DFSMS/MVS Version 1 Release 5

Planning for Installation

SC26-4919-06

DFSMS/MVS Version 1 Release 5

Planning for Installation

SC26-4919-06

Note!

Before using this information and the product it supports, be sure to read the “Notices” on page xiii.

Seventh Edition (March 1999)

This edition applies to Version 1 Release 5 of DFSMS/MVS (5695-DF1), Release 7 of OS/390 (5647-A01), and any subsequent releases until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

A form for readers' comments appears at the back of this publication. If the form has been removed, address your comments to:

International Business Machines Corporation
RCF Processing Department
G26/050
5600 Cottle Road
SAN JOSE, CA 95193-0001
U.S.A.

Or you can send your comments electronically to starpubs@vnet.ibm.com.

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1979, 1999. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

Notices . . . xiii
Programming Interface Information . . . xiii
Trademarks . . . xiv

About This Book . . . xv
How to Tell if this Book is Current . . . xvi
References to Product Names Used in DFSMS/MVS Publications . . . xvi

Summary of Changes . . . xix
Seventh Edition, March 1999 . . . xix
Sixth Edition, March 1998 . . . xix
Fifth Edition, June 1997 . . . xx
Fourth Edition, September 1996 . . . xx
Third Edition, October 1995 . . . xxi
Second Edition, March 1994 . . . xxi

Planning to Install DFSMS/MVS . . . 1

The DFSMS/MVS Environment—an Overview . . . 3
DFSMS/MVS in the MVS/ESA or OS/390 Operating System . . . 3
The DFSMS System-Managed Storage Environment . . . 3
DFSMS/MVS Functional Components . . . 3
DFSMSdfp . . . 4
DFSMSdss . . . 4
DFSMShsm . . . 5
DFSMSrmm . . . 5
Additional DFSMS/MVS Features and Supporting Products . . . 6
DFSMS Optimizer . . . 6
Network File System . . . 7
DFSORT . . . 7

DFSMS/MVS Packaging, Enablement, and Support . . . 9
Product Offerings . . . 9
Distribution Package . . . 9
Publications . . . 10
Licensing and Enablement . . . 10
Service Agreements . . . 11

DFSMS/MVS Installation Requirements . . . 13
Hardware Requirements . . . 13
Software Requirements . . . 13
Program Products Used for Installation . . . 13
Corequisite System Products . . . 14
Products Recommended for DFSMS/MVS Full-Function Support . . . 14
Storage Requirements . . . 15
SMP/E Data Sets . . . 15
New System Data Sets . . . 16
Device Support Considerations . . . 16
I/O Devices . . . 16

© Copyright IBM Corp. 1979, 1999

Device Initialization 16
Operating Modes 17

Planning for Migration and Coexistence 21

Year 2000 Support for DFSMS/MVS 25
Summary of Year 2000 Changes 25
Migration Actions 27
Special Actions During Testing 27
Migration Assistance 28

Considerations for Multisystem Environments 29
Shared Resources in a Storage Management Subsystem Complex 29
Shared Catalogs 30
System-Managed and Non-System-Managed Data 31
Partitioned Data Sets Extended 31
VSAM Data Sets 31
Checkpoint Data Sets 32
Tape Libraries 32
Program Management 32
Serialization for DFSMShsm Functions 32
Mixed Releases of DFDSS or DFHSM 32
VTOC Compatibility Between MVS/ESA and MVS/XA Systems 33

Migration Planning for DFSMS/MVS Version 1 Release 5 35
Catalog Alias and Usability Improvements 36
Catalog Sharing Enhancements for Sysplex Environments 38
Control Data Set Structure Changes for SMS 39
DEB Table Expansion 40
DFSMSdss Logical Release Support 41
DFSMSdss SnapShot Support 41
DFSMSdss Storage Group Filtering 42
DFSMShsm Secondary Host Promotion 43
DFSMShsm Control Data Set Extended Addressability 44
DFSMShsm Overhead Reduction 45
DFSMShsm GRSplex Serialization 46
DFSMShsm General Function and Usability Improvements 47
DFSMSrmm Application Programming Interface 49
DFSMSrmm Catalog Status Tracking 50
DFSMSrmm Disposition Control 51
DFSMSrmm Subsystem Enhancements 52
DFSMSrmm TSO Subcommand Enhancements 53
DFSMSrmm Vital Record Specification Enhancements 53
Extended Remote Copy Enhancements 55
Hierarchical File System Performance and Other Enhancements 55
ISO/ANSI Version 4 Tape Support 57
NaviQuest Batch Support 60
OAM Parallel Sysplex Support 61
OAM Pseudo Library Concept 62
OAM Support for the 3995-SW4 Optical Drive 64
OAM New and Changed Operator Commands for Object Support 64
Reformatting a 3995 Optical Disk 64
Relabeling a 3995 Optical Disk Volume 65

Displaying OAM XCF Status . . . 65
Displaying SETOPT and SETOAM Parameters . . . 65
Using the Update Command to Set SETOAM and SETOPT Values . . . 66
Virtual Tape Server Import/Export Support . . . 67
Exporting Volumes . . . 68
Importing Volumes . . . 69
VSAM Extended Format Data Set Enhancements . . . 72

Migration Planning for DFSMS/MVS Version 1 Release 4 . . . 77
DFM/MVS DataAgent . . . 78
DFSMSdss Enhanced Protection of Checkpointed Data Sets . . . 79
DFSMSdss Multivolume Selection . . . 80
DFSMSdss Stand-Alone Services . . . 80
DFSMShsm ABARS Activity Log Deletion . . . 81
DFSMShsm ABARS Externalization of the TGTGDS and OPTIMIZE Parameters . . . 82
DFSMShsm ABARS File Stacking . . . 83
DFSMShsm ABARS Generation Data Group Base Name in Allocate . . . 84
DFSMShsm ABARS 64 Tasks . . . 85
DFSMShsm CDS RLS Serialization . . . 85
DFSMShsm CPU Time in WWFSR and ABR Records . . . 87
DFSMShsm Dump Analysis Elimination . . . 88
DFSMShsm Duplex Tape Function . . . 89
DFSMShsm Enhanced Protection of Checkpointed Data Sets . . . 91
DFSMShsm Extended Invocation of the ARCBEEXT Exit . . . 93
DFSMShsm General Function and Usability Improvements . . . 93
DFSMSrmm Enhancements . . . 95
IBM 3590 Tape Subsystem . . . 96
NaviQuest . . . 97
OAM Command Changes . . . 98
OAM DUMP Command . . . 98
OAM RESTART Command . . . 99
OAM QUERY Command . . . 99
OAM MODIFY UPDATE Command . . . 100
OAM LIBRARY DISABLE Command . . . 100
OAM Move Volume Enhancements for Supporting Tape Media . . . 100
OAM System Management Facility Recording . . . 102
OAM Optical Library 3995-Cxx Series . . . 103
OAM Volume Stacking and the Virtual Tape Server Subsystem . . . 104
OPEN/CLOSE/End-of-Volume Serviceability Improvements . . . 105
Program Management Extensions . . . 106
Sequential Access Method Tailored Compression . . . 107
Storage Management Subsystem Out-of-Space (X37 Abend) Reduction . . . 109
VSAM Catalog Search Interface . . . 110
VSAM DCOLLECT Enhancements . . . 111
VSAM Last-Referenced Date at Close Processing . . . 112
VSAM Load Enhancements . . . 112
VSAM LSR Enhancements . . . 113
VSAM System-Managed Buffering . . . 113

Migration Planning for DFSMS/MVS Version 1 Release 3 . . . 115
Access Method Services Usability Enhancements . . . 115
BSAM, QSAM, and VSAM Access to Files in a Hierarchical File System . . . 117
Catalog Usability Enhancements . . . 120

Character Data Representation Architecture Services 121
Concatenated Parmlib 122
Data Set Account Field Support 122
DFSMSdss Multivolume Data Set Allocation Enhancement 123
DFSMSdss Support for IMS Backup-While-Open 124
DFSMSdss Utility Message Control 124
DFSMShsm CPOOL Performance Improvement 125
DFSMShsm Greater than 13 Backup Versions 126
DFSMShsm Multitasking Recovery 128
DFSMShsm General Performance, Function and Usability Improvements . . 130
DFSMSrmm Enhancements 132
Enhanced LNKLIST Processing 133
Expiration Date Override User Notification 133
Extended Remote Copy Support 134
Hierarchical File System Standards Compliance 134
Installation Control of PDSE Hiperspace Size 135
ISMF Usability Enhancements 136
Licensing Compliance Changes 137
Program Management Extensions 137
Storage Management Subsystem 32 Name Support 141
Storage Management Subsystem Usability Enhancements 148
Tape Mount Management Enhancements 149
Unit Control Block Virtual Storage Constraint Relief 154
VSAM Extended Addressability 157
VSAM Record-Level Sharing 161

Migration Planning for DFSMS/MVS Version 1 Release 2 177
Backup-While-Open Improvements 177
DFSMShsm Recycle Enhancements 178
DFSMShsm Tape I/O Improvements 179
DFSMShsm Virtual Storage Constraint Relief 180
Distributed FileManager Target Server 180
Enhanced Volume Selection for the Storage Management Subsystem 182
IBM 3495 Model M10 Tape Library Dataserver 185
IBM 3995 Optical Library Dataserver Models 133 and 113 186
OAM Objects on Tape Support 187
OAM RMF Transaction Reporting 188
OS/390 UNIX System Services 189
Overriding Expiration Dates When Deleting Data Sets 190
QSAM or BSAM Compression 191
Storage Management Subsystem System Groups 193
Unit Control Block Constraint Relief 196
VSAM Compression 198
VSAM Partial Release 200

Migration Planning for DFSMS/MVS Version 1 Release 1 203
Aggregate Backup and Recovery Support 203
BSAM, QSAM, BPAM, and BDAM 31-Bit Addressing Support 203
Concurrent Copy 204
Data Set Performance Statistics 204
Device Console Service Improvements 204
Dynamic Cache Management Enhancements 205
Empty Data Set Support Enhancements 205
Extended Format Sequential Data Sets 205

PDSE Copy Performance Improvements . . . 206
PDSE Cross-System Sharing Support . . . 206
Program Management Enhancements . . . 206
Tape Library Support . . . 206
Volume Mount Analyzer . . . 207

Planning to Use DFSMS/MVS Services . . . 209

Setting up the Interactive Storage Management Facility . . . 213
Storing ISMF Load Module Libraries . . . 213
Library Placement . . . 213
Library Access Control . . . 214
Storing ISMF Distribution Libraries . . . 214
Allocating ISPF Table Libraries . . . 214
Making ISMF Available to TSO Users . . . 215
Modifying the TSO Logon Procedure . . . 215
Setting Up a CLIST . . . 216
Modifying Menus to Invoke ISMF . . . 217

