted from SMS management, some data sets require special consideration. The following sections discuss some of the special requirements for converting data sets from SMS management.

### Multivolume Data Sets

All pieces of a multivolume data set must be converted from SMS management at the same time. You can do this by using the SELECTMULTI keyword.

If you do not specify SELECTMULTI, then you must specify all the volumes in the DDNAME or DYNAM volume list on which the data set resides.

If you specify input volumes (with either the DDNAME or DYNAM volume list) for NONSMS processing, a data set is selected based on the following criteria:

- When you either specify SELECTMULTI(ALL) or specify input volumes without specifying the SELECTMULTI keyword, **all** of the volumes that contain a part of the non-VSAM data set or VSAM base cluster must be in the volume list.
- When you specify SELECTMULTI(ANY), **any part** of the non-VSAM data set or VSAM base cluster can be on a volume in the volume list.
- When you specify SELECTMULTI(FIRST), the volume list must include the volume that contains the **first part** of either the non-VSAM data set or the primary data component of the base cluster for a VSAM sphere.

Those volumes not included in the volume list will be placed in the INITIAL state. Being in the INITIAL state locks all allocations to the volume until all data sets residing on it are converted.

DFSMSdss cannot determine whether or not a volume being converted is a candidate volume for one or more data sets in the system. If such a volume is converted, DFSMSdss cannot ensure that all of the volumes of the data set (or sets) for which the volume is a candidate, are converted consistently. This can result in a data set having both SMS-managed and non-SMS-managed volumes in its volume list, which can cause the data set to become unusable.

To avoid this situation when performing CONVERTV operations, if you specify any volume of a multivolume data set in the list of volumes to be converted, ensure that you also include at least one of the primary volumes of the data set. This allows DFSMSdss to ensure that all of the volumes of the data set are converted consistently.

## GDG Data Sets

When you convert from SMS management, generation data group (GDG) data sets require special consideration with regard to cataloging. Data sets marked as “deferred roll in and rolled out” are uncataloged.

## Temporary Data Sets

Data set VTOC entries for temporary data sets are updated to non-SMS status.

## VTOC and VVDS

Data set VTOC entries for the VTOC, VTOC index, and VVDS are updated to non-SMS status.

---

## Special Considerations for Using Non-SMS-Managed Targets

When moving to non-SMS-managed targets, there are some special considerations for certain data sets:

- Extended-format data sets cannot be moved to a non-SMS-managed target.
- COPY with DELETE and without RENAMEU is not supported for data sets with DFM attributes. DFM attributes are not maintained for non-SMS data sets.



---

## Chapter 9. Managing Space with DFSMSdss

You can use DFSMSdss to help manage your DASD space. This chapter is organized as follows:

- “**Reclaiming DASD Space**” discusses how to use DFSMSdss to reclaim DASD space.
- “**Consolidating Free Space on Volumes**” on page 108 discusses how to use the DEFrag command to reduce fragmentation on volumes.

---

### Reclaiming DASD Space

You can reclaim DASD space with DFSMSdss in the following ways:

- Releasing unused space in data sets
- Compressing partitioned data sets to consolidate unused space at the end of the data sets and then releasing the unused space
- Deleting unwanted data sets
- Combining data set extents

### Releasing Unused Space in Data Sets

The RELEASE command releases allocated but unused space from all sequential, partitioned, and extended-format data sets that you select with INCLUDE, EXCLUDE, or BY criteria. For an explanation of these criteria, see “Choosing Data Sets for Processing—Filtering” on page 18. DFSMSdss selects only data sets that have releasable space. You can also use ISMF to build a list of data sets based on the amount of unused space and to invoke DFSMSdss to release the unused space in them.

Exclude data sets whose last block pointer in the data set VTOC entry is not maintained in the VTOC by using the EXCLUDE keyword. This can occur if you use an access method other than BSAM, QSAM, BPAM, or VSAM. DFSMSdss does not release space for data sets whose last block pointer in the data set entry is 0.

The following options can help you use the release function more effectively:

#### **MINSECQTY(n)**

Allows you to specify that space not be released unless the user’s secondary allocation is greater than or equal to *n*. In this way, you ensure that the user can still add to the data set after the release. The default value for *n* is 1.

#### **MINTRACKSUNUSED(n)**

Allows you to specify that space not be released unless the number of unused tracks is greater than or equal to *n*. Without MINTRACKSUNUSED, space is released if the data set has one or more unused tracks.

**Note:** When space in a data set is released, all unused space is released, not just the amount beyond the minimum unused (as specified by MINTRACKSUNUSED).

To protect the user, DFSMSdss does not release any space in a data set if:

- The data set has the maximum number of used extents. A data set with the maximum number of allocated extents but fewer than the maximum number of used extents will have the unused space released.
- The cylinder-allocated data set has unused tracks but not an entire unused cylinder.
- The data set's name begins with SYS1, unless the PROCESS(SYS1) keyword is specified. To limit the use of PROCESS, you need to set up a RACF facility class profile.

## Compressing a PDS

The COMPRESS command compresses a PDS on a specified volume. Compression removes unused space between members in a partitioned data set. This recovered space is then available for reuse at the end of the data set. Depending on the filtering criteria you specify, you can compress all the partitioned data sets or only some of the data sets. This command is useful for compressing system partitioned data sets before applying maintenance (thus avoiding certain space-related abends). You must not compress the data sets that contain DFSMSdss or IEBCOPY executable code.

The actual PDS compression is done in place. To prevent loss of data if the system or the compression operation abnormally ends during processing, back up your volume or data sets that meet the filtering criteria before using this command.

COMPRESS does not support processing partitioned data sets that:

- Are unmovable
- Have no directory

## Deleting Unwanted Data Sets

You can use the DELETE and PURGE keywords and data set filtering with a *physical* data set dump to delete unwanted data sets from DASD.

**Note:** This does not apply to VSAM data sets, multivolume non-VSAM data sets, or migrated data sets.

On a logical data set dump when using the DELETE keyword, VSAM, non-VSAM, and multivolume data sets are deleted. DFSMSdss cannot be used to delete migrated data sets.

The following steps show how to delete (scratch and uncatalog) all data sets that have expired and all data sets that have not been referred to in the last year. The data sets are not actually moved to a dump volume.

1. JCL requirement:

```
//NOTAPE DD DUMMY
```

The above JCL prevents moving any data sets.

2. Issue the following control statements to delete (scratch and uncatalog) all data sets not referred to in the last year:

```
DUMP INDD(VOL111) OUTDD(NOTAPE) -  
DATASET(BY(REFDT,LE,*, -366)) -  
DELETE PURGE
```

3. Issue the following control statements to delete all expired data sets:

```
DUMP INDD(VOL111) OUTDD(NOTAPE) -  
DATASET(INCLUDE(**) -  
BY(EXPDT,LT,*) -  
DELETE
```

**Note:** You can modify the above example to apply to VSAM and multivolume data sets by omitting the INDD statement or by specifying LOGINDD. This JCL results in a logical data set dump operation.

4. Issue the following control statements to delete uncataloged non-VSAM data sets:

**For a physical data set dump:**

```
DUMP DATASET(INCLUDE(**) -  
BY((DSORG NE VSAM) -  
(CATLG EQ NO))) -  
INDDNAME(DASD1,DASD2) -  
OUTDDNAME(TAPE) -  
DELETE PURGE
```

**Note:** The DD named TAPE can be a DD dummy if a dump of the uncataloged data sets is not wanted. DASD1 and DASD2 identify the input volumes. Because a physical data set dump processes each volume in order one at a time, it can handle multiple, uncataloged, single-volume data sets with the same name when multiple input volumes are specified. It cannot handle a multivolume data set even if all the volumes on which it resides are specified as input volumes.

**For a logical data set dump:**

```
DUMP DATASET(INCLUDE(**) -  
BY((DSORG NE VSAM) -  
(CATLG EQ NO))) -  
LOGINDDNAME(DASD1,DASD2) -  
OUTDDNAME(TAPE) -  
DELETE PURGE
```

**Note:** The DD named TAPE can be a DD dummy if a dump of the uncataloged data sets is not wanted. DASD1 and DASD2 identify the input volumes. A logical data set dump cannot handle multiple, uncataloged data sets with the same name in the same job even when all the volumes on which they reside are specified as input volumes.

A logical dump can handle a legitimate multivolume uncataloged data set if all the volumes on which it resides are specified as input volumes and if there is no cataloged data set by the same name on the system.

## Combining Data Set Extents

The DUMP command used with the DELETE and PURGE keywords scratches and uncatalogs the data sets from DASD after they are dumped. If you restore those data sets to the same DASD, allocation attempts to get the space for the entire data set. If the DASD volume has sufficient contiguous unused space, the allocated space will most likely be in one contiguous extent. Because unmovable data sets are not deleted, however, the volume might be fragmented, preventing a complete restore for all data sets.

If you copy those data sets to the same DASD, allocation attempts to get the space for the entire data set. If the DASD volume has sufficient contiguous unused space, the allocated space will most likely be in one contiguous extent. If you do not specify ALLDATA and ALLEXCP for sequential and partitioned data sets, only used spaces are allocated.

**Note:** Do not use this technique for unmovable data sets such as ABSTR allocated or indexed sequential data sets.

The following steps show how to dump and delete (scratch and uncatalog) all movable non-VSAM data sets, defragment volumes, and restore all movable non-VSAM data sets.

1. Issue the following control statements to dump and delete all movable, single-volume, non-VSAM data sets:

```
DUMP INDD(DASD1) OUTDD(TAPE1) OPTIMIZE(3) -  
DATASET(BY((DSORG,NE,VSAM),(ALLOC,EQ,MOV),(MULTI,EQ,NO))) -  
DELETE PURGE
```

2. Issue the following control statement to defragment the volume:

```
DEFRAG DDN(DASD1)
```

3. Issue the following control statements to restore and catalog all dumped data sets:

```
RESTORE INDD(TAPE1) OUTDD(DASD1) -  
DATASET(INCLUDE(**)) -  
CATALOG
```

You can specify the CONSOLIDATE keyword when you defragment the volume as an alternative to the above three-step approach (dump and delete, defragment, and restore).

Issue the following control statement to defragment the volume, and perform extent reduction if possible:

```
DEFRAG DDN(DASD1) CONSOLIDATE
```

---

## Consolidating Free Space on Volumes

Because of the nature of allocation algorithms and the frequent creation, extension, and deletion of data sets, free space on DASD volumes becomes fragmented. This results in:

- Inefficient use of DASD storage space
- An increase in space-related abends (abnormal endings)
- Performance degradation caused by excessive DASD arm movement
- An increase in the time required for functions that are related to direct access device space management (DADSM)

By using the DEFRAG command, you can consolidate the free space on volumes and avoid this problem. The DEFRAG command relocates data set extents on a DASD volume to reduce or eliminate free-space fragmentation, and prints a report about free space and other volume statistics. Also, you can specify which data sets,

if any, are to be excluded from data-set-extent relocation. Data set extents are not combined as a result of DEFrag processing unless you also specify the optional CONSOLIDATE keyword.