Preparing for System-Managed Storage . . . 219
Catalogs in a System-Managed Storage Environment . . . 219
Indexed VTOCs for System-Managed Volumes . . . 220
VSAM Volume Data Sets for System-Managed Volumes . . . 220
Virtual Storage Requirements for the Storage Management Subsystem . . . 220
Steps for Developing an Implementation Plan . . . 220

Evaluating Program and Data Management Functions . . . 225
Planning for PDSEs . . . 225
Understanding the Structure of a PDSE . . . 226
Defining PDSEs . . . 227
Migrating Data From Partitioned Data Sets to PDSEs . . . 230
Sharing Access to a PDSE . . . 231
PDSE Processing Restrictions . . . 232
Planning to Access Program Objects . . . 234
Processing Restrictions for Program Objects . . . 234
Planning for Extended Format Data Sets . . . 234
Planning for Hierarchical File System Data Sets . . . 235
Creating Hierarchical File System Data Sets . . . 236
Managing File System Size . . . 238
Managing File System Activity . . . 238
Transporting a File System . . . 238
Removing (Deleting) a File System . . . 239
Migrating a File System . . . 239
Backing Up File Systems . . . 239
Recovering a Backed-Up File System . . . 239

Planning for Tape in a System-Managed Environment . . . 241
Optimizing Your Current Tape Environment . . . 241
Analyzing Tape Usage with the Volume Mount Analyzer . . . 242
Converting Tape Volumes to System Management . . . 243
Conversion Activities . . . 244

Preparing to Implement DFSMSrmm . . . 247

Evaluating Removable Media Management Needs 247
Assessing DASD Space Requirements 248
DASD Space for the DFSMSrmm Control Data Set 248
DASD Space for the Journal 250
DASD Space for the Problem Determination Aid Log Data Sets 251
Planning DFSMSrmm Authorization and Security 251
Deciding How to Run DFSMShsm with DFSMSrmm 252
Planning for Operating DFSMSrmm 252
Planning for DFSMSrmm Customization 253
Planning to Use DFSMSrmm Utilities 254
Planning Dependencies on Open, Close, End-of-Volume 255

Preparing to Implement DFSMShsm 257
Developing a DFSMShsm Implementation Plan 257
Planning the Processor Environment 257
Planning for Data Security 258
Preparing DFSMShsm Data Sets and Volumes 258
DFSMShsm Data Sets 258
DFSMShsm-Owned Volumes 260
Planning for DFSMShsm-Managed Data Sets and Volumes 261
Setting Availability Management Policies 261
On-Site Data Set Recoverability 261
On-Site Volume Recoverability 263
Off-Site Recoverability 265
Setting Space Management Policies 266
Management Class Migration and Expiration Parameters 266
Automatic Primary Space Management Window 266
Automatic Primary Space Management Cycle 267
Number of Automatic Volume Space Management Tasks 267
Interval Migration 267
Automatic Secondary Space Management Window 267
Automatic Secondary Space Management Cycle 267
Migration Level 1 Volumes 268
Migration Level 2 Media 268
Small Data Set Packing 268
Pooling Volumes for Non-SMS-Managed Data Sets 270
Pools in Systems Running under JES2 270
Pools in Systems Running under JES3 271
Setting Up Tape Management for DFSMShsm 271
Scratch Pool Environment 273
Output Tape Selection 273
Recycling Tapes 273
Device Selection 274
Tape Operations 274
Fully unattended operation 274
Partially unattended operation 274
Testing DFSMShsm 275
Customizing and Tuning DFSMShsm 276
Coordinating DFSMShsm Processing 277
Sample Plans 279
Sample Setup Plan 279
Sample Plan for Allocating DFSMShsm Data Sets 280
Sample Plan for Maintaining Control Data Sets 281
Sample Plan for Availability Management 282

Sample Plan for Space Management . . . 283
Sample Plan for Customizing and Tuning DFSMShsm . . . 285

Securing DFSMS/MVS Resources . . . 287
Using RACF Profiles . . . 287
Protecting ISMF Functions . . . 288
Protecting Data Management Functions . . . 288
Protecting VSAM Record-level Sharing Functions . . . 288
Protecting DFSMSrmm Functions . . . 288
Protecting DFSMSdss Functions . . . 289
Data and Volume Access . . . 289
DFSMSdss Commands and Keywords . . . 289
Protecting DFSMShsm Functions . . . 289
Data Set Access . . . 289
Tape Access . . . 289
DFSMShsm Commands . . . 290

Appendixes . . . 291

Appendix A. DFSMS/MVS Library and Related Publications . . . 293
DFSMS/MVS Library . . . 293
Storage Subsystem Library . . . 298
Related Product Publications . . . 299
Electronic Access to Online IBM Books . . . 303
Internet Access . . . 304
IBM Public Books . . . 304
How To Subscribe . . . 304

Appendix B. Abbreviations . . . 305

Glossary . . . 309

Index . . . 323

Figures

1. Storage requirements for expanded SMS control data sets 144
2. Planning checklist for VSAM record-level sharing. 162
3. Example of TSO Logon Procedure Specifying ISMF Load Module Data Sets 216
4. Example of a CLIST That Makes Load Module Data Sets Available to TSO Users 217
5. Example of a modified ISPF Primary Option Menu 218
6. Overview of DFSMShsm Installation Environment for Control Data Sets, Journal, and Logs 258
7. Small-Data-Set-Packing Data Set Overview 269
8. Life Cycle of a Tape 272
9. Tape Efficiency Through Recycle Processing 274
10. Coordinating DFSMShsm Processing Functions 278

© Copyright IBM Corp. 1979, 1999

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates.

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 of the intellectual property rights of IBM may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, programs, or services, except those expressly designated by IBM, are the responsibility of the user.

IBM may have patents or pending patent applications covering subject matter 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
USA

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. DWZ
5600 Cottle Road
San Jose, CA 95193

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

Any pointers in this publication to non-IBM Web sites are provided for convenience only, and do not in any manner serve as an endorsement of these Web sites.

Programming Interface Information

This publication primarily documents information that is NOT intended to be used as a programming interface of DFSMS/MVS.

This book also documents intended programming interfaces that allow the customer to write programs to obtain the services of DFSMS/MVS. 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

© Copyright IBM Corp. 1979, 1999

Trademarks

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

3090
3890
AD/Cycle
ADSTAR
AFP
AIX
AnyNet
AS/400
BookManager
C/370
CICS
CICS/ESA
CICS/MVS
CUA
DATABASE 2
DB2
DFDSM
DFSMS
DFSMS/MVS
DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm
DFSORT
Enterprise Systems Architecture/390
ES/9000
ESA/370
ESA/390
ESCON
FFST/MVS
GDDM
Hardware Configuration Definition
Hiperbatch
Hiperspace
IBM
IMS
IMS/ESA
Language Environment
Magstar
MVS/DFP
MVS/ESA
MVS/SP
MVS/XA
NetView
OPC
OS/2
OS/390
Parallel Sysplex
Print Services Facility
Personal System/2
PS/2
PSF for OS/390
QMF
RACF
RAMAC
Resource Measurement Facility
RETAIN
RMF
RISC System/6000
RS/6000
S/370
S/390
SAA
SOMobjects
System/370
System/390
Systems Application Architecture
SystemView
VisualLift
VTAM

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Other company, product, and service names, which may be denoted by a double asterisk (**), may be trademarks or service marks of others.

About This Book

This planning guide is intended for system programmers, system analysts, storage administrators, and other members of a data processing team who make system resource and planning decisions, and who are responsible for installing DFSMS/MVS.

This book helps you plan for the installation of DFSMS/MVS. It describes the key components and requirements of DFSMS/MVS and presents information to help you position your current environment for a new release of the product. It also provides general planning considerations for converting your data to system-managed storage and for implementing other services DFSMS/MVS offers.

To use this book effectively, you should be familiar with software installation procedures; have a working knowledge of data storage environments; and have some understanding of the Storage Management Subsystem (SMS).

The planning information presented in this book assists you in migrating to a new release of DFSMS/MVS from an existing MVS or OS/390 system. Migration tasks are more complex for non-MVS users; this book does not cover migration from OS/VS1, VSE, or other operating systems.

For complete information on the actual installation process, including the most current list of installation requirements, refer to your copy of the *DFSMS/MVS Program Directory*. You can find a softcopy version of the directory on the Installation page of the OS/390 web site at:

`http://www.ibm.com/s390/os390/installation/`

This planning guide is most useful before and during installation. Detailed tasks related to implementing DFSMS/MVS features or facilities are described in other books more specific to those functions. For example, you can find information about tailoring the services of DFSMShsm, one of the DFSMS/MVS functional components, by reading the *DFSMS/MVS DFSMShsm Implementation and Customization Guide*.

To obtain a softcopy version of this book, along with other DFSMS/MVS publications, you can go to the following sources:

- At the Library page of the OS/390 web site at:

  `http://www.ibm.com/s390/os390/bkserv/`
- On CD-ROM, *DFSMS/MVS Online Product Library*
- On CD-ROM, *IBM Online Library Omnibus Edition OS/390 Collection*

For a complete list of DFSMS/MVS publications and related information sources, see Appendix A, “DFSMS/MVS Library and Related Publications” on page 293.

© Copyright IBM Corp. 1979, 1999

How to Tell if this Book is Current

IBM regularly updates its books with new and changed information. When first published, both hardcopy and BookManager softcopy versions of a book are identical, but subsequent updates might be available in softcopy before they are available in hardcopy. Here's how to determine the level of a book:

- Check the book's order number suffix (often referred to as the dash level). A book with a higher dash level is more current than one with a lower dash level. For example, in the publication order number SC26-4930-02, the dash level 02 means that the book is more current than previous levels, such as 01 or 00. Suffix numbers are updated as a product moves from release to release, as well as for hardcopy updates within a given release.
- Check to see if you are using the latest softcopy version. To do this, compare the last two characters of the book's file name (also called the book name). The higher the number, the more recent the book. For example, DGT1U302 is more recent than DGT1U301.
- Compare the dates of the hardcopy and softcopy versions of the books. Even if the hardcopy and softcopy versions of the book have the same dash level, the softcopy could be more current. This will not be apparent from looking at the edition notice. The edition notice number and date remain that of the last hardcopy version. When you are looking at the softcopy product bookshelf, check the date shown to the right of the book title. This will be the date that the softcopy version was created.

  Also, an asterisk (*) is added next to the new and changed book titles in the CD-ROM booklet and the README files.

Vertical lines to the left of the text indicate changes or additions to the text and illustrations. For a book that has been updated in softcopy only, the vertical lines indicate changes made since the last printed version.

References to Product Names Used in DFSMS/MVS Publications

DFSMS/MVS publications support DFSMS/MVS, 5695-DF1, as well as the DFSMSdfp base element and the DFSMShsm, DFSMSdss, and DFSMSrmm features of OS/390, 5647-A01. DFSMS/MVS publications also describe how DFSMS/MVS interacts with other IBM products to perform the essential data, storage, program and device management functions of the operating system.

DFSMS/MVS publications typically refer to another IBM product using a generic name for the product. When a particular release level of a product is relevant, the reference includes the complete name of that product. This section explains the naming conventions used in the DFSMS/MVS library for the following products:

MVS can refer to:

- MVS/ESA SP Version 5, 5695-047 or 5695-048
- The MVS base control program (BCP) of OS/390, 5647-A01

All MVS book titles used in DFSMS/MVS publications refer to the OS/390 editions. Users of MVS/ESA SP Version 5 should use the corresponding MVS/ESA book.

Refer to *OS/390 Information Roadmap* for titles and order numbers for all the elements and features of OS/390.

For more information about OS/390 elements and features, including their relationship to MVS/ESA SP and related products, please refer to *OS/390 Planning for Installation*.

RACF can refer to:

- Resource Access Control Facility (RACF), Version 2, 5695-039
- The RACF element of the OS/390 Security Server, an optional feature of OS/390

All RACF book titles refer to the Security Server editions. Users of RACF Version 2 should use the corresponding book for their level of the product. Refer to *OS/390 Security Server (RACF) Introduction* for more information about the Security Server.

CICS can refer to:

- CICS/MVS, 5665-403
- CICS/ESA, 5685-083
- The CICS element of the CICS Transaction Server for OS/390, 5665-147

All CICS book titles refer to the CICS Transaction Server for OS/390 editions. Users of CICS/MVS and CICS/ESA should use the corresponding books for those products. Please see *CICS Transaction Server for OS/390: Planning for Installation* for more information.

Summary of Changes

This section outlines the history of changes to the book's content. Technical changes and additions for the most recent edition (listed below) are indicated by a vertical line to the left of the change. Maintenance and editorial updates are also included in each revision.

Product highlights are not included in this section. For similar product information, please see “Migration Planning for DFSMS/MVS Version 1 Release 5” on page 35, which describes each new functional enhancement and its requirements for implementation, or take a look at *DFSMS/MVS General Information*, “What's New in Version 1 Release 5”, which provides an overview of each new function.

Seventh Edition, March 1999

This publication is a major revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 5. Following is a summary of the new or changed information:

- Throughout the book, references to the OpenEdition product have changed to the new name OS/390 UNIX System Services, or simply OS/390 UNIX.
- The chapter on “Considerations for Multisystem Environments” on page 29 explains how the structure of SMS control data sets has changed for this release and the compatibility requirements for sharing with lower releases.
- “Migration Planning for DFSMS/MVS Version 1 Release 5” on page 35 is a new chapter that describes functional enhancements and requirements for this release of the product.
- Discussions about the VSAM extended format KSDS in “Planning for Migration and Coexistence” chapters are expanded to include the other extended format record organizations.
- “PDSE Storage Requirements” on page 228 includes revised information for PDSE data areas and other support requirements.
- “Extended Remote Copy Support” on page 134 has additional information.
- The DFSMS/MVS Macro List (Appendix A in past releases) has been removed.
- References to supplemental planning information have been expanded to include the url for the OS/390 web site. You can access the complete DFSMS/MVS library, along with the Program Directory from this site.

Note: For other important updates to this book, please check informational APAR II11474, a repository of DFSMS/MVS 1.5 information that was not available at the time DFSMS/MVS books were published for general availability.

Sixth Edition, March 1998

This publication is a minor revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 4. Included here is a summary of the new or changed information.

© Copyright IBM Corp. 1979, 1999

A new chapter, “Year 2000 Support for DFSMS/MVS” on page 25 was added to present Year 2000 migration planning considerations. “Migration Planning for DFSMS/MVS Version 1 Release 4” on page 77 has been updated with more details about the following functions:

- Storage Management Subsystem out-of-space (X37 Abend) reduction
- VSAM catalog search interface
- VSAM DCOLLECT enhancements
- VSAM last-referenced date at close
- VSAM load enhancements
- VSAM LSR enhancements
- VSAM system-managed buffering

Fifth Edition, June 1997

This publication is a major revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 4. The following summarizes the new or changed information.

Updates have been made to “Considerations for Multisystem Environments” on page 29.

“Migration Planning for DFSMS/MVS Version 1 Release 4” on page 77 is a new chapter that describes functional enhancements and associated requirements for this release of the product.

Updates have been made to “Migration Planning for DFSMS/MVS Version 1 Release 3” on page 115 for “DFSMSrmm Enhancements” on page 132 and “VSAM Record-Level Sharing” on page 161.

Updates have been made to “Evaluating Program and Data Management Functions” on page 225, “Sharing Access to a PDSE” on page 231.

Fourth Edition, September 1996

This publication is a minor revision in support of the service updates and functional changes introduced since general availability of DFSMS/MVS Version 1 Release 3.

Information has been added to “Migration Planning for DFSMS/MVS Version 1 Release 3” on page 115 for these new functions:

- BSAM, QSAM, and VSAM access to hierarchical file system (HFS) files.
- The DFSMSdss optional VOLCOUNT parameter for data set RESTORE and data set COPY of SMS data sets.
- Backup-while-open serialization for IMS data sets.
- DFSMShsm support for the MVS concatenated parmlib function, which allows multiple data sets to be used as a logical parmlib.
- DFSMShsm year 2000 support, which allows users to specify dates beyond 1999.
- Back up of the DFSMSrmm control data set using DFSMSdss concurrent copy to minimize impact to the tape environment.
- DFSMSrmm enhancements for running DFSMSrmm with JES3.

- Enhanced DFSMSrmm reporting capability.

"VSAM Record-Level Sharing" on page 161 includes new information about IGDSMSxx parmlib members, SMF record types, fallback and coexistence considerations, and more.

The section "Sharing Access to a PDSE" on page 231 was rewritten to more clearly describe PDSE sharing.

Third Edition, October 1995

This publication is a major revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 3.

This book was reorganized into three new parts to expand migration and coexistence information for the DFSMS/MVS 1.3 functions. Planning topics are now subdivided into the following parts:

- Planning to install DFSMS/MVS
- Planning for migration and coexistence
- Planning to implement DFSMS/MVS services

Appendix B is a new section that consolidates the lists of DFSMS/MVS books and related product publications into a single reference. The table containing DFSMS/MVS publications is expanded to include audience information for each book.

Second Edition, March 1994

This publication is a major revision in support of the functional changes introduced with DFSMS/MVS Version 1 Release 2.

This book was reorganized. Part 2 of the previous edition was removed as specific installation steps are already contained in the DFSMS/MVS *DFSMS/MVS Program Directory* Information about installation requirements contained in Part 2 are now presented in "DFSMS/MVS Installation Requirements" on page 13.

The chapter on Migration and Coexistence is new, and contains information to help you take advantage of new features.

Planning to Install DFSMS/MVS

The DFSMS/MVS Environment—an Overview . . . 3
DFSMS/MVS in the MVS/ESA or OS/390 Operating System . . . 3
The DFSMS System-Managed Storage Environment . . . 3
DFSMS/MVS Functional Components . . . 3
DFSMSdfp . . . 4
DFSMSdss . . . 4
DFSMShsm . . . 5
DFSMSrmm . . . 5
Additional DFSMS/MVS Features and Supporting Products . . . 6
DFSMS Optimizer . . . 6
Network File System . . . 7
DFSORT . . . 7

DFSMS/MVS Packaging, Enablement, and Support . . . 9
Product Offerings . . . 9
Distribution Package . . . 9
Publications . . . 10
Licensing and Enablement . . . 10
Service Agreements . . . 11

DFSMS/MVS Installation Requirements . . . 13
Hardware Requirements . . . 13
Software Requirements . . . 13
Program Products Used for Installation . . . 13
Corequisite System Products . . . 14
Products Recommended for DFSMS/MVS Full-Function Support . . . 14
Storage Requirements . . . 15
SMP/E Data Sets . . . 15
New System Data Sets . . . 16
Device Support Considerations . . . 16
I/O Devices . . . 16
Device Initialization . . . 16
Operating Modes . . . 17

© Copyright IBM Corp. 1979, 1999

The DFSMS/MVS Environment—an Overview

This chapter gives a brief overview of DFSMS/MVS, its components, available configurations, and functions. For a more comprehensive discussion of DFSMS/MVS features and functions, see *DFSMS/MVS General Information*.

DFSMS/MVS in the MVS/ESA or OS/390 Operating System

DFSMS/MVS and MVS/ESA, or OS/390 BCP, comprise the base operating system where DFSMS/MVS performs the essential data, storage, program, and device management functions of the system.

DFSMS/MVS is the central component of both system-managed and non-system-managed storage environments. The operating system supports both 24-bit and 31-bit addressing used by DFSMS/MVS functions. Many DFSMS/MVS components have modules or data in extended virtual storage above 16MB, leaving more space below the 16MB line for user applications.

The DFSMS System-Managed Storage Environment

The *DFSMS environment* consists of a set of IBM hardware and software products that together provide a system-managed storage solution for MVS installations. DFSMS/MVS is an integral part of this environment.

The components of DFSMS/MVS automate and centralize storage management based on installation-defined policies for availability, performance, space, and security. The Interactive Storage Management Facility (ISMF) provides the user interface for defining and maintaining these policies and the Storage Management Subsystem (SMS) governs these policies for the system.

In this environment, the Resource Access Control Facility (RACF) and Data Facility Sort (DFSORT) complement the functions of the base operating system: RACF provides resource security functions, and DFSORT adds the capability for faster and more efficient sorting, merging, copying, reporting and analyzing of business information.

DFSMS/MVS Functional Components

DFSMS/MVS is comprised of the following functional components:

DFSMSdfp
DFSMSdss
DFSMShsm
DFSMSrmm

DFSMS/MVS is built upon the functions formerly provided by MVS/DFP, Data Facility Data Set Services (DFDSS), and the Data Facility Hierarchical Storage Manager (DFHSM). DFSMS/MVS eliminates the need to order and install each product individually, making the installation task much easier. The tighter integration of functional components facilitates cross-functional testing, enhancing the overall quality of the product. Most importantly, DFSMS/MVS improves upon the earlier offerings by adding support for many new functions.