When you specify the CONSOLIDATE keyword, the DEFrag command attempts to consolidate data set extents and perform extent reduction for data sets that occupy multiple extents. When you process a volume with the CONSOLIDATE keyword, DFSMSDss searches each moveable data set. A data set that has multiple extents and is not excluded from data movement is eligible for extent consolidation and extent reduction. For eligible data sets that consist of contiguous extents that are in sequential order, DFSMSDss consolidates without extent relocation. Otherwise, eligible data sets are relocated if enough contiguous free space exists on the volume to hold the resulting data set. When DFSMSDss has completed consolidation for all eligible data sets, DEFrag processing consolidates the remaining free space extents using existing DEFrag algorithms.

## When to Run the DEFrag Function

Although the DEFrag function can be run at any time on a volume, it does lock the VTOC (via RESERVE) and VVDS if it exists on the volume. The DEFrag function also serializes on data sets via ENQ or dynamic allocation. These activities might cause excessive wait time for other jobs to update the VTOC. Therefore, times of low system activity are best for DEFrag runs.

When a volume being defragmented is snapshot-capable (for example, the volume is located on RAMAC Virtual Array), DFSMSDss uses snapshot to move the extents instead of using traditional I/O. Snapshot provides a significant performance improvement to the DEFrag function.

## Data Sets Excluded from DEFrag Processing

DFSMSDss automatically excludes and does not relocate the following types of data sets in a DEFrag operation:

- User-specified data sets (EXCLUDE)
- Data sets that do not satisfy all BY criteria
- Indexed sequential data sets
- VSAM data sets not cataloged in an integrated catalog facility catalog
- Key range VSAM data sets
- Catalogs (system and user)
- The VTOC index data set
- RACF control data sets (any data set with a name in the form SYS1.RACF\*.\*\*) )
- Page, swap, and SYS1.STGINDEX data sets
- VSAM volume data sets (VVDS)
- Unmovable data sets
- Data sets allocated by absolute track
- Data sets that it cannot serialize for exclusive access
- VSAM data sets that have Record Level Sharing (RLS) information associated with them (only the first extent of this type of data set is excluded from the DEFrag operation)

Because the DEFrag function does not relocate these data sets, the effectiveness of a DEFrag run is affected by their presence.

Place the following data sets in the EXCLUDE list if they are present on the volume being defragmented:

1. If you plan to defragment a volume containing the active RACF database, you must place the RACF database data sets in the EXCLUDE list.

2. Any data that has been defined as a retained DLF object for use with Hiperbatch™.

**Note:** Exclude system data sets that are opened and are being accessed without an enqueue.

## DEFRAG Options

You can use the following keywords to make more efficient use of the DEFRAG command:

- CONSolidate** Perform extent reduction by consolidating multiple extent data sets when possible.
- DYNALLOC** The use of dynamic allocation, rather than enqueue, to serialize the use of data sets does not always provide cross-system serialization.
- FRAGI(n)** Perform a DEFRAG operation only if the fragmentation index is more than *n*.
- MAXMOVE(n,p)**  
Stop the DEFRAG run when *n* contiguous free tracks are assembled. If *n* contiguous free tracks already exist, the DEFRAG function tries to further reduce the fragmentation of the volume but no more than *n* tracks are relocated. If more than *n* tracks must be relocated, no DEFRAG is performed.
- n** The number of free tracks that DFSMSDss is to try to assemble in a contiguous area.
- p** The number of passes DFSMSDss is to make in attempting to assemble the tracks.
- PASSDelay** Specify the time delay between the passes (*p*) specified in MAXMOVE(*n,p*) to allow access to the volume between passes.
- WAIT(s,r)** If the data set is unavailable, wait *s* seconds before retrying to obtain control of it and retry only *r* times.

To determine the fragmentation index of a volume without actually performing the DEFRAG operation, code the NORUN parameter on the EXEC statement in your JCL. In addition to the fragmentation index, the NORUN parameter lists the number of free cylinders, the number of free tracks, the number of free extents, the largest free extent size, and the percentage of free space on the volume. A map of the volume with the CCHH location of each data set or free space (in ascending order from cylinder 0, track 0) is also issued.

### General Hints

1. If you want the DEFRAG function to perform in the shortest period of time or to create the largest single free-space extent, perform only the first pass. Do so by coding the MAXMOVE(*n*) parameter, using a very high value (9999) for *n*. When the value is higher than the DEFRAG function can assemble, the process stops at the end of the first pass. For example:  
DEFRAG DYNAM(388002) MAXMOVE(9999)
2. Experimenting with the DEFRAG FRAGI and MAXMOVE parameters will allow you to compare results when you operate on DASD with different fragmentation characteristics. The fragmentation index that can be specified by the DEFRAG options represents a number between 0 and 1 and can be one to three digits long. (FRAGI(333) represents 0.333 and FRAGI(3) represents 0.3. The

recommended basic DEFRAG parameters are to let MAXMOVE default and use FRAGI(3). To defragment DASD volume 388001, the following could be used:

```
DEFRAG DYNAM(388001) FRAGI(3)
```

## Serialization

The DEFRAG command serializes access to the VTOC. The DEFRAG command releases this serialization before it generates the ending statistics provided by message ADR213I. Therefore, the information in message ADR213I may not reflect the state of the volume at the completion of DFSMSDss processing because another job may allocate or delete data sets on the processed volume between the time the serialization is released and the ending statistics are obtained. The serialization scheme is described in *OS/390 DFSMSDss Storage Administration Reference*.

The DEFRAG command does a RESERVE on the VTOC to serialize access to the VTOC. The DEFRAG command also serializes access to each data set before relocating the extent of a data set. The enqueue scheme used by the DEFRAG function ensures integrity on a single processor but does not ensure integrity for data sets on DASD shared between processors. This is due to the use of an ENQ scope of SYSTEM for the SYSDSN resource name. To ensure the integrity of data sets on a shared DASD, you must do one of the following:

- Vary the volume offline from all processors except the one on which DEFRAG runs. After the DEFRAG function finishes, you can vary the volume back online for the other processors.
- In either a JES2 or a JES3 environment, you can use multisystem GRS (or equivalent function) to convert the scope of all enqueues with a resource name of SYSDSN from SYSTEM to SYSTEMS by placing SYSDSN in the GRS SYSTEM INCLUSION resource name list (RNL). This allows all systems in the GRS ring to be made aware of all SYSDSN enqueues. The default GRS System Inclusion RNL includes SYSDSN. However, ensure that this has not been changed on your system before using the DEFRAG command on a volume shared between two or more processors.

**Note:** GRS must not be used to convert the scope of any of DEFRAG function's SYSTEMS enqueues (including SYSVSAM) to SYSTEM by placing the resource names in the GRS RNL. GRS may, however, be used to convert DEFRAG function's RESERVE on SYSVTOC to a simple enqueue with a scope of SYSTEMS by including it in the GRS "RESERVE CONVERSION RNL". If you choose not to do this, you can avoid doing two global serializations on the volume's VTOC by placing SYSVTOC in the GRS Systems Exclusion RNL, thus changing RESERVE's global enqueue to a local enqueue. Refer to *OS/390 MVS Planning: Global Resource Serialization*, or *OS/VS2 MVS Planning: Global Resource Serialization* to find out which restrictions apply to enqueues and dequeues. The DYNALLOC serialization mechanism of DFSMSDss does not solve all cross-system serialization problems. GRS (Global Resource Serialization) is recommended with shared DASD.

- If you are running on a system using JES2 and are not using multisystem GRS (or equivalent function), you can use DEFRAG function's BY filtering to specifically include or exclude data sets from processing. Both creation date and last-referenced date criteria are needed to ensure that only those data sets that are not in use are selected for DEFRAG function processing. For example, if you choose to defragment a volume with typical TSO or batch data sets, you could select only those data sets that were created and referenced more than two days previously. Two days should be a minimum selection age because of the level of

precision of the creation date. In the following example, data set A.B.C is created two minutes before the DEFRAG function begins.

```

TIME OF DAY    ACTION
  2359         Create data set A.B.C
.
.
0001         Begin DEFRAG BY(CREDT,LT,-1). Data set A.B.C is selected
              because it has now been one day since creation

```

However, because there is a date change between the two actions, A.B.C was selected for the DEFRAG operation. A two day delay enforces a convention that more than 24 hours must pass before a data set is eligible for DEFRAG. In an environment with TSO and batch data sets, the probability that one of these data sets will be open more than 24 hours is low. In the following example, choosing data sets that have not been manipulated within the preceding 24 hours causes DEFRAG processing only for those data sets on volume SHARE3 that were created and referenced more than two days previously and that are not temporary data sets.

```

DEFRAG BY(LIST((CREDIT LT *,-2),(REFDT LT *,-2))) -
EXCLUDE(LIST(SYS8*.T*.**)) DYNAM(SHARE3)

```

The two-day criterion is probably sufficient for TSO and batch type data sets. However, if there are applications that use the volumes being defragmented, consider setting the delay time to the maximum time that the application would have a data set open.

- If you are running on a system using JES3 with MDS enabled and are not using multisystem GRS (or equivalent function), you can use the DEFRAG command DYNALLOC keyword to provide serialization for data sets on shared DASD.

**Note:** Not all data sets allocated within a JES3 environment are known to the global. The use of the DYNALLOC keyword does not provide cross-system serialization for these data sets.

- Allocation of existing (old) data sets whose names appear in the RESDSN and DYNALDSN lists are not protected by the DYNALLOC serialization mechanism of DFSMSdss. DEFRAG processing for these data sets can be prevented by placing their names (or filters for the names) in the EXCLUDE LIST for the DEFRAG command.
- New data sets created with nonspecific allocation (no volume serial supplied) are not protected by the DYNALLOC serialization mechanism of DFSMSdss. However, you can use BY filtering with the DEFRAG command to specifically include or exclude data sets from processing. In the following example, the DEFRAG function processes only those data sets on volume SHARE3 that were created more than two days before.

```

DEFRAG BY(LIST(CREDIT,LT,*, -2)) DYNALLOC DYNAM(SHARE3)

```

You can also use the EXCLUDE parameter to avoid processing data sets that were created by long-running programs or subsystems more than two days previously but that are still allocated. In the following example, if the newly created data sets are temporary, the DEFRAG operation processes only those data sets on volume SHARE3 that were created more than two days before and are not temporary data sets.

```

DEFRAG BY(LIST(CREDIT,LT,*, -2)) -
EXCLUDE(LIST(SYS8*.T*.**)) DYNALLOC DYNAM(SHARE3)

```

You can ensure successful DEFRAG processing of volumes having a significant number of free or allocated extents by specifying appropriate SIZE and REGION

parameters in the EXEC statement. If you receive a message that the region size is not large enough, specify a larger region size in the EXEC or JOB statement and rerun your job.