© Copyright IBM Corp. 1979, 1999

The sections that follow describe the primary services that each DFSMS/MVS functional component provides.

DFSMSdfp

DFSMSdfp provides the foundation for:

Storage management
DFSMSdfp includes ISMF, an interactive facility that allows you to define and maintain policies to manage your storage resources. These policies help to improve the utilization of storage devices, and to increase levels of service for user data, with minimal effort required from users. The Storage Management Subsystem (SMS) manages these policies for the operating system.

Tape mount management
DFSMSdfp's Storage Management Subsystem provides a means for implementing tape mount management (TMM), a methodology for improving tape utilization and reducing tape costs. This methodology involves intercepting selected tape data set allocations through the SMS automatic class selection (ACS) process, and redirecting them to a DASD buffer. Once on DASD, these data sets can be migrated to a single tape or small set of tapes, thereby reducing the overhead associated with multiple tape mounts.

Data management
DFSMSdfp helps you store and catalog information on DASD, optical, and tape devices, so that it can be quickly identified and retrieved from the system.

Program management
DFSMSdfp combines programs into executable modules, prepares them to run on the operating system, stores them into program libraries, and reads them into storage for execution.

Device management
DFSMSdfp is involved in defining your input and output devices to the system, and in controlling the operation of those devices in the MVS/ESA environment.

Distributed data access

Distributed data access allows all authorized systems and users in a network to exploit the powerful features of system-managed storage provided by DFSMS/MVS. DFSMSdfp uses the Distributed FileManager (DFM) to support remote access of MVS/ESA data and storage resources from workstations, personal computers, or any other system on an SNA LU 6.2 network.

DFSMSdss

DFSMSdss is used for:

Data movement and replication
DFSMSdss lets you move or copy data between volumes of like and unlike device types. It can also copy data that has been backed up.

Space management
DFSMSdss can reduce or eliminate DASD free-space fragmentation.

Data backup and recovery
DFSMSdss provides host system backup and recovery functions at both the data set and volume levels. It also includes a stand-alone restore program that you can run without a host operating system.

Data set and volume conversion
DFSMSdss can be used to convert your data sets and volumes to system-managed storage. It can also return your data to a non-system-managed state as part of a recovery procedure.

DFSMShsm

DFSMShsm provides functions for:

Storage management
DFSMShsm uses a hierarchy of storage devices in its automatic management of data, relieving end-users from manual storage management tasks.

Space management
DFSMShsm improves DASD space utilization by keeping only active data on fast-access storage devices. It automatically frees space on user volumes by deleting eligible data sets, releasing over-allocated space, and moving low-activity data to lower-cost-per-byte devices.

Tape mount management
DFSMShsm can write multiple output data sets to a single tape, making it a useful tool for implementing tape mount management (TMM) under SMS. When you redirect tape data set allocations to DASD, DFSMShsm can move those data sets to tape, as a group, during interval migration. This greatly reduces the number of tape mounts initiated on the system. DFSMShsm uses single file format, which also improves your tape usage and search capabilities.

Availability management
DFSMShsm backs up your data—automatically or by command—to ensure availability in the event of accidental loss of data sets or physical loss of volumes. DFSMShsm also allows a storage administrator to copy backup and migration tapes. These copies can be stored on site as protection from media damage, or off site as protection from site damage. Disaster backup and recovery is also provided for user-defined groups of data sets (aggregates), so that critical applications can be restored at the same location or at an off-site location.

Full exploitation of DFSMShsm services in a system-managed storage environment requires the use of DFSMSdss for certain functions. You can also use the DFSMS Optimizer feature to monitor and tune DFSMShsm functions.

DFSMSrmm

DFSMSrmm manages your removable media resources, including tape cartridges and reels. It provides functions for:

Library Management
You can create tape libraries, or collections of tape media associated with tape drives, to balance the workload of your tape drives and operators. DFSMSrmm can be used to manage system-managed tape libraries such as the IBM 3494 and IBM 3495 Automated Tape Library Dataservers or the

manual IBM 3495 Tape Library Dataserver Model M10. DFSMSrmm can also manage the more traditional non-system-managed tape libraries.

Shelf Management
DFSMSrmm groups information about removable media by shelves into a central online inventory and keeps track of the volumes residing on those shelves. DFSMSrmm can also record the shelf location for optical disks and track their vital records status.

Volume management
DFSMSrmm helps manage the movement and retention of tape volumes throughout their life cycle.

Data set management
DFSMSrmm records information about the data sets on tape volumes to validate volume and data set information and to help maintain data integrity. It can also control the retention of those data sets.

Additional DFSMS/MVS Features and Supporting Products

There are a number of separately orderable features and products available to complement DFSMS/MVS functions:

DFSMS Optimizer
DFSMS/MVS Network File System
DFSORT

The special system services offered by these features and products are introduced in the sections that follow.

DFSMS Optimizer

The DFSMS Optimizer provides analysis and simulation information for both SMS and non-SMS users. The DFSMS Optimizer can help you maximize storage use and minimize storage costs. It provides methods and facilities for you to:

- Monitor and tune DFSMShsm functions.
- Create and maintain a historical database of system and data activity.
- Perform in-depth analysis of:
  - management class policies, including simulations and cost-benefit-analysis using your storage component costs.
  - storage class policies for SMS data, with recommendations for both SMS and non-SMS data.
  - high I/O activity data sets, including recommendations for placement and simulation for cache and expanded storage.
  - storage hardware performance of subsystems and volumes including I/O rate, response time, and caching statistics.
- Fine tune an SMS configuration, by helping you:
  - understand how current SMS policies and procedures are managed.
  - determine associated costs of current data management practices.
  - simulate potential policy changes and understand the costs of those changes.

- Produce presentation-quality charts.

For more information on the DFSMS Optimizer, see the *DFSMS Optimizer User's Guide and Reference*.

Network File System

The OS/390 Network File System (formerly DFSMS/MVS Network File System) enables an MVS system to act as a file server or client to workstations, personal computers, or other authorized systems that use the SUN NFS Version 2 protocols in a TCP/IP network.

The clients of the DFSMS/MVS server can remotely access OS/390 UNIX files as well as conventional MVS data sets, bringing the resources of an MVS system—system-managed storage, high-performance storage access, file access security, and centralized data access—to client platforms.

The Network File System client allows MVS applications, including ones using BSAM, QSAM, and VSAM, to access remote files on a variety of platforms.

For more information on the Network File System, refer to *OS/390 Network File System Customization and Operation* and *OS/390 Network File System User's Guide*.

DFSORT

Data Facility Sort (DFSORT) is a highly flexible data processing tool that provides fast and efficient sorting, merging, copying, reporting, and analysis of your business information. Following are a few of the many tasks you can perform with DFSORT:

- Sort, merge, or copy files; including or excluding records, and reformatting records.
- Analyze data and produce detailed reports using the ICETOOL utility or the OUTFIL function. In addition, OUTFIL allows you to create different views of the data and different reports with a single pass over the data.
- Adapt to the sorting and merging needs of different countries using DFSORT's national language support.
- Provide correct ordering of 2-digit years and transformation of 2-digit years to 4-digit years as part of IBM's year 2000 solution.

For more information on DFSORT, its capabilities and requirements, see the *DFSORT Brochure* (GC33-4033).

DFSMS/MVS Packaging, Enablement, and Support

This chapter introduces the available configurations of the DFSMS/MVS product and explains how the product is shipped to your installation. It also describes the procedures you use to enable the DFSMS/MVS configuration selected for your installation, and how you can easily update this configuration to reflect changes in your licensing agreement.

Product Offerings

The full-function DFSMS/MVS product provides all four functional components. There are also several alternate offerings. With an alternate offering, you obtain a license to activate only the DFSMS/MVS functional components that meet your business needs. Table 1 shows the functional components you can use for each available offering.

Table 1. Configurations of DFSMS/MVS

Configuration	Functional Components of DFSMS/MVS
Full-Function Product	DFSMSdfp, DFSMSdss, DFSMShsm, DFSMSrmm
Alternate Offering 1	DFSMSdfp
Alternate Offering 2	DFSMSdfp, DFSMSdss
Alternate Offering 3	DFSMSdfp, DFSMSdss, DFSMShsm
Alternate Offering 4	DFSMSdfp, DFSMSrmm
Alternate Offering 5	DFSMSdfp, DFSMSdss, DFSMSrmm

Note: Distributed FileManager support in DFSMSdfp is available with any of the DFSMS/MVS offerings listed.

Because of DFSMS/MVS inter-component dependencies, you must install all four functional components shipped on your program tape, regardless of which offering you intend to use. You are charged, however, for only the functional components you activate on your system.

Be aware that DFSMS/MVS cannot coexist with an earlier version of its component parts; for example, you cannot run DFSMSdfp from Alternate Offering 1 on the same system with DFHSM Version 2 Release 6.

The DFSMS Optimizer and other DFSMS/MVS features are available as separately orderable licensed programs. Selected features are also available as an option of the DFSMS/MVS tape distribution package.

Distribution Package

As a DFSMS/MVS license holder, you receive the following in the product distribution package:

- DFSMS/MVS program tape
- Optional DFSMS/MVS feature tape
- DFSMS/MVS Program Directory
- Single printed copy of selected DFSMS/MVS publications

© Copyright IBM Corp. 1979, 1999

- CD-ROM containing the *DFSMS/MVS Online Product Library*

Note: The *DFSMS/MVS Online Product Library*, (LK2T-8731) is a CD-ROM that contains unencrypted licensed materials and therefore is available only to licensees of DFSMS/MVS.

The *DFSMS/MVS Program Directory* provides information about the program materials along with a complete list of requirements and instructions for installing DFSMS/MVS.

When you order the DFSMS Optimizer or other DFSMS/MVS features, you receive a separate program directory containing instructions for installing those features on your system.

Publications

You can order publications from the DFSMS/MVS library through the IBM Distribution Center or through your IBM representative. Electronic access is also available. See Appendix A, “DFSMS/MVS Library and Related Publications” on page 293 for order and access information.

In addition to the CD-ROM you receive on the product tape, you can also receive the entire DFSMS/MVS library in BookManager readable format by ordering the *IBM Online Library Omnibus Edition OS/390 Collection*, a CD-ROM that contains several MVS-based product libraries. A single copy of this CD-ROM is available to you at no charge when you purchase the appropriate level of OS/390.

Licensing and Enablement

In an MVS/ESA SP installation (non-OS/390), you selectively enable DFSMS/MVS functional components and features on your system by updating SYS1.PARMLIB member IGDDFPKG according to the product offerings your installation is licensed to use. IGDDFPKG includes these parameters:

DFSMS_OFFERING	Use this keyword to specify your DFSMS/MVS offering, either the full-function product or an alternate offering.
DFSMS_FEATURE	Use this keyword to specify any of the optional features found on your DFSMS/MVS product tape.

Normally, you update IGDDFPKG as part of the DFSMS/MVS installation procedure and the settings become effective with the initial system IPL. After installation, you can easily modify IGDDFPKG to reflect changes in your licensing agreement; for example, to add another DFSMS/MVS functional component or feature to your system. By invoking a procedure called DFSMSPKG, these updates can become effective on your system without an IPL.[1]

DFSMSPKG is invoked as a started task using the operator START command (START DFSMSPKG). As a cataloged procedure residing in SYS1.PROCLIB,

[1] Procedure DFSMSPKG is not available for DFSMS/MVS Version 1 Release 1 or Release 2 systems. Toleration PTFs are available to allow systems at these release levels to ignore the IGDDFPKG parameter DFSMS_FEATURE.

DFSMSPKG can only be started by a user with operator authority. The procedure executes program IGDSMSPK. Modifications to IGDDFPKG are in effect when this function completes.

In an OS/390 installation, you do not use the IGDDFPKG parmlib to enable DFSMS/MVS. Instead, IBM supplies a tailored SYS1.PARMLIB member, IFAPRD00, to enable the elements and features ordered with OS/390. The IFAPRD00 member is not active by default. For enablement, you must copy the contents of IDAPRD00 to an active IFAPRDxx member that you establish through the PROD parameter in IEASYSxx or the SET PROD operator command.

Characteristics and syntax rules of IGDDFPKG, and associated information about DFSMSPKG, along with details about IDAPRDxx, are explained in the *OS/390 MVS Initialization and Tuning Reference*.

Service Agreements

DFSMS/MVS is a licensed program supported by IBM Central Service which includes the IBM Support Center. Consult your IBM representative for information on the service agreement for DFSMS/MVS.

DFSMS/MVS Installation Requirements

This chapter identifies the hardware and software products required to install DFSMS/MVS and to fully exploit all of the available functions of the product. Hardware and software dependencies specific to individual functions in this product release are discussed in “Migration Planning for DFSMS/MVS Version 1 Release 5” on page 35.

Hardware Requirements

DFSMS/MVS has the same minimum hardware requirements as its MVS/ESA or OS/390 counterpart. Requirements include at least one Enterprise Systems Architecture (ESA) processor, printer, and console, in addition to DASD and tape devices for storage.

Additional hardware is needed to exploit some product features, as outlined in the *DFSMS/MVS Program Directory*. Requirements for the following are included:

- Concurrent copy (CC)
- Data compression
- DFSMSdss SnapShot and virtual concurrent copy (VCC)
- Enhanced catalog sharing (ECS)
- Enhanced dynamic cache management
- Extended remote copy (XRC)
- Virtual tape server (VTS)
- VSAM record-level sharing (RLS)

More information about these functions is also provided in the “Planning for Migration and Coexistence” chapters of this book.

Software Requirements

DFSMS/MVS requires several licensed programs for the installation process. Other program products are recommended to fully activate the functions available with DFSMS/MVS. This section discusses both the required and optional programs.

Some of the programs listed here require additional maintenance. See the *DFSMS/MVS Program Directory* and Remote Technical Assistance and Information Network Preventive Service Planning (RETAIN PSP) installation information for the most up-to-date maintenance recommendations.

Program Products Used for Installation

The following licensed programs are prerequisites for installing and maintaining DFSMS/MVS. Subsequent versions, releases, and modification levels can be used unless otherwise announced.

- A base MVS system with the binder provided by DFSMS/MVS, or a linkage editor provided by MVS/DFP Version 3 (5665-XA3)

© Copyright IBM Corp. 1979, 1999

- High Level Assembler (HLA) for MVS, VM, and VSE (5696-234), with current maintenance
- System Modification Program Extended (SMP/E) Version 1 Release 8.1 (5668-949)

SMP/E is used to incorporate new products into your system libraries. SMP/E must be installed on the generating system before MVS/ESA and DFSMS/MVS are installed. Modifications to the new MVS system should be made in the SMP/E installation format to simplify future maintenance. For more information on SMP/E, see the *OS/390 SMP/E User's Guide*.

The SMP/E RECEIVE, APPLY, and ACCEPT processes used to install DFSMS/MVS are documented in the *DFSMS/MVS Program Directory*.

Corequisite System Products

This release of DFSMS/MVS requires, at a miminum, OS/390 Version 2 Release 5 or MVS/ESA SP Version 5 Release 2, along with a compatible level of either JES2 or JES3. *OS/390 Planning for Installation* documents the supported releases of JES2 and JES3 and also provides a list of the licensed programs required to install the OS/390 product.

Some DFSMS/MVS functions require a higher release level of these system products, or a comparable PTF, for enablement. See “Migration Planning for DFSMS/MVS Version 1 Release 5” on page 35 for function-specific software dependencies.

Products Recommended for DFSMS/MVS Full-Function Support

The licensed programs listed in Table 2 are recommended for DFSMS/MVS full-function support. Some functions also require compatibility PTFs, which are listed in the *DFSMS/MVS Program Directory*.

OS/390 customers should use the corresponding element or feature of OS/390. See *OS/390 Planning for Installation* for information about the relationship between OS/390 and root product levels.

Table 2 (Page 1 of 2). Licensed Program Release Levels for Full-Function Support

Licensed Program	Product Number	Rel.	Function
DB2 Version 4	5695–DB2	1	OAM parallel sysplex support.
DFSORT	5740-SM1	13	Sorting, merging, and copying data sets.
ICKDSF	5655-257	16	Device support facilities for initialization of system-managed volumes and track recovery.
EREP Version 3	5658-260	5	Error analysis and recording, and for the 3380 Models AD4, BD4, AE4, BE4, AJ4, AK4, BJ4, BK4, and CJ2.
ISPF/PDF Version 4	5655–042	2	Dialog manager for interactive applications. Required for ISMF functions.
ICSF/MVS	5685-051	1	Cryptographic services to allow data encryption using a programmed algorithm.

Table 2 (Page 2 of 2). Licensed Program Release Levels for Full-Function Support

Licensed Program	Product Number	Rel.	Function
Language Environment for MVS and VM Version 1	5688-198	3	Language environment for numeric and character data (CDRA) conversion.
Print Services Facility for OS/390 Version 3	5655-B17	1	Print service facilities.
RACF Version 2	5740-XXH	1	Authorization and protection of data and system resources.
RMF Version 5	5685-029	3	Resource measurement facility for I/O device activity reporting.
TSO/E Version 2	5685-025	4	Data set creation and display of DFSMS/MVS information.

Licensed programs listed in Table 3, if installed, must be at the minimum release levels indicated to exploit all the functions of DFSMS/MVS.

Table 3. Licensed Program Release Levels for Database and Data Control Programs

Licensed Program	Product Number	Rel.	Function
CICS Transaction Server for OS/390	5665-083	1.0	VSAM record-level sharing support.
CICSVR MVS/ESA Version 2	5695-010	3.0	Forward recovery of CICS VSAM data sets backed up by concurrent copy support.
IMS/ESA Version 6	5655-158	1.0	Data sharing; with database manager and transaction manager features.

Storage Requirements

The MVS operating system requires storage for DFSMS/MVS, for JES2 or JES3, for all licensed programs installed on the system, and for the various distribution and target libraries supported by each product. Storage requirements vary depending upon the products installed on your system and the DASD you choose to store on. Information on storage requirements is provided by the program directories or installation guides for these products.

Virtual storage and DASD space requirements for the installation of DFSMS/MVS are specified in the *DFSMS/MVS Program Directory*.

SMP/E Data Sets

DFSMS/MVS installation requires a large amount of space in the SMP/E data sets. Check the amount of space available in your SMP/E data sets prior to installation, and between installation steps, to ensure that sufficient space is available.

Because of the large number of macros distributed with DFSMS/MVS, you might want to use a temporary SMPMTS data set during the installation; it can then be deleted after the installation is complete. For more information on using a temporary SMPMTS data set, see *DFSMS/MVS Program Directory*.

Note: SMP/E does not support PDSEs used as work libraries for programs.

New System Data Sets

You need to initialize DASD volumes to contain the new system data sets for the product. Use the ICKDSF INIT command to write the volume labels, VTOCs (with or without an index), and IPL text on the volumes. The *ICKDSF R16 User's Guide* describes the steps for initializing DASD volumes.

Device Support Considerations

This section describes I/O devices currently supported by DFSMS/MVS. It also lists the device types that are no longer supported in this environment.

I/O Devices

The hardware configuration of your processors and I/O devices determines how many devices you can attach to your system. MVS supports up to 65,536 attached devices, with eight access paths each. The specific number of devices you can attach depends on the configuration of your processor and I/O devices.

OS/390 and MVS/ESA SP provide support for I/O devices from earlier MVS/ESA and MVS/XA systems, with the exception of the following:

- IBM System/3 or 1130 operating as a JES2 remote workstation
- IBM 3850 Mass Storage System
- IBM 2305, 3330, 3333, 3340, 3344, 3350, 3350P, 3351P, and 3375 DASD
- IBM 3830 and 3880 Models 1, 4, 11, and 21 Storage Controls

If you have data stored on unsupported DASD devices, you must move the data to an IBM 3380, 3390, or 9345 device geometry before you migrate your system to this product release.

Table 4 on page 18 summarizes the most commonly used IBM I/O devices supported by MVS/ESA SP that are also directly supported by DFSMS/MVS functions.

Some devices require a specific level of hardware maintenance to operate properly on an MVS system. DFSMS/MVS software support for new hardware devices might also require the installation of PTFs. These PTFs are listed in the RETAIN PSP Hardware Install Indexes.

If you have a question about support for a device not listed here, or if you want information about hardware maintenance levels, contact your IBM marketing representative.

Device Initialization

You can define new devices to the system by using the interactive panels of the hardware configuration definition (HCD) program. HCD has dynamic I/O capabilities, changing hardware definitions without the need for an IPL or hard power-on reset.[2]

You have the option to define devices either above or below 16MB by specifying the LOCANY parameter on the HCD panel. You can find information on defining

[2] CVOLs are not supported on devices defined using dynamic I/O.

UCBs above the 16MB line under “Unit Control Block Virtual Storage Constraint Relief” on page 154. For other information on HCD, see *OS/390 HCD Planning*.

Operating Modes

Most devices attached to MVS operate in *full function mode*; that is, all features on the device are compatible with, and usable on, the operating system. Some of these features include:

- With DASD devices: dynamic path reconnection, extended count-key-data operation, and caching and cache-related facilities
- With tape devices: cartridge stack loading and data compaction

Some devices also operate in *compatibility mode*, which allows you to simulate the function of another device or model. Compatibility mode causes the device to function like a different device of the same type, ignoring some or all of the additional features the device might have. This allows you to migrate between devices with minimal impact on programs that have device dependencies.