**Note:** During DEFRAG processing, a data set VTOC entry with the unique name “SYS1.DFDSS.DEFRAG.xxxxxxx.volser.DUMMY” is allocated on the volume being defragmented. This data set is not cataloged but is automatically deleted after a successful run. If a job is canceled or abnormally ends, this data set remains on the volume. After the restart, DADSM functions might fail with message IEC602. To correct this problem or to delete the “SYS1.DFDSS.DEFRAG.xxxxxxx.volser.DUMMY” entry, rerun the DEFRAG function on the volume.

## Security Considerations

For security purposes, the data set tracks used before the relocation are erased after relocation under these conditions:

- When the OS/390 Security Server (RACF element) Version 1 Release 7 is installed and either:
  - The data set was defined to RACF with the RACF ERASE option
  - The VSAM data set has the ERASE attribute
  - The data set is password protected. (In this case, if the data set is also defined to RACF, the RACF ERASE option is taken. Table 9 provides more detail.)

*Table 9. Data Set Erase Table for DEFRAG with OS/390 Security Server (RACF element) Version 1 Release 7*

	Password Protected	Defined ERASE	RACF Protected	Erased on Scratch
User Install Exit=ERASE (default)	No	No	No	No
	Yes	No	No	Yes
		Yes	No	Yes
	No	No	Yes (=ERASE)	Yes
	No	No	Yes (=NOERASE)	No
	No	Yes	Yes (=ERASE)	Yes
	No	Yes	Yes (=NOERASE)	Yes
	Yes	Yes	Yes (=ERASE)	Yes
	Yes	Yes	Yes (=NOERASE)	Yes
User Install Exit=NOERASE	No	No	No	No
<b>Note:</b> The catalog entry contains the ERASE attribute specified when the data set was defined (VSAM only).				

You can prevent the tracks from being erased by using the installation options exit routine. Refer to the installation options exit routine in *OS/390 DFSMS Installation Exits*.

The DEFRAG function does not relocate protected data sets unless:

- You have RACF DASDVOL update access to the volume.
- You have RACF DATASET read access to the data sets on the volume.

- You specify the read or update password for password-protected data sets, or the Installation Authorization Exit Routine supplied with DFSMSdss is changed to allow relocation of protected data sets.

When RACF DASDVOL class is active and a profile exists for the volume, a DASDVOL authorization failure causes the DEFrag task to abend with a system code 913. This happens regardless of RACF data set access authority.

---

## Maximizing Track Utilization by Reblocking Data Sets

DFSMSdss provides a REBLOCK keyword that allows users to maximize the track usage by data sets during copy and restore processing. When REBLOCK is specified on a fully or partially qualified name of a sequential or partitioned data set during copy or restore processing, DFSMSdss will choose an optimal block size for the data set and the device. However, the installation reblock exit can specify that a different block size be used.<sup>5</sup>

REBLOCK is ignored for:

- Unmovable data sets
- Data sets with record format of U<sup>5</sup>, V, VB, VBS, or F
- Partitioned data sets with note lists<sup>5</sup>
- Partitioned data sets that are also specified in the NOPACKING keyword

Partitioned load modules can be reblocked in copy operations, even if they have NOTELISTS.

The reblockable indicator in the data set's VTOC entry also determines whether a data set is to be reblocked or not. When the indicator is on, the data set is always reblocked to a system determined, optimal block size, except when the data set:

- Is a partitioned data set that is also specified in the NOPACKING keyword
- Is an unmovable data set
- Has a record format of V, VS, VBS, or F

The installation reblock exit is not called if the reblockable indicator is ON. Refer to *OS/390 DFSMSdss Storage Administration Reference* for a more detailed description of the operations of the installation reblock exit routine.

A PDSE being converted to a PDS will always be reblocked except when the data set has a record format of V, VS, VBS, or F; or when the data set's record length is '0'.

---

<sup>5</sup>. Except for partitioned load modules during copy operations.

---

## Appendix. ACS Routine Information

General-use Programming Interface and Associated Guidance Information is contained in this appendix.

This appendix lists the variables available to automatic class selection (ACS) routines during DFSMSdss copy, restore, and CONVERTV operations, and provides additional information about ACS routine processing. This information is provided for guidance purposes only. It is not associated with any interface provided by DFSMSdss. For details on writing ACS routines, refer to *OS/390 DFSMSdss Storage Administration Reference*.

Messages generated by ACS routines are not printed by DFSMSdss unless the ACS routine returns a nonzero return code.

---

### ACS Variables Available during Copy Function

When automatic class selection (ACS) is invoked during a DFSMSdss copy function, the following variables, as shown in Table 10 are passed to the ACS routines.

*Table 10. Variables Passed to ACS Routines during DFSMSdss Copy Function. The following variables are not available to the storage group ACS routine: &ACCT\_JOB, &ACCT\_STEP, &DD, &JOB, &PGM, and &XMODE.*

Variable Name	Description
&ACCT_JOB	Accounting information from the JOB statement.
&ACCT_STEP	Accounting information on the EXEC statement.
&ACSENVIR	Environment in which ACS was invoked. Set to 'ALLOC.'
&ALLVOL	Output volume serial numbers. References the same volume list as &ANYVOL, but when used in a comparison, returns true only if all volume serial numbers satisfy the condition. &ALLVOL is not available to the storage group ACS routine when VOLCOUNT(ANY) is specified.
&ANYVOL	Output volume serial numbers. References the same volume list as &ALLVOL, but when used in a comparison, returns true if any volume serial numbers satisfy the condition. &ANYVOL is not available to the storage group ACS routine when VOLCOUNT(ANY) is specified.
&APPLIC	Application identifier associated with the data set (available only if RACF is installed).
&DD	DDNAME
&DEF_DATACLAS	Default data class name (available only if RACF is installed).
&DEF_MGMTCLAS	Default management class name (available only if RACF is installed).
&DEF_STORCLAS	Default storage class name (available only if RACF is installed).
&DSN	Data set name.
&DSNTYP	Data set name type (for example: EXT, HFS, LIBRARY, PDS, or null).
&DSORG	Data set organization.

Table 10. Variables Passed to ACS Routines during DFSMSdss Copy Function (continued). The following variables are not available to the storage group ACS routine: &ACCT\_JOB, &ACCT\_STEP, &DD, &JOB, &PGM, and &XMODE.

Variable Name	Description
&DSOWNER	RACF owner or group that is considered the data set owner (available only if RACF is installed).
&DSTYPE	Data set type (for example, GDS, PERM, or TEMP).
&EXPDT	Expiration date.
&GROUP	Group identifier from the JOB statement.
&HLQ	High-level qualifier of the data set name.
&JOB	Job name, started task name, or TSO user ID from the JOB statement.
&LLQ	Low-level qualifier of the data set name.
&MAXSIZE	Maximum size of data set in kilobytes. For non-VSAM data sets, the primary value plus 15 secondary extents, or 122 secondary extents for PDSE and extended-sequential data sets. For VSAM data sets, the primary value plus 122 secondary extents. See "Using SIZE and MAXSIZE Variables" on page 118 for more information about this value.
&NQVAL	Number of qualifiers in the data set name.
&NVOL	Number of output volumes specified by the user.
&PGM	Program name from the EXEC card.
&RECOG	Data set record organization.
&RETPD	Retention period.
&SIZE	Size of the data set in kilobytes. See "Using SIZE and MAXSIZE Variables" on page 118 for more information about this value.
&UNIT	Actual unit name (not esoteric names).
&USER	User ID from the JOB statement or the user ID propagated from the environment when a security package, such as RACF, is active.
&XMODE	Execution mode (for example, TSO, BATCH, or TASK).

## ACS Variables Available during RESTORE and CONVERTV Processing

When ACS is invoked during DFSMSdss RESTORE or CONVERTV processing, the variables shown in Table 11 are passed to the ACS routines.

Table 11. Variables Passed to ACS Routines during DFSMSdss Restore and CONVERTV Processing

Variable Name	Description
&ACSENVIR	Environment in which ACS was invoked. Set to 'RECOVER' for RESTORE. Set to 'CONVERT' for CONVERTV.
&ALLVOL	For restore processing, the output volume serial numbers. For CONVERTV processing, the volumes on which the data set resides. References the same volume list as &ANYVOL, but when used in a comparison, returns true only if all volume serial numbers satisfy the condition. &ALLVOL is not available to the storage group ACS routine when VOLCOUNT(ANY) is specified.

Table 11. Variables Passed to ACS Routines during DFSMSdss Restore and CONVERTV Processing (continued)

Variable Name	Description
&ANYVOL	For restore processing, the output volume serial numbers. For CONVERTV processing, the volumes on which the data set resides. References the same volume list as &ALLVOL, but when used in a comparison, returns true if any volume serial numbers satisfy the condition. &ANYVOL is not available to the storage group ACS routine when VOLCOUNT(ANY) is specified.
&APPLIC	Application identifier associated with the data set (available only if RACF is installed).
&DEF_DATACLAS	Default data class name (available only if RACF is installed).
&DEF_MGMTCLAS	Default management class name (available only if RACF is installed).
&DEF_STORCLAS	Default storage class name (available only if RACF is installed).
&DSN	Data set name.
&DSNTYP	Data set name type (for example: EXT, HFS, LIBRARY, PDS, or null).
&DSORG	Data set organization.
&DSOWNER	RACF owner or group that is considered the data set owner (available only if RACF is installed).
&DSTYPE	Data set type (for example, GDS, PERM, or TEMP).
&EXPDT	Expiration date.
&GROUP	Group identifier from the JOB statement. (This variable is passed to ACS routines during processing of RESTORE only, not CONVERTV.)
&HLQ	High-level qualifier of the data set name.
&LLQ	Low-level qualifier of the data set name.
&MAXSIZE	Maximum size of data set in kilobytes. For non-VSAM data sets, the primary value plus 15 secondary extents, or 122 secondary extents for PDSE and extended-sequential data sets. For VSAM data sets, the primary value plus 122 secondary extents. See "Using SIZE and MAXSIZE Variables" on page 118 for more information about this value.
&NQUAL	Number of qualifiers in the data set name.
&NVOL	For restore processing, number of volumes specified by the user. For CONVERTV processing, the number of volumes (including candidate volumes) on which the data set resides.
&RECORG	Data set record organization.
&RETPD	Retention period.
&SIZE	Size of the data set in kilobytes. See "Using SIZE and MAXSIZE Variables" on page 118 for more information about this value.
&UNIT	Actual unit name (not esoteric names).
&USER	User ID from the JOB statement or the user ID propagated from the environment when a security package, such as RACF, is active. (This variable is passed to ACS routines during processing of RESTORE only, not CONVERTV.)

## Using SIZE and MAXSIZE Variables

The values for SIZE and MAXSIZE (and the space units the values represent) depend on the type of allocation of the data set. For all VSAM data sets, and non-VSAM data sets allocated in device dependent units (tracks or cylinders), the value represents the number of kilobytes using the maximum block size of the device. For example, the maximum block size for a 3390 is 56664. For all other non-VSAM data sets allocated in device independent units (blocks, average block, AVGREC=U, AVGREC=K, AVGREC=M), the value represents the number of data kilobytes (based on the average block size, specified block size, or 4096 if no block size is available).