Devices in an IBM 3495 Tape Library Dataserver are defined to the operating system automatically by the IBM 3495; however, you should use HCD to ensure that the devices are set properly if they are offline during the system IPL. Procedures for installing and maintaining the IBM 3495 are described in the *3495 Tape Library Dataserver Installation Planning and Migration Guide*. For JES3, there are additional initialization deck requirements. These are discussed in the *OS/390 JES3 Initialization and Tuning Guide*.

Table 4 (Page 1 of 2). IBM I/O Devices and Subsystems Supported by OS/390

Direct Access Storage

3380 Direct Access Storage[1]
3390 Direct Access Storage[1]
Models 1, 2, 3, and 9
9340 Direct Access Storage[2]
9341—9345 Direct Access Storage
9343—9345 Direct Access Storage
9396 Ramac Scalable Array Storage
9397 Ramac Electronic Array Storage
RAMAC Array Device[11]
9391 RAMAC Array Subsystem[11]

Storage Control Unit

3880 Storage Control
Model 3[14]
3990 Storage Control
Models 1 and 2[1]

Cache Storage Control Unit

3880 Storage Control
Model 23 with 3880 AJ4/AK4 Attachment (Feature 3010)
3990 Storage Control
Models 3 and 6[1 11 12]
9393 RAMAC Virtual Array Storage Subsystem
Model xx2[15]

Console

3180 Display Station
Models 140 and 145
3205 Color Display Station
Model 100
3206 Display Station
Models 100 and 110
3251 Display Station
3277 Display Station
Models 1 and 2
3278 Display Station
Models 1, 2, 2A, 3, and 4
3279 Color Display Station
Models 2A, 2B, 2C, 3A, and 3B
5080 High Function Graphics System

Optical

9246 Optical Library Unit RPQ# 8B6001[10]
9247 Optical Disk Drive RPQ# 8B6003[10]
Expand 9246 to Four Drives RPQ# 8B6002[10]
MVS/ESA Direct Optical Attachment RPQ# 8B6004[10]
3995 Optical Library Dataserver
Models 11x, 13x, 15x, C1x, C3x
3995 Optical Library Drive
Models–SWx
S/370 and S/390 Optical Media Attach/2[13]

Tape

2440 Magnetic Tape Subsystem
Models A01, A02, B01, and J02[3]
3420 Magnetic Tape Unit
Models 3, 4, 5, 6, 7, and 8
3422 Magnetic Tape Subsystem
3423 Device[13]
3424 Magnetic Tape Subsystem[4]
3430 Magnetic Tape Subsystem
3480 Magnetic Tape Subsystem
3490 Magnetic Tape Subsystem[5]
Base models
Models A01, A02, B02, and B04
Enhanced capability models
Models A10, A20, B20, B40, C10, C11, C1A, C22, C2A, D41, and D42
3494 Tape Library Dataserver
Model L10
3494 Virtual Tape Server
Models B16, B18
3495 Tape Library Dataserver
Automated
Models L20, L30, L40, and L50
Manual
M10
3590 Magnetic Tape Subsystem
3590-1 Magnetic Tape Subsystem

Printer

1403 Printer
Models 2, 7, and N1
3203 Printer
Model 5
3211 Printer
3262 Line Printer
Model 5[6]
3284 Printer
Models 1 and 2
3286 Printer
Models 1 and 2
3800 Printing Subsystem
Models 3, 6, 8[7] and 1
3812 Page Printer
Model 2[7]
3816 Page Printer
Models 01S, and 01D[7]
3820 Page Printer[8]
3825 Page Printer[8 9]
3827 Page Printer[8 9]
3828 Advanced Function MICR Printer[8 9]
3835 Page Printer[8 9]
3900 Advanced Function Printer[8 9]
4028 LaserPrinter
Model NS1[7]
4245 Line Printer
Models 01, 12, and 20
4248 Impact Line Printer
Models 1 and 2
6262 Impact Line Printer
Models 014, and 022[6]

Other

2501 Card Reader
Models B1 and B2
2540 Card Read Punch
Models 1 and 2
3505 Card Reader
3525 Card Punch
37xx Communications Controllers
3838 Array Processor
3848 Cryptographic Unit
3890 Document Processor

Table 4 (Page 2 of 2). IBM I/O Devices and Subsystems Supported by OS/390

Note:

1. Most models of the IBM 3380 Direct Access Storage and all models of the IBM 3390 Direct Access Storage can operate with Enterprise Systems Connection (ESCON) channels when attached to a 3990 Storage Control Model 2, 3, or 6 with ESCON capability.
2. All models of the IBM 9345 Direct Access Storage Devices mounted in an IBM 9343—9345 Model D04 cabinet can operate with ESCON channels.
3. These models of the IBM 2440 Magnetic Tape Subsystem are supported in 3420-4 compatibility mode.
4. The 3424 Magnetic Tape Unit is available only in Brazil, S.A.
5. All models of the IBM 3490 Magnetic Tape Subsystem can operate with ESCON channels if an ESCON adapter is installed.
6. This printer must be initialized to the system as 4248.
7. This printer requires the use of PSF for OS/390 if you want it to run as an Advanced Function Printer (AFP*). PSF for OS/390 is a separate product.
8. This printer requires the use of PSF for OS/390, which is a separate product.
9. This printer must be defined to the system as AFP1.
10. This feature or device is supported by the object access method.
11. All RAMAC subsystems, and the 3990 Storage Control Models 3 and 6 support SAM striping and SAM and VSAM KSDS compression.
12. The 3990 Storage Control Models 3 and 6 Extended Platform support concurrent copy.
13. The IBM S/370 and the S/390 Optical Media Attach/2 products emulate an IBM 3422 Magnetic Tape Subsystem, supporting both 1600 and 6250 bpi formats. They are defined to the system as 3423 tape devices.
14. The 3880 Storage Control Model 3 can support the 3380 AJ4/AK4 Attachment (Feature 3005).
15. The RAMAC Virtual Array supports both 3380 and 3390 device types in single, double, and triple capacity volume sizes.

Planning for Migration and Coexistence

Year 2000 Support for DFSMS/MVS 25
Summary of Year 2000 Changes 25
Migration Actions 27
Special Actions During Testing 27
Migration Assistance 28

Considerations for Multisystem Environments 29
Shared Resources in a Storage Management Subsystem Complex 29
Shared Catalogs 30
System-Managed and Non-System-Managed Data 31
Partitioned Data Sets Extended 31
VSAM Data Sets 31
Checkpoint Data Sets 32
Tape Libraries 32
Program Management 32
Serialization for DFSMShsm Functions 32
Mixed Releases of DFDSS or DFHSM 32
VTOC Compatibility Between MVS/ESA and MVS/XA Systems 33

Migration Planning for DFSMS/MVS Version 1 Release 5 35
Catalog Alias and Usability Improvements 36
Catalog Sharing Enhancements for Sysplex Environments 38
Control Data Set Structure Changes for SMS 39
DEB Table Expansion 40
DFSMSdss Logical Release Support 41
DFSMSdss SnapShot Support 41
DFSMSdss Storage Group Filtering 42
DFSMShsm Secondary Host Promotion 43
DFSMShsm Control Data Set Extended Addressability 44
DFSMShsm Overhead Reduction 45
DFSMShsm GRSplex Serialization 46
DFSMShsm General Function and Usability Improvements 47
DFSMSrmm Application Programming Interface 49
DFSMSrmm Catalog Status Tracking 50
DFSMSrmm Disposition Control 51
DFSMSrmm Subsystem Enhancements 52
DFSMSrmm TSO Subcommand Enhancements 53
DFSMSrmm Vital Record Specification Enhancements 53
Extended Remote Copy Enhancements 55
Hierarchical File System Performance and Other Enhancements 55
ISO/ANSI Version 4 Tape Support 57
NaviQuest Batch Support 60
OAM Parallel Sysplex Support 61
OAM Pseudo Library Concept 62
OAM Support for the 3995-SW4 Optical Drive 64
OAM New and Changed Operator Commands for Object Support 64
Reformatting a 3995 Optical Disk 64
Relabeling a 3995 Optical Disk Volume 65
Displaying OAM XCF Status 65
Displaying SETOPT and SETOAM Parameters 65

© Copyright IBM Corp. 1979, 1999

Using the Update Command to Set SETOAM and SETOPT Values 66
Virtual Tape Server Import/Export Support 67
Exporting Volumes 68
Importing Volumes 69
VSAM Extended Format Data Set Enhancements 72

Migration Planning for DFSMS/MVS Version 1 Release 4 77
DFM/MVS DataAgent 78
DFSMSdss Enhanced Protection of Checkpointed Data Sets 79
DFSMSdss Multivolume Selection 80
DFSMSdss Stand-Alone Services 80
DFSMShsm ABARS Activity Log Deletion 81
DFSMShsm ABARS Externalization of the TGTGDS and OPTIMIZE Parameters 82
DFSMShsm ABARS File Stacking 83
DFSMShsm ABARS Generation Data Group Base Name in Allocate 84
DFSMShsm ABARS 64 Tasks 85
DFSMShsm CDS RLS Serialization 85
DFSMShsm CPU Time in WWFSR and ABR Records 87
DFSMShsm Dump Analysis Elimination 88
DFSMShsm Duplex Tape Function 89
DFSMShsm Enhanced Protection of Checkpointed Data Sets 91
DFSMShsm Extended Invocation of the ARCBEEXT Exit 93
DFSMShsm General Function and Usability Improvements 93
DFSMSrmm Enhancements 95
IBM 3590 Tape Subsystem 96
NaviQuest 97
OAM Command Changes 98
OAM DUMP Command 98
OAM RESTART Command 99
OAM QUERY Command 99
OAM MODIFY UPDATE Command 100
OAM LIBRARY DISABLE Command 100
OAM Move Volume Enhancements for Supporting Tape Media 100
OAM System Management Facility Recording 102
OAM Optical Library 3995-Cxx Series 103
OAM Volume Stacking and the Virtual Tape Server Subsystem 104
OPEN/CLOSE/End-of-Volume Serviceability Improvements 105
Program Management Extensions 106
Sequential Access Method Tailored Compression 107
Storage Management Subsystem Out-of-Space (X37 Abend) Reduction . . . 109
VSAM Catalog Search Interface 110
VSAM DCOLLECT Enhancements 111
VSAM Last-Referenced Date at Close Processing 112
VSAM Load Enhancements 112
VSAM LSR Enhancements 113
VSAM System-Managed Buffering 113