The values DFSMSdss computes for SIZE and MAXSIZE may not match that of the original allocation. The values may be different if the device type on which the data set now resides is not the same as either of the following:

- The device type used at allocation, or
- The default device type in the CDS when the original allocation was done

DFSMSdss calculates the value based on the device type where the data set currently resides. DFSMSdss has no way of “knowing” what device type was specified or used for the original allocation.

DFSMSdss calculates SIZE and MAXSIZE variables as follows:

- For **PDS and PDSE data sets** — The values DFSMSdss computes may be different from those computed by SMS because SMS adds space for the directory. Also, DFSMSdss computes MAXSIZE for PDSE data sets based on 122 secondary extents.
- For **VSAM data sets** — DFSMSdss computes the SIZE and MAXSIZE for key-sequenced data sets (KSDS) from the current size and space values of the data component. The index component size is not included. Also, DFSMSdss computes MAXSIZE for VSAM data sets based on 122 secondary extents.
- For **extended-sequential data sets** — DFSMSdss computes MAXSIZE for extended-sequential data sets based on 122 secondary extents.
- For data sets allocated with **AVGREC=U, K, or M** — The size values computed during the initial allocation used the specified average block value. Since the average block size value is not stored anywhere and was only available during the initial allocation, DFSMSdss uses the DCB BLKSIZE value. If the DCB BLKSIZE value is not the same as the average block size value, the values computed for SIZE and MAXSIZE may be different from those computed at initial allocation.

---

## Notices

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:**

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Information Enabling Requests  
Dept. DZWA

5600 Cottle Road  
San Jose, CA 95193 U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

---

## Programming Interface Information

This publication primarily documents information that is NOT intended to be used as a Programming Interface of DFSMSdss.

This publication also documents intended Programming Interfaces that allow the customer to write programs to obtain the services of DFSMSdss. This information is identified where it occurs, either by an introductory statement to a chapter or section or by the following marking:

┌ **Programming Interface information** \_\_\_\_\_

└ **End of Programming Interface information** \_\_\_\_\_

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Information Enabling Requests  
Dept. DZWA  
5600 Cottle Road  
San Jose, CA 95193  
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

---

## Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

DATABASE 2	MVS/ESA
DB2	MVS/XA
DFSMS	OS/390
DFSMSdfp	RACF
DFSMSdss	RAMAC
DFSMSHsm	SecureWay
Hiperbatch	System/390
IBM	System/370
IBMLink	S/370
MVS	S/390

---

# Glossary

This glossary defines technical terms and abbreviations used in DFSMSdss documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMSdss manual or view *IBM Dictionary of Computing*, located at

<http://www.ibm.com/networking/nsg/nsgmain.htm>

## A

**ABEND.** Abnormal end of task. End of a task, a job, or a subsystem because of an error condition that cannot be resolved by recovery facilities while the task is performed.

**ABENDxxx.** The keyword that identifies the abnormal end of DFSMSdss because of a system-detected error.

**ABSTR.** A subparameter of the SPACE parameter in a DD statement. It indicates that specified tracks be assigned to a data set.

**ACCEPT processing.** An SMP/E process necessary for installing the FMIDs. SMP/E ACCEPT processing uses JCL to accept the modules and macros necessary to run the FMIDs. The FMIDs are accepted into the DLIBs from the temporary data sets.

**access method services.** A multifunction service program that is used to manage both VSAM and non-VSAM data sets and integrated catalog facility or VSAM catalogs. It is used to define data sets and allocate space for them; convert indexed-sequential data sets to key-sequenced data sets; modify data set attributes in the catalog; reorganize data sets; facilitate data portability between operating systems; create backup copies of data sets, data set records, and catalog entries; help make inaccessible data sets accessible; list the records of data sets and catalogs; define and build alternate indexes; and convert OS CVOLs and VSAM catalogs to integrated catalog facility catalogs.

**ACDS.** Active control data set.

**ACS.** Automatic class selection.

**alias.** An alternate name for a member of a partitioned data set.

**ALLOC.** A space allocation parameter that indicates type, such as cylinders or tracks.

**alternate index.** In systems with VSAM, a key-sequenced data set containing index entries organized by the alternate keys of its associated base

data records. It provides an alternate means of locating records in the data component of a cluster on which the alternate index is based.

**alternate index cluster.** In VSAM, the data and index components of an alternate index.

**APAR.** authorized program analysis report

**APF.** Authorized program facility.

**application interface.** An interface used to invoke DFSMSdss from another program.

**apply processing.** In SMP and SMP/E, the process, initiated by the APPLY command, that places system modifications (SYSMODS) into the target system libraries.

**attach.** In programming, to create a task that can be performed asynchronously with the performance of the mainline code.

**authorization.** (1) The right granted to a user to communicate with or make use of a computer system. (2) The process of giving a user either complete or restricted access to an object, resource, or function.

**authorized program analysis report (APAR).** A request for correction of a problem caused by a suspected defect in a current unaltered release of a program.

**automatic class selection (ACS).** A mechanism for assigning SMS classes and storage groups.

## B

**backup.** The process of creating a copy of a data set to ensure against accidental loss.

**backup while open.** DFSMSdss can perform backup of data sets that are open for update for a long period of time (like CICS). DFSMSdss can perform a logical data set dump of these data sets even if another application has them serialized.

**base cluster.** In systems with VSAM, a key-sequenced or entry-sequenced data set over which one or more alternate indexes are built.

**basic catalog structure (BCS).** The name of the catalog structure in the integrated catalog facility environment. An integrated catalog facility catalog consists of a BCS and its related VSAM volume data sets (VVDSs).

**basic direct access method (BDAM).** An access method used to directly retrieve or update particular blocks of a data set on a direct access device.

**basic partitioned access method (BPAM).** An access method that can be applied to create program libraries in direct access storage for convenient storage and retrieval of programs.

**basic sequential access method (BSAM).** An access method for storing or retrieving data blocks in a continuous sequence, using either a sequential access or a direct access device.

**BCS.** Basic catalog structure.

**BDAM.** Basic direct access method.

**BLK.** A subparameter of the SPACE parameter in a DD statement. It specifies that space is allocated by blocks.

**block length.** Synonym for block size.

**block size.** (1) The number of data elements in a block. (2) A measure of the size of a block, usually specified in units such as records, words, computer words, or characters. (3) Synonymous with block length. (4) Synonymous with physical record size.

**BPAM.** Basic partitioned access method.

**bpi.** Bits per inch.

**BSAM.** Basic sequential access method.

## C

**CA.** Control area.

**call.** (ISO) The action of bringing a computer program, a routine, or a subroutine into effect, usually by specifying the entry conditions and jumping to an entry point.

**card image.** A one-to-one representation of the hole patterns of a punched card; for example, a matrix in which a one represents a punch and a zero represents the absence of a punch.

**CC-compatible snapshot.** See *virtual concurrent copy*.

**CCHHR.** Cylinder, cylinder, head, head, record.

**CCW.** Channel command word.

**CDE.** Contents directory entry.

**CDS.** Control data set.

**channel command word (CCW).** A doubleword at the location in main storage specified by the channel address word. One or more CCWs make up the channel program that directs data channel operations.

**CI.** Control interval.

**CLIST.** Command list.

**component identification keyword.** The first keyword, represented as a number, in a set of keywords used to describe a DFSMSdss program failure.

**COMPRESS command.** The DFSMSdss function that reduces partitioned data sets by taking unused space and consolidating it at the end of the data set.

**concatenation.** An operation that joins two characters or strings in the order specified, forming one string whose length is equal to the sum of the lengths of the two characters or strings.

**concurrent copy.** A function to increase the accessibility of data by letting you make a consistent backup or copy of data concurrent with normal application program processing.

**concurrent copy-compatible (CC-compatible) snapshot.** See *virtual concurrent copy*.

**control area (CA).** A group of control intervals used as a unit for formatting a data set before adding records to it. Also, in a key-sequenced data set, the set of control intervals, pointed to by a sequence-set index record, that is used by VSAM for distributing freespace and for placing a sequence-set index record adjacent to its data.

**control interval (CI).** A fixed-length area of auxiliary storage space in which VSAM stores records. It is the unit of information transmitted to or from auxiliary storage by VSAM.

**control volume (CVOL).** A volume that contains one or more indexes of the catalog.

**constructs.** A collective name for data class, storage class, management class, and storage group.

**CONVERTV command.** The DFSMSdss function that converts volumes to and from Storage Management Subsystem management without data movement.

**COPY command.** The DFSMSdss function that performs data set, volume, and track movement.

**CP.** Control program.

**CREDIT.** Creation date.

**CSW.** Channel status word.

**CVAF.** Common VTOC access facility.

**CVOL.** Control volume.

**CVT.** Communication vector table.

**CYL.** A subparameter of the SPACE parameter in a DD statement. It specifies that space is allocated by cylinders.

## D

**DADSM.** The direct access space management program that maintains the VTOC, VTOCIX, and space on a volume.

**DAM.** Direct access method.

**DASD.** Direct access storage device.

**DASD ERP.** DASD error recovery procedure.

**DASD volume.** A DASD space identified by a common label and accessed by a set of related addresses.

**data class.** A list of data set allocation parameters and the values that are used when allocating a new SMS-managed data set.

**data compression (run-length).** A method of encoding repetitive series of identical characters so that they occupy less space on a dump tape. Data compression is supported by both physical dump and logical dump processing.

**Data Facility Storage Management Subsystem/MVS (DFSMS/MVS).** The complementary functions of DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm which, together with RACF provide a system-managed, administrator-controlled storage environment.

**data set backup.** Backup to protect against the loss of individual data sets.

**data set change indicator.** A bit that is set by OPEN when the data set is opened for processing other than input. This flag is supported on MVS systems that have data-set-changed flag support installed.

**DAU.** Direct access unmovable.

**DB2.** IBM DATABASE 2.

**DCB.** Data control block.

**DEFRAG command.** The DFSMSdss function that consolidates the free space on a volume to help prevent out-of-space abends on new allocations.

**DEQ.** An assembler language macro instruction used to remove control of one or more serially reusable resources from the active task.

**DFSMS.** Data Facility Storage Management Subsystem.

**DFSMS environment.** An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. See also *system-managed storage*.

**DFSMSdfp.** A DFSMS/MVS functional component that provides functions for storage management, data

management, program management, device management, and distributed data access.

**DFSMSdss.** A DFSMS/MVS functional component used to copy, move, dump, and restore data sets and volumes. DFSMSdss is the primary data mover of DFSMS/MVS.

**DFSMShsm.** A DFSMS/MVS functional component used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

**DFSMS/MVS.** Data Facility Storage Management Subsystem/MVS.

**DFSORT.** Data Facility Sort.

**DIAGNOSE.** An access method services command that scans an integrated catalog facility basic catalog structure (BCS) or a VSAM volume data set (VVDS) to validate the data structure.

**DIRF.** DADSM interrupt recording facility. If a system fails, or a permanent I/O error occurs during allocation of space or during performance of a routine that updates the VTOC, the VTOC may be in error. To ensure that an error is recorded, the DADSM routines turn on a bit in the VTOC upon entry to a DADSM function, and, if no errors occur during processing, turn off that bit upon exiting from that function.