Migration Planning for DFSMS/MVS Version 1 Release 3 115
Access Method Services Usability Enhancements 115
BSAM, QSAM, and VSAM Access to Files in a Hierarchical File System . . 117
Catalog Usability Enhancements 120
Character Data Representation Architecture Services 121
Concatenated Parmlib 122

Data Set Account Field Support 122
DFSMSdss Multivolume Data Set Allocation Enhancement 123
DFSMSdss Support for IMS Backup-While-Open 124
DFSMSdss Utility Message Control 124
DFSMShsm CPOOL Performance Improvement 125
DFSMShsm Greater than 13 Backup Versions 126
DFSMShsm Multitasking Recovery 128
DFSMShsm General Performance, Function and Usability Improvements . . 130
DFSMSrmm Enhancements 132
Enhanced LNKLIST Processing 133
Expiration Date Override User Notification 133
Extended Remote Copy Support 134
Hierarchical File System Standards Compliance 134
Installation Control of PDSE Hiperspace Size 135
ISMF Usability Enhancements 136
Licensing Compliance Changes 137
Program Management Extensions 137
Storage Management Subsystem 32 Name Support 141
Storage Management Subsystem Usability Enhancements 148
Tape Mount Management Enhancements 149
Unit Control Block Virtual Storage Constraint Relief 154
VSAM Extended Addressability 157
VSAM Record-Level Sharing 161

Migration Planning for DFSMS/MVS Version 1 Release 2 177
Backup-While-Open Improvements 177
DFSMShsm Recycle Enhancements 178
DFSMShsm Tape I/O Improvements 179
DFSMShsm Virtual Storage Constraint Relief 180
Distributed FileManager Target Server 180
Enhanced Volume Selection for the Storage Management Subsystem 182
IBM 3495 Model M10 Tape Library Dataserver 185
IBM 3995 Optical Library Dataserver Models 133 and 113 186
OAM Objects on Tape Support 187
OAM RMF Transaction Reporting 188
OS/390 UNIX System Services 189
Overriding Expiration Dates When Deleting Data Sets 190
QSAM or BSAM Compression 191
Storage Management Subsystem System Groups 193
Unit Control Block Constraint Relief 196
VSAM Compression 198
VSAM Partial Release 200

Migration Planning for DFSMS/MVS Version 1 Release 1 203
Aggregate Backup and Recovery Support 203
BSAM, QSAM, BPAM, and BDAM 31-Bit Addressing Support 203
Concurrent Copy 204
Data Set Performance Statistics 204
Device Console Service Improvements 204
Dynamic Cache Management Enhancements 205
Empty Data Set Support Enhancements 205
Extended Format Sequential Data Sets 205
PDSE Copy Performance Improvements 206
PDSE Cross-System Sharing Support 206

Program Management Enhancements 206
Tape Library Support 206
Volume Mount Analyzer 207

Year 2000 Support for DFSMS/MVS

As an element of the MVS or OS/390 operating system, DFSMS/MVS is certified as Year 2000-ready by the Information Technology Association of America (ITAA).

DFSMS/MVS Version 1 Release 2 and Release 3 are Year 2000-ready with maintenance applied. DFSMS/MVS Version 1 Release 4 and subsequent releases are shipped Year 2000-ready. Always consult the Year 2000 PSP bucket for the latest maintenance information.

This chapter provides an overview of the changes introduced by DFSMS/MVS in support of the year 2000. It includes considerations for migration and testing and also presents a list of additional sources where you can get assistance with the migration effort.

Summary of Year 2000 Changes

DFSMS/MVS includes a number of changes to messages, reporting records, data area fields, and catalog information to handle date information in the new millennium.

Modifications have been made to input and output date formats to recognize century values, allowing data with dates beyond the year 2000 to be processed correctly. For example, some dates in the form of *mm/dd/yy* were changed to *mm/dd/yyyy*. Other dates in the form of *yy.ddd* were changed to *yyyy.ddd*. Refer to the specific command or function for the exact format. Inclusion of this century information allows a valid comparison to be made between data stored before, during, and after the year 2000.

Restarting a system with old log data or older record formats for constructs such as catalogs, VTOCs, or control data sets continues to work after the year 2000. In some cases, however, formatting has changed to accomodate a full date specification; therefore, care must be taken to process any date information in the same format in which it was stored.

It is especially important to note that only ICF catalogs are supported in this environment. Data cataloged in VSAM catalogs will no longer be accessible once the date goes beyond December 31, 1999.

DFSMSdfp has added support to the storage management subsystem to correctly calculate dates, including leap year dates, for the year 2000 and beyond. Other DFSMSdfp changes include the following:

- All access methods and catalog services correctly handle date attributes created before, during, and after the year 2000.
- IEBCOPY utility uses 4-digit years in all date displays and calculations.
- ISMF data set lists now display “NEVER” for a data set with no expiration date, rather than display a date of “1999/12/31.”
- DFSMSdfp messages written to SYSLOG now include dates with a century field.

© Copyright IBM Corp. 1979, 1999

DFSMSdss commands can all be used to process data sets associated with dates that follow 12/31/1999 (1999-December-31). DFSMSdss Year 2000 support includes changes to the following:

- Expiration date processing now enables DFSMSdss to process data sets with never-expiring dates such as “99.365” or “99.366.”
- BY filtering using keywords CREDT, REFDT, or EXPDT now allows 7-digit dates (*yyyyddd*). The explanation of DFSMSdss message ADR140E was modified to support the changes in BY filtering.
- DFSMSdss messages now have 4-digit years in their titles and headers. Changed messages include:

  ```
  ADR006I        ADR109I        ADR734I
  ADR012I        ADR208I        ADR735W
  ADR013I        ADR209I        ADR738E
  ADR014I        ADR213I        ADR767I
  ADR017E        ADR720I        ADR962I
  ADR108I        ADR726W        ADR963I
  ```

  Messages now show the date as, for example, “1998.305” rather than “98305.”

DFSMShsm is enhanced to enable automatic functions to continue to start after 12/31/1999 (1999-December-31). All commands can be used to process data sets with dates later than 12/31/1999. DFSMShsm Year 2000 support includes changes to the following:

- ARCHRCOV macro, DATE=*dateaddr* parameter
- ARCPRPDO program, START and END parameters
- DEFINE command, CYCLESTARTDATE parameter
- DEFINE command, DUMPCLASS parameter, RETENTIONPERIOD(*days*) and TAPEEXPIRATIONDATE(*yyyyddd*|*yyddd*) subparameters
- HRECOVER and RECOVER commands, DATE parameter
- REPORT command, FROMDATE(*date*) and TODATE(*date*) parameters
- SETSYS command, CDSVERSIONBACKUP parameter, BACKUPDEVICECATEGORY(TAPE) and (EXPIRATIONDATE(*expirationdate*)) subparameters
- TAPECOPY command, INDATASET, EXPDT, and RETPD parameters
- Fields in the DFSMShsm data areas: CDSV-CDSVERSIONBACKUP Parameters Table, and MCD-Migration Control Data Set Data Set Record
- DFSMShsm messages with new or changed text:

  ```
  ARC0103I       ARC0284I       ARC2006I
  ARC0216I       ARC0424I       ARC2007I
  ARC0281I       ARC0430I       ARC2008I
  ARC0282I       ARC1067I       ARC2009I
  ARC0283I
  ```

- DFSMShsm messages with text changed specifically to include a 4-digit year:

```
ARC0526I        ARC0718I        ARC0742I
ARC0530I        ARC0722I        ARC0743I
ARC0622I        ARC0724I        ARC0750I
ARC0623I        ARC0735I        ARC0832I
ARC0680I        ARC0740I        ARC6045I
ARC0681I        ARC0741I        ARC6102I
```

DFSMSrmm and DFSMSopt are relatively new to the MVS environment and were developed using 4-digit year formats as part of their initial offerings.

Migration Actions

Consider the following actions for your Year 2000 migration plan:

- VSAM catalogs cannot be processed by MVS or OS/390 once the system date goes beyond 1999; therefore, you must convert these catalogs to an ICF structure before the end of 1999.
- Stored values for date formats have been redefined and expanded to represent dates beyond 1999. A date in the VTOC, for example, uses a 3-byte binary value represented by *yyddd*. The format has not actually changed but the definition of the contents has changed. The *yy* value, which is the offset from the year 1900, now has a valid range of “0” to “255.” Originally, *yy* had a maximum value of “99.” The *ddd* value represents the day of the year, which continues to have a valid range between “1” and “366.”

  Other data structures use a 4-byte hexadecimal date field with a format of *0cyydddf*. The century value is contained in the high-order byte, whereas in the past, this byte was reserved.

  You need to assess the impact these changes have on older applications that process dates based on the older data area formats.
- Many installation exits allow your code to examine or set date values. You must evaluate these exits to make sure they continue to process correctly as you implement other Year 2000 changes. For example, ensure that the code can adjust to the updated formats provided by other interfacing programs, so that they can provide or accept 4-digit years. They also need to properly handle the specification of “never-expire” for an expiration date.

Special Actions During Testing

Keep the following in mind when setting up tests of the Year 2000 environment:

- If you run tests with the system date set ahead, be sure to isolate all your test data at both the volume and catalog level. Production data should not be accessible to the Year 2000 test environment.
- When setting the system clock ahead for testing, also be careful when running products that delete expired data sets. After creating data sets for Year 2000 testing, you must ensure that when expiration date processing is run after the system clock change, these test data sets are not automatically deleted due to lack of reference or age. For example, if a data set indicates that it is to be kept one year after it is last referenced and the data set was created on 12/31/1997, the data set could be deleted by automatic expiration processing once the system date is set to 1/1/1999 or later. This results in your test data being deleted before it is actually processed.

Migration Assistance

For further information on Year 2000 support and migration assistance, refer to the following sources:

- IBM's Year 2000 web site describes migration tools, offerings, and other services that are available. Visit the web site at:

  `http://www.ibm.com/year2000`

- *The Year 2000 and 2-Digit Dates: A Guide for Planning and Implementation*, GC28-1251, provides information to help you plan your migration to a Year 2000-ready environment, including how to determine the best techniques for reformatting year-date notation and what techniques to use for testing Year 2000 changes. This guide is available on-line, which you can download from IBM's Year 2000 web site. It can also be ordered through your IBM representative.
- The YR2000MVS PSP bucket contains the most up-to-date Year 2000 maintenance information.
- The IBM Year 2000 Technical Support Center can be contacted through the internet at `y2ktsc@us.ibm.com`, or by calling 1-800-IBM-4YOU and mentioning the “Call-down Keyword” of “Y2K.”