**distribution libraries.** IBM-supplied partitioned data sets on tape containing one or more components that the user restores to disk for subsequent inclusion in a new system.

**DLIB.** Distribution library.

**DOC.** In diagnosing program failures, the keyword that identifies an error in the documentation of a program.

**DOS.** Disk Operating System.

**DOS bit.** On a volume without an indexed VTOC, a bit that indicates that the free space map is invalid.

**DOS/VSE.** DOS/Virtual Storage Extended.

**DSCB.** Data set control block.

**DSCHA.** A DFSMSdss keyword that is used in BY filtering. It indicates that the data set is to be selected if the data set has been changed.

**dsname.** Data set name.

**DSORG.** Data set organization. It is specified in the JCL as "DSORG=".

**DUMP command.** The DFSMSdss function used to back up data sets, tracks, and volumes.

**dynamic allocation.** Assignment of system resources to a program when the program is performed rather than when it is loaded main storage.

## E

**early warning system (EWS).** A microfiche copy of the information contained in the software support facility (SSF), organized by component identification number, and indexed by APAR symptom code. EWS is published monthly and available to customers of IBM licensed programs.

**ECB.** Event control block.

**EC mode.** Engineering change mode.

**empty data set.** A data set in which the pointer to the last-used block is 0.

**ENQ.** An assembler language macro instruction that requests the control program to assign control of one or more serially reusable resources to the active task. It is also used to determine the status of a resource; that is, whether it is immediately available or in use, and whether control has been previously requested for the active task in another ENQ macro instruction.

**entry-sequenced data set (ESDS).** In VSAM, a data set whose records are loaded without respect to their contents and whose RBAs cannot change. Records are retrieved and stored by addressed access, and new records are added at the end of the data set.

**EOF.** End-of-file.

**EOJ.** End of job.

**erase-on-scratch.** The physical erasure of data on a DASD data set when the data set is deleted (scratched).

**ESA.** Enterprise Systems Architecture.

**ESS.** Enterprise Storage Server.

**ESDS.** Entry-sequenced data set.

**ESTAE.** Extended specify task abnormal exit.

**EQ.** Equal to.

**EWS.** Early warning system.

**EXCP.** Execute channel program.

**execute channel program (EXCP).** A macro used to access a data set without specifying the organization.

**EXPDT.** Expiration date.

**extended sequential data set.** A sequential data set that is structured logically the same as a physical sequential data set but that is stored in a different

physical format. Extended sequential data sets consist of one or more stripes and can take advantage of the sequential data striping access technique. See also *sequential data striping* and *stripe*.

**extended specify task abnormal exit (ESTAE).** A task recovery routine that provides recovery for those programs that run enabled, unlocked, and in task mode.

**extent.** A continuous space on a DASD volume occupied by a data set or portion of a data set. An extent of a data set contains a whole number of control areas.

## F

**FC.** CVAF function code.

**FCEC.** CVAF function-error code.

**filtering.** The process of selecting data sets based on specified criteria. These criteria consist of fully- or partially-qualified data set names, or of certain data set characteristics, or of both.

**FlashCopy.** A function of the Enterprise Storage Server (ESS) and DFSMSdss that allows data to be instantly copied.

**FMID.** Function modification identifier.

**fragmentation index.** The qualitative measure of the scattered free space on a volume.

**fully-qualified data set name.** A data set in which all the qualifiers are completely spelled out.

**function modification identifier (FMID).** A code that identifies the release levels of a program product.

**FVL.** Function vector list.

## G

**GDG.** Generation data group.

**GDS.** Generation data set.

**generalized trace facility (GTF).** An optional OS/VS service program that records significant systems events, such as supervisor calls and start I/O operations, for the purpose of problem determination.

**generation data group (GDG).** A collection of historically related non-VSAM data sets that are arranged in chronological order; each data set is a generation data set.

**generation data set.** One generation of a generation data group.

**GT.** Greater than.

**GTF.** Generalized trace facility.

## H

**HFS.** Hierarchical file system.

## I

**ICKDSF.** Device Support Facilities.

**IDCAMS.** Access Method Services.

**IMS/VS.** Information Management System/Virtual Storage.

**INCORROUT.** In diagnosing program failures, the keyword that identifies incorrect or missing program output.

**incremental backup.** A process in which data sets are backed up only if they have changed since their last backup.

**installation-wide exit.** The means specifically described in an IBM software product's documentation by which an IBM software product may be modified by a customer's system programmers to change or extend the functions of the IBM software product. Such modifications consist of exit routines written to replace one or more existing modules of an IBM software product, or to add one or more modules or subroutines to an IBM software product, for the purpose of modifying (including extending) the functions of the IBM software.

**integrated catalog facility.** A facility by which VSAM data set volume-related fields are separated from the catalog and maintained in the VVDS on the volume on which the data set resides.

**integrated catalog facility catalog.** A catalog that is composed of a basic catalog structure (BCS) and its related volume table of contents (VTOC) and VSAM volume data sets (VVDSs).

**Interactive Problem Control System (IPCS).** A component of MVS that permits online problem management, interactive problem diagnosis, problem tracking, and problem reporting.

**Interactive Storage Management Facility (ISMF).** An interactive interface of DFSMS/MVS that allows users and storage administrators access to the storage management functions.

**Interactive System Productivity Facility (ISPF).** An IBM licensed program used to develop, test, and run application programs interactively. ISPF is the interactive interface for all storage management functions.

**I/O.** Input/output.

**IPCS.** Interactive Problem Control System.

**IPL.** Initial program load.

**ISMF.** Interactive Storage Management Facility.

**ISAM.** Indexed sequential access method.

**ISMF.** Interactive Storage Management Facility.

**ISPF.** Interactive Systems Productivity Facility.

**ISPF/PDF.** Interactive Systems Productivity Facility/Program Development Facility.

## J

**JCL.** Job control language.

**JES.** Job entry subsystem.

**JES2.** An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for operation, processes their output, and purges them from the system. In an installation site with more than one processor, each JES2 processor independently controls its job input, scheduling, and output processing.

**JES3.** An MVS subsystem that receives jobs into the system, converts them to internal format, selects them for operation, processes their output, and purges them from the system. In complexes that have several loosely coupled processing units, the JES3 program manages processors so that the global processor exercises centralized control over the local processors and distributes jobs to them via a common job queue.

**JFCB.** Job file control block.

**job control language (JCL).** A problem-oriented language used to identify the job or describe its requirements to an operating system.

**job entry subsystem (JES).** A system facility for spooling, job queuing, and managing I/O.

**JSCB.** Job step control block.

## K

**K.** Kilobyte: 1 024 bytes.

**key-sequenced data set.** A VSAM file or data set whose records are loaded in ascending key sequence and controlled by an index. Records are retrieved and stored by keyed access or by addressed access, and new records are inserted in key sequence by means of distributed free space. Relative byte addresses can change because of control interval or control area splits.

**keyword.** A symptom that describes one aspect of a program failure.

**KRDS.** Keyrange data set. Also known as a key-sequenced data set with key ranges.

**KSDS.** Key-sequenced data set.

## L

**LASTCC.** Last condition code.

**LDS.** Linear data set.

**like devices.** Devices that have the same track capacity and number of tracks per cylinder (for example, 3380 Model D, Model E, and Model K).

**LINK.** An assembler language macro instruction that causes control to be passed to a specified entry point. The linkage relationship established is the same as that created by a basic assembler language (BAL) instruction.

**link-pack area (LPA).** An area of virtual storage that contains reenterable routines that are loaded at IPL (initial program load) time and can be used concurrently by all tasks in the system.

**load module.** A computer program in a form suitable for loading into main storage for operation.

**load module library.** A partitioned data set used to store and retrieve load modules.

**logical (data set) DUMP operation.** A DUMP operation in which logical processing is performed.

**logical (data set) processing.** Processing that treats each data set and its associated information as a logical entity. As an example, DFSMSdss processes an entire data set before beginning with the next one.

**logical (data set) RESTORE operation.** A RESTORE operation that uses as input a data set produced by a logical DUMP operation.

**logical volume.** The output produced from a physical DUMP operation, for which all data is derived from a single DASD volume.

**LOOP.** In diagnosing program failures, the keyword that identifies a program failure in which some part of the program repeats endlessly.

**LPA.** Link-pack area.

**LT.** Less than.

**LRECL.** Logical record length.

**LVOL.** Logical volume.

## M

**Mb.** Megabit; 1 048 576 bits.

**MB.** Megabyte; 1 048 576 bytes.

**maintenance-level keyword.** In diagnosing program failures, a keyword that identifies the maintenance level of DFSMSdss.

**management class.** A list of the migration, backup, and retention parameters and the values for an SMS-managed data set.

**map record.** The record that maps the tracks dumped by DFSMSdss.

**MAXCC.** Maximum condition code.

**MCS.** Multiple console support.

**MENTITY.** Model entity.

**minivolume.** In an MVS system running on VM/370, an OS/VS-formatted VM/370 minidisk whose size is equal to or less than that of the real volume. DFSMSdss uses the device size specified in the VTOC. Minivolumes are supported only by the system version of DFSMSdss.

**MSGADRnnnt.** In diagnosing program failures, the DFSMSdss message keyword that tells of an error, or seems itself to be in error.

**MVS.** Multiple Virtual Storage.

## N

**native snapshot.** An operation that "snaps" data directly from the source location and copies it to the target location whenever the source and target data are on like devices.

**NVR.** Non-VSAM volume record.

## O

**operating system (OS).** Software that controls the execution of programs; an operating system may provide services such resource allocation, scheduling, input/output control, and data management.

**OS.** Operating system.

## P

**pageable link-pack area (PLPA).** Link-pack area.

**PAM.** Partitioned access method.

**partially qualified data set name.** A data set name in which the qualifiers are not spelled out. Asterisks and percent signs are used in place of the undefined qualifiers.

**partitioned data set (PDS).** A data set in direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

**partitioned data set extended (PDSE).** A system-managed, page-formatted data set on direct access storage. A PDSE contains an indexed directory and members similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

**PDS.** Partitioned data set.

**PDSE.** Partitioned data set extended.

**PERFM.** In diagnosing program failures, the keyword that identifies degradation in program performance.

**physical (data set) DUMP operation.** A DUMP operation in which physical processing is performed.

**physical (data set) processing.** Processing that moves data at the track-image level and can operate against volumes, tracks, and data sets. As an example, DFSMSdss may only process one volume of a multivolume data set.

**PLPA.** Pageable link-pack area.

**POU.** Partitioned organization unmovable.

**PRB.** Program request block.

**private library.** A user-owned library that is separate and distinct from the system library.

**PSU.** Physical sequential unmovable.

**PSW.** Program status word.

**PTF.** Program temporary fix.

## Q

**QSAM.** Queued sequential access method.

**qualified name.** A data set name consisting of a string of names separated by periods; for example, "TREE.FRUIT.APPLE" is a qualified name.

**qualifier.** Each component name in a qualified name other than the rightmost name. For example, "TREE" and "FRUIT" are qualifiers in "TREE.FRUIT.APPLE."

**queued sequential access method (QSAM).** An extended version of the basic sequential access method (BSAM). Input data blocks awaiting processing or output

data blocks awaiting transfer to auxiliary storage are queued on the system to minimize delays in I/O operations.

## R

**RACF.** Resource Access Control Facility.

**RAMAC Virtual Array (RVA).** A DASD that uses a virtual array architecture.

**RB.** Request block.

**RBA.** Relative byte address.

**RDJFCB.** Read job file control block.

**RECEIVE processing.** An SMP/E process necessary to install new product libraries. During this process, the code, organized as unloaded partition data sets, is loaded into temporary SMPTLIB data sets. SMP/E RECEIVE processing automatically allocates the temporary partitioned data sets that correspond to the files on the tape, and loads them from the tape.

**RECFM.** Record format.

**recovery.** The process of rebuilding data after it has been damaged or destroyed, often by restoring a backup version of the data or by reapplying transactions recorded in a log.

**REFDT.** A DFSMSdss keyword used in BY filtering. It indicates the last-referenced date.

**relative byte address (RBA).** The displacement (expressed as a fullword binary integer) of a data record or a control interval from the beginning of the data set to which it belongs, independent of the manner in which the data set is stored.

**relative record data set (RRDS).** A VSAM data set whose records are loaded into fixed-length slots.

**RELEASE command.** The DFSMSdss function that releases the unused space in sequential and partitioned data sets for use by other data sets.

**RESERVE.** A method of serializing DADSM update accesses to the VTOC. It is also a method of serializing processor accesses to a shared DASD volume.

**Resource Access Control Facility (RACF).** An IBM program product that provides for access control by identifying and verifying users to the system, authorizing access to DASD data sets, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected data sets.

**RESTORE command.** The DFSMSdss function used to recover data sets, tracks, and volumes.

**RMID.** Replacement module identifier.

**RRDS.** Relative record data set.

**RVA.** RAMAC Virtual Array.

**run-length data compression.** Data compression (run-length).

## S

**SAF.** System authorization facility.

**SAM.** Sequential access method.

**scheduler task.** A DFSMSdss subtask that interprets and schedules commands.

**SCP.** System control program.

**SEQ.** Sequential or sequential processing.

**sequential data striping.** A software implementation of a disk array that distributes data sets across multiple volumes to improve performance.

**SEREP.** System environmental recording, editing, and printing

**SMF.** System management facilities.

**SML.** MVS Storage Management Library.

**SMP.** System Modification Program.

**SMP/E.** System Modification Program/Extended.

**SMPE.** A cataloged procedure that includes the required DD statements for running SMP/E and is used in the RECEIVE, APPLY, and ACCEPT steps of SMP/E processing.

**SMS.** Storage Management Subsystem.

**snapshot.** A function of the RAMAC Virtual Array (RVA) that allows an instantaneous copy to be made of data sets using DFSMS software.

**software support facility (SSF).** An IBM online data base that allows for storage and retrieval of information about all current APARs and PTFs.

**SP.** System Product.

**sphere.** A VSAM cluster with one or more associated alternate indexes and paths. The VSAM cluster (sometimes called the base cluster), alternate indexes, and paths are sometimes referred to as sphere components.

**SSF.** Software support facility.

**standalone restore.** One of two DFSMSdss programs. The standalone restore program runs independently of

the MVS system environment and is limited to one function—a full or partial (tracks) RESTORE from a dump tape.

**storage class.** A named list of data set storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

**storage constructs.** The group of predefined models (data class, management class, storage class, and storage group) that are used to classify storage management needs and procedures for data sets under the Storage Management Subsystem. Each data set has construct names associated with it by explicit specification or defaulting.

**storage group.** A named collection of DASD volumes that have been grouped to meet a defined service strategy.

**storage management.** The task of managing auxiliary storage resources for an installation.

**Storage Management Subsystem (SMS).** An MVS subsystem that helps automate and centralize the management of storage. To manage storage, the storage management subsystem provides the storage administrator with control over data class, storage class, management class, storage group, and automatic class selection routine definitions.

**stripe.** The portion of a striped data set (for example, an extended sequential data set) that resides on one disk. The records in that portion are not necessarily logically consecutive. The system distributes records among the stripes such that the disks can be read or written simultaneously to gain better performance.

**subtask.** A task initiated and ended by a higher order task.

**SVC.** Supervisor call instruction.

**SVRB.** Supervisor request block.

**SYSRES.** System residence disk

**system data.** The data sets required by MVS or its subsystems for initialization.

**system-managed data set.** A data set that has been assigned a storage class.

**system-managed storage.** Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, space, and security to applications.

**system library.** A collection of data sets or files in which the parts of an operating system are stored.

**system link library.** System library.

**System Modification Program (SMP).** A program used to install software and software changes on the MVS system.

**System Modification Program Extended (SMP/E).** An IBM licensed program used to install software and software changes on the MVS system. In addition to providing the services of SMP, SMP/E consolidates installation data, allows more flexibility in selecting changes to be installed, provides a dialog interface, and supports dynamic allocation of data sets.

## T

**TCB.** Task control block.

**Time sharing option (TSO).** An option on the operating system for a System/370 that provides interactive time sharing from remote terminals.

**TIOT.** Task input/output table.

**TLIB.** Target library.

**track packing.** A technique used by DFSMSdss that builds target tracks for any DASD device using input physical record information.

**TRK.** A subparameter of the SPACE parameter in a DD statement. It specifies that space is to be allocated by tracks.

**TSO.** Time sharing option.

**TSO/E.** TSO/Extensions.

**TTR.** Track-track-record.

**type-of-failure keyword.** In diagnosing program failures, a keyword that identifies the type of program failure that has occurred in DFSMSdss.

## U

**UACC.** Universal access authority.

**UCB.** Unit control block.

**UIM.** User interaction module.

**unlike devices.** Devices that have different track capacities or a different number of tracks per cylinder.

**used tracks.** Tracks from the beginning of data sets to the last-used track.

**user exit.** A programming service provided by an IBM software product that may be requested by an application program for the service of transferring control back to the application program upon the later occurrence of a user-specified event.

## V

**VDRL.** Volume restore limits.

**VDSS.** VTOC/Data Set Services.

**virtual concurrent copy.** An operation that uses snapshot to provide a concurrent copy-like function when the source volume supports snapshot, but not concurrent copy. (Also called CC-compatible snapshot.)

**virtual storage access method (VSAM).** An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by the relative-record number.

**VM.** Virtual machine.

**VOLID.** Volume ID.

**VOLSER.** Volume serial number.

**volume.** The storage space on DASD, tape or optical devices, which is identified by a volume label.

**volume backup.** Backup of an entire volume to protect against the loss of the volume.

**volume header record.** The record in the DFSMSdss dump tape that identifies and contains data pertinent to the whole volume, and identifies the type of operation that created a dump.

**volume trailer record.** The record in the DFSMSdss dump tape that identifies the end of the data for a DASD volume.

**VRRDS.** A VSAM variable record RRDS.

**VSAM.** Virtual storage access method.

**VSAM volume data set (VVDS).** A data set that describes the VSAM and SMS-managed non-VSAM data sets on a volume. The name of the data set is SYS1.VVDS.Vvolser.

**VSE.** Virtual storage extended.

**VTOC.** Volume table of contents.

**VTOCIX.** The data set on which the location of the data set VTOC entries are kept in an index for quick access by DADSM. The name of the data set is SYS1.VTOCIX.Vvolser.

**VVDS.** VSAM volume data set.

**VVR.** VSAM volume record.

## **W**

**WAIT.** In diagnosing program failures, the keyword that identifies DFSMSDss suspended activity, while waiting for some condition to be satisfied. DFSMSDss does not issue a message to tell why it is waiting.

**WTO.** Write to operator.

## **X**

**XA.** Extended Architecture.

---

# Index

## Special Characters

- \*\* (double asterisk) used in partially qualified data set names 19
- % (percent sign) used in partially qualified data set names 18
- \* (single asterisk) used in partially qualified data set names 18

## Numerics

- 16 MB virtual storage, storage requirements per command 9
- 16 megabytes virtual storage
  - storage requirements per command 10
- 3380 Direct Access Storage 11
- 3390 Direct Access Storage 11
- 9345 DASD module 11

## A

- Access Method Services 74
- ACS information 115
- ACS variables
  - CONVERTV 116
  - COPY 115
    - name/description 115
    - passed in COPY command 115
    - passed in RESTORE/CONVERTV 116
  - RESTORE 116
- ACTIVE state 64, 84
- aliases of non-VSAM data sets 39, 55, 79
- ALLDATA keyword 48
  - dumping records past last-used-block pointer 40
  - to control what DFSMSdss copies 70
  - with preallocated target 89
- ALLEXCP keyword 48
  - dumping records past last-used-block pointer 40
  - to control what DFSMSdss copies 70
  - with preallocated target 89
- ALLOC keyword 19
- ALTER LOCK, IDCAMS command 54
- ANTMAIN data sets 46
- application interface
  - function 23
  - invoking 23
  - module names 26
- archive 4, 29, 32
- auditing information 23
- authorization checking for EXPORT/IMPORT (IDCAMS commands) 74
- authorization installation exit routine 3
- authorization structure 25
- AUTORELBLOCKADDRESS keyword 56, 84
- availability 29
- availability management 3
- availability strategy, planning 29

## B

- backing up HFS data sets 37

- backup
  - concurrent copy 4, 35, 75
  - data set 3, 29, 33, 37
  - disaster recovery 4, 17
  - incremental 3
  - integrated catalog facility user catalog 39
  - scenario 36
  - SMS-managed data sets 41
  - SMS-managed environment 30
  - special requirements 37
  - system volumes 42
  - vital records 4, 16, 29
  - volumes 3, 29, 30, 42
- block size (DFSMSdss dump data set)
  - default when dumping to tape or DASD 43
  - minimum 43
- BY criteria 18
- BY keyword
  - criteria 19
  - operator meaning 20

## C

- card readers supported 11
- catalog 52
  - during logical restore 50
  - integrated catalog facility 54
  - locking 54
  - moving 78
  - recovery 54
  - temporary copied 13
- CATALOG keyword
  - during logical restore processing 50
  - with preallocated target 89
- cataloging non-VSAM data sets during restore 52
- CATLG keyword 19
- changing
  - management class with restore 62
  - storage class with RESTORE 61
- characteristics of data sets 18
- class names
  - filter 41
  - saved 41
- cluster, VSAM, restoring 58
- combining data set extents 107
- compaction 29
- COMPRESS command 18
  - definition 8
  - module name 26
  - PDS (partitioned data set) 106
- concurrent copy 4, 6, 70
  - backup 35, 75
  - performance considerations 44
- consoles supported 11
- CONSOLIDATE option 108, 110
- controlling DFSMSdss
  - using ISMF 23
  - using JCL 23

- controlling what DFSMSdss copies 70
- conversion
  - by data movement 7
  - from SMS
    - by data movement 97
    - special requirements 102
    - without data movement 102
  - GDG data set 102, 103
  - in an SMS-managed environment 60
  - ineligible data sets 95, 96
  - multivolume 101, 102
  - preparing a volume 98
  - simulating 97
  - to and from SMS management 7, 95
  - to SMS
    - by data movement 96
    - special requirements 100
    - without data movement 99
    - without data movement 7, 97
- CONVERTV command 7
- CONVERTV operation, variables passed to ACS routines 116
- COPY command 18
  - changing
    - management class 87
    - storage class 87
  - data mover selection matrix 74
  - indexed sequential data sets 83
  - logical volume 90
  - module name 26
  - moving data 6
  - physical volume 91
  - sphere restrictions 81
  - user catalog 78
  - variables passed to ACS routines 115
  - VSAM data sets 80
- COPY DATASET, data mover selection matrix 74
- copying multivolume data sets 80
- COPYVOLID keyword 67
- CPVOLUME keyword 42, 67, 94
- CREDIT keyword 19
- criteria for filtering 18
- critical data sets 31
- customer program, invocation 36, 47
- D**
- damaged PDS, restoring 59
- DASD (direct access storage device)
  - devices supported 11
  - initialized 11
  - reclaiming space 105, 106
  - space fragmentation 108
  - space utilization 108
- data class 1
- data compaction
  - hardware exception 43
  - software exception 43
- data integrity 17
  - during data processing 17
  - shared DASD considerations 47
- data movement
  - conversion by 7
- data movement (*continued*)
  - criteria for 69
  - preparing for 69
  - with FlashCopy 92
  - with snapshot 92
- data mover selection matrix 74
- data mover selection matrix for Data Set Copy 74
- data security 113
- data set
  - absolute track 55
  - backup 3, 29, 33, 37
  - broken 17
  - characteristics (BY criteria) 18
  - converting to multivolume in an SMS-managed environment 60
  - critical 31
  - DEFRAG 12
  - direct access 12, 55, 56, 83
  - dummy 14
  - erase table for DEFRAG 113
  - expiration date handling 73
  - extents, combining 107
  - filtering 15, 18, 19
  - GDG 63, 84
  - HFS 37
  - indexed sequential 12, 55, 83
  - last-used-block pointer 40
  - line operator module names 26
  - logical dump 31, 34
  - logical restore 48
  - message 12
  - moving 70
  - multivolume 37, 79
  - name
    - fully qualified 18
    - partially qualified 18
  - non-SMS-managed 64
  - organization 12
  - partitioned (PDS) 12
  - physical Dump 34
  - physical restore 51
  - preallocated 49, 88
  - restoring 48
    - in an SMS-managed environment 59
    - multivolume 53
    - to multiple target volumes 53
  - sequential 12
  - SMS-managed 41, 60, 63
  - special requirements 37, 77
  - SYS1 system 40
  - system 78
  - temporary 12, 13, 102
  - uncataloged 21
  - undefined DSORG 57
  - unmovable 55, 56, 82
  - VSAM 40
    - with phantom catalog entries 64
- DATABASE 2 12
- DATACLAS keyword 19
- DATASET keyword
  - data mover selection matrix for copy 74

DATASET keyword *(continued)*  
 with logical processing 15  
 with physical data set restore 51  
 DB2 (DATABASE 2) data sets 12  
 default block size when dumping to tape or DASD 43  
 DEFERRED state 64, 84  
 DEFRAG command 18  
 data set erase table 113  
 data sets excluded 109  
 definition 8  
 general hints 110  
 operation interrupted 13  
 options 110  
 serialization 111  
 when to run 109  
 with snapshot 109  
 DEFRAG data set 12  
 DEFRAG/RACF database 110  
 DELETEDCATALOGENTRY keyword 64  
 deleting unwanted data sets 106  
 Device Support Facilities utility 11  
 devices supported 11, 66  
 DFSMSdss  
 control 23  
 devices supported 11, 66  
 filtering 18  
 function protection 25  
 interactive 23  
 invoking 23  
 invoking with application interface 23  
 line operators 26  
 module names 26  
 overview 1  
 protecting modules 26  
 storage requirements 9  
 system requirements 9  
 volume formats supported 11  
 direct access  
 data set  
 moving 83  
 restoring 56  
 supported 12  
 storage device 11  
 disaster recovery 4, 17, 29, 30, 64  
 disk drives 11  
 DSCHA keyword 19  
 DSORG keyword 19  
 dummy data set 14  
 DUMP command 4, 8, 18, 44  
 disaster recovery of logical data sets 31  
 exceptions to  
 hardware data compaction 43  
 software data compaction 43  
 logical data set 34  
 logical data set disaster recovery 31  
 logical volume 42  
 module name 26  
 non-VSAM data sets 39  
 OPTIMIZE keyword 44  
 performance considerations 44  
 physical data set 34

DUMP command 4, 8, 18, 44 *(continued)*  
 physical volume 42  
 printed output  
 produced by integrated catalog facility user  
 catalog 39  
 VALIDATE keyword 40  
 dumping  
 data sets 40  
 efficiently 43  
 HFS data sets 37  
 indexed VSAM data sets 40  
 integrated catalog facility user catalog 39  
 multivolume data sets 37  
 non-VSAM data sets 39  
 SYS1 system data sets 40  
 VSAM spheres 40  
 DYNALLOC option 110

## E

eligibility for conversion 96  
 empty non-VSAM data sets 77  
 enqueue installation exit routine 3  
 environment, system 9  
 EQ operator 20  
 ESDS data sets, supported 12  
 EXCLUDE criteria 18  
 EXCP data sets, supported 12  
 exit functions, User Interaction Module 24  
 exits, installation 3  
 EXPDT keyword 19  
 expiration date handling 73  
 EXPORT/IMPORT (IDCAMS commands), authorization  
 checking 74  
 extended-addressable VSAM KSDS 12  
 extended-sequential data set  
 processing 2  
 support for 2  
 extents, combining 107  
 EXTNT keyword 19

## F

filter, class names 41  
 filtering 15  
 COMPRESS 18  
 COPY 18  
 data set characteristics 19  
 examples 20  
 FILTERDD keyword 20  
 general description 18  
 logical DUMP 18  
 logical RESTORE 18  
 physical DUMP 18  
 physical RESTORE 18  
 RELEASE 18  
 RESTORE command 48  
 FlashCopy  
 moving data with FlashCopy 7  
 moving volumes with FlashCopy 92  
 FORCE keyword 55, 83, 84  
 FRAGI keyword 110

fragmentation index 110  
free-space fragmentation 108  
FSIZE keyword 19  
fully qualified data set names 18

## G

GDG (generation data group) data set  
    conversion from SMS 103  
    conversion to SMS 102  
    moving 84  
    restoring 63, 64  
GE operator 20  
generation data group 63  
generation data sets, moving to non-SMS-managed  
    volumes 85  
GT operator 20

## H

HFS (hierarchical file system) data set 12  
HFS data set, dumping 37

## I

ICKDSF, initialize DASD volumes with 11  
IDCAMS utility 74  
IEBCOPY utility 74  
IEBISAM utility 74  
IGWFAMS utility 74  
IMPORT keyword 65  
INCLUDE criteria 18  
incremental backup 3, 33  
index, fragmentation 110  
indexed sequential data set  
    copying 83  
    moving 83  
    restoring 55  
    supported 12  
indexed VSAM data sets, logical data set dump of 40  
indexed VTOCs 11  
INITIAL state 98  
INITIAL status 63  
initialize all DASD volumes 11  
input volumes, selecting 70  
installation exit routines 3  
integrated catalog facility user catalog  
    backing up a user catalog, example of 39  
    dumping 39  
    printed output for a dumping a user catalog 39  
    printed output for restoring a user catalog 54  
    restore 54  
Interactive Storage Management Facility 23  
invocation  
    customer program 36  
invocation, from a customer program 47  
invoking DFSMSdss  
    application interface 23  
    using ISMF 23  
    using JCL 23  
    via an application program 23

invoking ISMF 23  
ISMF (Interactive Storage Management Facility)  
    display panels 23  
    function protection 25  
    invoking 23  
    line operators 26  
    logging on 23  
    menu-driven panels 23  
    module names 26  
    MVS/EA 23  
    online panels 23  
    protecting modules 26  
    use and examples 23  
    volume list 69

## J

JCL (job control language)  
    invoking DFSMSdss 23  
    restore integrated catalog facility user catalog 54  
job control language 23  
JOB CAT DD statement, JCL 52

## K

key range data sets, supported 12  
keyword  
    module protection 27  
    preallocated targets 89  
    profile names 27  
KSDS data sets, supported 12

## L

LDS data sets, supported 12  
LE operator 20  
like devices, moving volumes to 93  
line operators, DFSMSdss/ISMF 26  
location-dependent data 49, 55  
locking a user catalog 54  
logging on to ISMF 23  
logical copy 69  
logical COPY 90  
logical data set dump of indexed VSAM data sets 40  
logical data set restore 48  
logical dump volume 42  
logical processing 15  
logical restore 59  
    cataloging data sets 50  
    data sets with phantom catalog entries 64  
    renaming data sets 51  
    user catalog aliases 54  
    without preallocated targets 55  
LOGICALVOLUME keyword 52  
LT operator 20

## M

magnetic tape devices supported 11  
management class 2  
    changing with copy 87

- management class 2 *(continued)*
  - changing with RESTORE 62
- maximize track utilization by reblocking 114
- MAXMOVE keyword 110
- MAXSIZE variable 118
- message data set 12
- MGMTCLAS keyword 19, 60
- minivolumes 11
- MINSECQTY keyword 105
- MINTRACKSUNUSED keyword 105
- module names for data set application commands 26
- moving
  - catalogs 78
  - damaged PDS 82
  - data in an SMS-managed environment 6
  - data sets 70
    - special requirements 77
    - utilities 74
  - data sets to unlike devices 83
  - direct access data set 83
  - empty non-VSAM data sets 77
  - generation data sets
    - to non-SMS-managed volumes 85
    - to SMS-managed volumes 84
  - indexed sequential data set 83
  - multivolume data set 79
  - non-VSAM data sets with aliases 79
  - PDSE 82
  - SMS-managed data sets 85
  - system data sets 78
    - to preallocated data set 88
  - undefined DSORG data sets 77
  - unmovable data sets 82
  - volumes
    - logical volumes 90
    - physical volumes 91
      - to devices of equal capacity 93
      - to devices of greater capacity 93
      - to unlike devices 93
    - VM-format volumes 94
    - VTOC considerations 92
- moving data
  - with concurrent copy 6
  - with FlashCopy 7
  - with snapshot 7
- moving volumes
  - with FlashCopy 92
  - with snapshot 92
- MULTI keyword 19
- multiple target volumes
  - restoring 53
  - specifying 89
- multivolume data set 79
  - conversion 101, 102
  - copying 80
  - dumping 37
  - restore 53
  - restoring 80
- multivolume VSAM data sets 60, 80
- MVS environments supported 9

## N

- names, data set 18
- native snapshot 76, 77
- NE operator 20
- non-SMS-managed data set, restoring 64
- non-SMS-managed target, special considerations 103
- non-SMS-managed volumes, moving generation data sets to 85
- non-VSAM data sets with aliases 39, 55, 79
- non-VSAM preallocation 88
- nonindexed VTOC 11
- NOPACKING keyword
  - restoring to preallocated target data sets 50
  - to restore damaged partitioned data sets 59
  - with preallocated target 89
- NULLMGMTCLAS keyword 60
- NULLSTORCLAS keyword 60

## O

- operating environment 9
- OPTIMIZE keyword 44
- options installation exit routine 3
- organization, data set 12
- OUTDDNAME keyword 49, 71
- OUTDYNNAME keyword 49, 71
- output volume
  - selecting 71
  - specifying 49, 52, 66
- overview of DFSMSdss 1

## P

- PARALLEL command 9
- PARALLEL keyword 44
- partially qualified data set names 18
- partitioned data set 11
- partitioned data set extended 11
- PASSDELAY option 110
- PDS (partitioned data set) 12
  - abnormal conditions 59
  - compressing 106
  - monitoring PDSs during 82
  - moving damaged 82
  - prevention of 82
  - repairs by DFSMSdss during 82
  - restoring, damaged 59
  - storage requirements 11
  - supported 12
  - translation during 82
- PDSE (partitioned data set extended) 12
  - moving 82
  - restoring 58
- PERCENTUTILIZED keyword
  - with a logical data set restore operation 49
  - with preallocated target 89
- performance
  - concurrent copy 44
  - DEFRAG command 109
  - DUMP 44
  - read I/O pacing 47

- phantom catalog entries 64
- physical copy 69, 91
- physical processing 15, 16
- physical restore 51, 54, 59
  - INITIAL status 63
  - output volume selection 52
- physical restore of SMS-managed data sets 63
- physical volume dump 42
- planning an availability strategy 29
- preallocated data set
  - COPY command 88
  - moving to 88
  - restoring to 49
- preallocated targets
  - restoring 56
  - restoring without 55
- preallocation
  - non-VSAM 88
  - VSAM 88
- PREPARE keyword 7, 98
- printers supported 11
- PROCESS keyword 77, 88
- processing
  - logical 15, 16
  - physical 15, 16, 17
- protection
  - DFSMSdss function 25
  - DFSMSdss/ISMF modules 26
  - functions with RACF 25
  - ISMF access 25
  - keywords with RACF 27

## R

- RACF (Resource Access Control Facility)
  - data set erase table for DEFRAG 113
  - database 110
  - keyword profiles 27
  - protecting keyword modules 27
  - protecting keywords 27
- RAMAC Virtual Array (RVA) 7, 36, 46, 76
- read I/O pacing, performance considerations 47
- readers, card, supported 11
- reblock installation exit routine 3
- REBLOCK keyword, with preallocated target 89
- REBLOCK processing
  - determining block size 114
  - ineligible data sets 114
  - track usage 114
- RECATALOG keyword
  - during logical restore processing 50
  - with preallocated target 89
- reclaiming DASD space 105
- record counting for copy, dump, and restore 3
- records past last-used-block pointer, dumping 40
- recovery
  - disaster 30
  - SMS-managed environment 30
  - system volumes 67
  - user catalog 54
- REDETERMINE keyword 99

- REFDT keyword 19
- RELBLOCKADDRESS keyword 57, 84
- RELEASE command 18
  - definition 8
  - module name 26
  - unused space in data sets 105
- remote site 30
- RENAME keyword, during logical restore processing 51
- RENAMEUNCONDITIONAL keyword
  - with COPY command 72
  - with preallocated target 89
- renaming data sets 72
- REPRO (IDCAMS command) 74
- Resource Access Control Facility 27
- RESTORE command 4, 8, 18, 48, 49
  - absolute track data set 55
  - changing management class 62
  - changing storage class 61
  - data sets with phantom catalog entries 64
  - direct data sets 55
  - filtering 48
  - indexed sequential data sets 55
  - logical 48
  - module name 26
  - non-VSAM data sets 55
  - physical 48, 51
  - preallocated target data sets 49
  - printed output
    - produced by integrated catalog facility user catalog 54
    - unmovable data sets 55
    - variables passed to ACS routines 116
- restoring 49
  - damaged PDS 59
  - data sets 48
  - direct access data sets 56
  - GDG data sets 63
  - in an SMS-managed environment 59
  - multivolume data sets 80
  - non-SMS-managed data set 64
  - PDSE 58
  - preallocated targets 56
  - SMS-managed data sets 60, 63
  - SMS-managed GDG data sets 64
  - undefined DSORG data set 57
  - volumes 64
  - VSAM cluster 58
  - VSAM sphere 57
  - without preallocated targets 55
- ROLLED-OFF state 64, 84
- RRDS data sets, supported 12
- RVA (RAMAC Virtual Array) 7, 36, 46, 76

## S

- SAM compressed data set, copying 81
- SDM (system data mover) 46
- SELECTMULTI keyword
  - with backup function 37
  - with conversion function 101, 103
  - with COPY DATASET function 70

- SELECTMULTI keyword (*continued*)
  - with COPY function 79
- sequential data sets, supported 12
- sequential data striping, support for 2
- serialization 111
- SHARE keyword, with backup function for HFS 37
- simulating conversion 97
- SIZE variable 118
- SMS conversion
  - eligibility 96
  - ineligibility 95, 96
- SMS-managed data sets 41
  - backup 41
  - moving 85
  - physical RESTORE 63
  - restoring 60
  - restoring, GDG data sets 64
- SMS-managed environment
  - backup 30
  - converting data sets to multivolume 60
  - moving data 6
  - Non-SMS-managed data 30
  - recovery 30
  - restoring data sets 59
  - SMS-managed data 30
- SMS-managed volumes, moving data sets to 84
- SMS management
  - conversion by data movement 96
  - conversion from, by data movement 97
  - conversion to and from 7
  - conversion without data movement 99
  - converting from, without data movement 102
  - converting to and from 95
- SMS Report 100
- snapshot 6
  - moving data sets with snapshot 76
  - moving volumes with snapshot 92
  - native 76
  - native mode 7, 76, 77
  - virtual concurrent copy 36
  - virtual concurrent copy mode 76, 77
- space
  - considerations 43
  - fragmentation on DASD 108
  - management 8
- special considerations for non-SMS-managed target 103
- special requirements 90
  - conversion from SMS 102
  - conversion to SMS 100
  - data set backup 37
  - moving data sets 77
  - restoring data sets 53
- specified operating environment 9
- sphere eligibility 100
- SPHERE keyword 40, 57, 81
- sphere processing, VSAM eligibility 100
- spheres
  - dumping VSAM 40
  - restoring VSAM 57

- Stand-Alone restore
  - overview 1
  - physical processing 17
  - system volumes, backing up 17
  - system volumes, recovering 67
  - VM, running Stand-Alone restore under 67
  - volume dump, physical 42
- statistical information 23
- STEPCAT DD statement, JCL 52
- storage class 1
  - changing with COPY 87
  - changing with RESTORE 61
- storage group 2
- Storage Management Subsystem, converting to and from 95
- storage requirements 9
  - PDS (partitioned data set) 11
  - per command
    - above 16MB 10
    - below 16MB 9
    - VSAM data set 11
- STORCLAS keyword 20, 60
- striped data sets, supported 12
- supported devices 11, 66
- SYS1.ANTMAIN.SNAPnnnn data sets 46
- SYSALLDA 70
- SYSDA 70
- system consoles supported 11
- system data mover (SDM) 46
- system data set
  - dumping 40
  - moving 78
- system environment 9
- system requirements 9
- system volumes
  - backing up 42
  - recovering 67

## T

- tape devices supported 11
- target volume 6, 86
- temporary copied catalogs 13
- temporary data set 12, 13, 102
- temporary work space 70
- TEST keyword 98
- TOL(ENQF) keyword, with backup function for HFS 37
- track utilization, maximize by reblocking 114
- TTRADDRESS keyword 57, 84

## U

- uncataloged data set 21
- undefined DSORG data sets
  - moving 77
  - restoring 57
- universal access authority (UACC) 25
- unlike devices 91
  - moving volumes 93
- unmovable data set
  - moving 82
  - restoring to preallocated targets 56

- unused space, releasing 105
- unwanted data sets, deleting 106
- user catalog, moving 78
- User Interaction Module exit functions 24
- using SIZE and MAXSIZE variables 118
- utilities
  - in data set copy operation 74
  - moving data sets 74

## V

- VALIDATE keyword 3, 40
- virtual concurrent copy mode 6
- virtual concurrent copy working space 46
- virtual input/output (VIO) device supported 11
- virtual storage access method 40
- vital records 4, 16, 29, 32
- VM-format volumes
  - backing up 42
  - moving 94
  - recovering 67
  - with DFSMSdss 11
- VM minivolumes 36
- volume
  - backup 3, 29, 30, 42
  - broken 17
  - conversion without data movement 7
  - formats supported by DFSMSdss 11
  - input, selecting 70
  - line operator module names 26
  - logical copy operation 90
  - logical dump 42
  - multiple target 89
  - multiple target, restore 53
  - output, selecting 71
  - output, selection 49, 52
  - output, specifying 66
  - physical copy 91
  - physical dump 42
  - preparing for conversion 98
  - selecting target 86
  - system, recovering 67
  - VM-formatted 11
- volume list, ISMF 69
- VSAM (virtual storage access method)
  - cluster, restoring 58
  - data set
    - converting 60, 80
    - copying 80
    - dumping 40
    - storage requirements 11
  - preallocation 88
  - sphere
    - dumping 40
    - eligibility 100
    - restoring 57
- VTOC
  - indexed 11
  - nonindexed 11

## W

- WAIT keyword 110
- when to use
  - logical processing 16
  - physical processing 17
- work space 70
- working space data set 46

---

# Readers' Comments — We'd Like to Hear from You

OS/390  
DFSMSdss Storage Administration  
Guide  
Release 10

Publication No. SC35-0393-00

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	<input type="checkbox"/>				

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	<input type="checkbox"/>				
Complete	<input type="checkbox"/>				
Easy to find	<input type="checkbox"/>				
Easy to understand	<input type="checkbox"/>				
Well organized	<input type="checkbox"/>				
Applicable to your tasks	<input type="checkbox"/>				

Please tell us how we can improve this book:

Thank you for your responses. May we contact you?  Yes  No

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

---

Name

---

Address

---

Company or Organization

---

Phone No.



Fold and Tape

Please do not staple

Fold and Tape



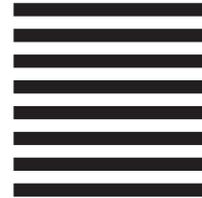
NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES

# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

International Business Machines Corporation  
Information Development  
Department 61C  
9000 South Rita Road  
TUCSON AZ 85775-4401



Fold and Tape

Please do not staple

Fold and Tape





Program Number: 5647-A01



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

SC35-0393-00