Considerations for Multisystem Environments

If you intend to share catalogs, DASD volumes, or data sets in a parallel sysplex or other multiprocessor environment, you must provide for data integrity across the systems sharing these resources. This chapter introduces resource-sharing considerations and also describes coexistence implications for multisystem environments using DFSMS/MVS with MVS/DFP Version 3.

For parallel sysplex migration planning, including storage management and data sharing techniques, also refer to the following books in the MVS/ESA Sysplex System Level Library:

OS/390 Parallel Sysplex Overview
OS/390 Parallel Sysplex Systems Management
OS/390 Parallel Sysplex Hardware and Software Migration
OS/390 Parallel Sysplex Application Migration

Shared Resources in a Storage Management Subsystem Complex

It is possible to combine up to 32 MVS systems in a Storage Management Subsystem (SMS) complex, to share both system-managed and non-system-managed resources.

In a multisystem SMS complex, the operating systems share a common set of SMS classes, groups, ACS routines, and a configuration base, which make up the storage management policy for the complex. This storage management policy is maintained in a source control data set (SCDS). When this policy is activated for the Storage Management Subsystem (SMS) the bound policy is maintained in processor storage and on DASD in an active control data set (ACDS). Systems in the complex communicate SMS information through a common communications data set (COMMDS).

When running in an SMS multiprocessor complex with mixed levels of the product, you must do the following on the system running the highest release level of DFSMS/MVS:

- Define your SMS constructs and ACS routines
- Translate your ACS routines
- Validate your SCDS.

You can then activate and share the validated SCDS among systems with mixed levels of DFSMS/MVS. If you do not translate your ACS routines and validate your SCDS on the highest level of DFSMS/MVS, the translation and validation might fail. This failure occurs because some constructs are only known to higher levels of DFSMS/MVS.

Also be aware that the structure of SMS control data sets changed dramatically with the functions introduced in DFSMS/MVS 1.3 and DFSMS/MVS 1.5. Maintenance is available for DFSMS/MVS 1.2, and above, to allow DFSMS/MVS 1.5 to share SMS control data sets with lower level systems in the SMS complex. MVS/DFP V3 and DFSMS/MVS 1.1 systems cannot share SMS control data sets

© Copyright IBM Corp. 1979, 1999

with systems at the DFSMS/MVS 1.5 level; however, they can share DASD data sets and volumes that are not SMS-managed.

In addition, sharing control data sets with DFSMS/MVS 1.2 is possible only if you run DFSMS/MVS in *compatibility mode*. With compatibility mode, no more than eight system names can be defined in the SMS configuration. This enables the system to share the SMS control data sets with DFSMS/MVS 1.2, and with other systems running in compatibility mode.

When you convert an SMS system to *32-name mode*, the SMS control data sets are converted to a new format that supports up to 32 system names. This conversion is permanent. Once converted to 32-name mode, the configuration can no longer be accessed by a release lower than DFSMS/MVS 1.3. For more information, see “Storage Management Subsystem 32 Name Support” on page 141.

In the case where SMS is not active on any one of the systems in a multisystem sharing environment, users on that system cannot:

- Create data sets on system-managed volumes.
- Delete system-managed data sets.
- Extend system-managed data sets to new volumes.
- Use JCL keywords supported by SMS.

The COMMDS does not record DASD space usage changes for a system that has not activated SMS.

Shared Catalogs

A *shared catalog* is a basic catalog structure that is used by more than one system. It must be defined with SHAREOPTIONS(3 4) and reside on a DASD volume defined through the HCD as a shared device. By default, all catalogs and VSAM volume data sets (VVDSs) are defined as shared.

In a system-managed storage environment, all permanent DASD data sets must be cataloged. Although a catalog contains entries for system-managed data, the catalog itself does not have to be system-managed; however, we recommend that catalogs containing entries for system-managed data sets also be system-managed. A catalog can contain entries for both data sets that are system-managed and data sets that are not system-managed.

To share catalogs between systems when some of the systems are running DFSMS/MVS or MVS/DFP Version 3 without SMS active, and the others are running MVS/XA DFP Version 2, you must apply toleration PTFs to the Version 2 systems. Otherwise, an MVS/ESA catalog which is accessed by an MVS/XA DFP Version 2 system could be permanently damaged, requiring a catalog recovery.

Note: These toleration PTFs do not allow the MVS/XA DFP system to recognize SMS information.

To successfully share a catalog between a DFSMS/MVS system and an MVS/XA DFP Version 2 system, you must ensure that:

- The catalog does not reside on a system-managed volume.
- Data sets defined in the catalog do not reside on system-managed volumes.
- Catalog toleration PTFs are applied to the MVS/XA DFP system.
- The catalog is defined with SHAREOPTIONS(3 4).

- The catalog's volume is initialized as shared.

If the configuration you use during SMS testing is a combination of an MVS/XA DFP Version 2 system and DFSMS/MVS or MVS/DFP Version 3, you must isolate the system-managed volumes from the MVS/XA DFP Version 2 system. This prevents the MVS/XA DFP system from:

- Deleting or renaming system-managed data sets
- Creating new data sets on system-managed volumes
- Using the access method services ALTER REMOVEVOLUMES command against the master catalog to perform a VSAM volume cleanup of a system-managed volume
- Sharing catalogs with the DFSMS/MVS or MVS/DFP Version 3 system.

For information on sharing catalogs in a sysplex environment, see “Catalog Sharing Enhancements for Sysplex Environments” on page 38. For further information on catalog sharing, see *DFSMS/MVS Managing Catalogs*.

System-Managed and Non-System-Managed Data

You can share system-managed data sets between any systems using the Storage Management Subsystem and sharing the same SMS control data sets.

If you transfer data between systems using the TSO TRANSMIT and RECEIVE commands, the Distributed FileManager (DFM), or the NetView File Transfer Program for MVS, SMS class names are not transmitted with the data set because SMS control information is stored externally, not in the data set.

If SMS receives a data set from either a system-managed or non-system-managed environment, and ACS processing determines that the data set should be system-managed, SMS assigns classes just as it does for any new data set.

If you are sending application JCL to other sites that use the facilities provided by DFSMS/MVS or MVS/DFP Version 3 (such as JCL parameters used with SMS), you need to synchronize the use of SMS at these sites.

Partitioned Data Sets Extended

Access to partitioned data sets extended (PDSEs) can be shared between users of a single processor, and can be extended to users on any processor in a multisystem complex. PDSEs are discussed in Evaluating Program and Data Management Functions. Refer to “Sharing Access to a PDSE” on page 231 for serialization information.

VSAM Data Sets

A parallel sysplex environment set up to access VSAM data sets for record-level sharing must be given careful consideration for maintaining data integrity. Planning for this environment is outlined under “VSAM Record-Level Sharing” on page 161.

Checkpoint Data Sets

Checkpoint data sets contain system data. To ensure integrity of this data, checkpoint data sets are, by default, allowed only on non-shared DASD and tape volumes.

To ensure data set integrity across multiple systems, you can authorize users to create checkpoints on shared DASD by using the RACF Facility Class IHJ.CHKPT.*volser*, where *volser* is the volume serial of the volume to contain the checkpoint data set. For more information, see *DFSMS/MVS Using Data Sets* and *OS/390 MVS Planning: Global Resource Serialization*.

Tape Libraries

A catalog containing the tape configuration database (TCDB) must reside on DASD that is shared among all hosts that can access the tape library.

Program Management

Program management (PM3) defaults to a P03 level program object when binding into a PDSE or into OMVS. On a system level lower than DFSMS/MVS 1.4, the P02 level program object is created. This format cannot be loaded or rebound on an earlier release of DFSMS/MVS.

There is a way for you to change the program management installation default. Until DFSMS/MVS 1.4 and OS/390 R4, LPA only supported partitioned data sets and load modules. DFSMS/MVS now supports the inclusion of PDSEs and program objects in LPA. This support is part of the dynamic LPA function in OS/390. See "Program Management Extensions" on page 106 for more information.

Serialization for DFSMShsm Functions

DFSMShsm provides two alternatives for serialization of user data sets. The first, DFHSMSERIALIZATION, relies on volume reserves and is needed when sharing data with VM, VSE, or MVS systems outside of a GRS environment. In this mode, multi-volume data sets will not be supported to avoid deadlock conditions.

The second, USERDATASETSERIALIZATION, relies on data set enqueues. It is ideal for single-host implementations or multi-host implementations where all MVS systems belong to the same GRS environment to propagate the enqueues correctly. Refer to the *DFSMS/MVS DFSMShsm Implementation and Customization Guide* for details.

Mixed Releases of DFDSS or DFHSM

When converting to DFSMS/MVS while still maintaining releases of DFDSS or DFHSM on other systems, keep the following in mind:

- Although the DFSMSdss functional component can restore data dumped by DFDSS, you might not be able to use DFDSS to restore data dumped by DFSMSdss, particularly extended format data sets and PDSEs.

- If you are going to have DFHSM and DFSMS/MVS with the DFSMShsm functional component running on shared systems, apply the appropriate toleration PTFs to all versions of DFHSM to ensure data integrity and compatibility between the release levels.
- You cannot use the DFSMShsm ARECOVER command with the INSTRUCTION or ACTIVITY parameters to recover the instruction data set or activity log of aggregate backups made with DFHSM. Otherwise, DFHSM aggregate backups can be recovered by DFSMShsm. However, SMS attributes and storage requirements are not written to the ARECOVER activity log.

VTOC Compatibility Between MVS/ESA and MVS/XA Systems

Although the structure of the volume table of contents (VTOC) has not changed between MVS/XA and MVS/ESA, or OS/390, some previously reserved fields in the VTOC's format-1 data set control block (DSCB) have changed. The format of the DSCB is described in *DFSMS/MVS DFSMSdfp Advanced Services*.

Programming Interface Information

MVS/DFP Version 3 and DFSMS/MVS use a previously unused field at offset 78 (`X'4E'`) in the DSCB to store information about both SMS-managed and non-SMS-managed data sets. Before you convert to DFSMS/MVS from versions or releases prior to MVS/DFP Version 3, you must ensure that this DSCB field contains binary zeros on all volumes. Also verify that no other software on your system changes this field. You can get unpredictable errors if this field is used to maintain information for programs other than the operating system. For information about inspecting its contents and resolving any problems, see your IBM representative.

End of Programming Interface Information

Migration Planning for DFSMS/MVS Version 1 Release 5

This chapter presents a technical overview of all new or enhanced functions provided in DFSMS/MVS 1.5:

- Catalog alias and usability improvements
- Catalog sharing enhancements for sysplex environments
- Control data set structure changes for SMS
- DEB table expansion
- DFSMSdss logical release
- DFSMSdss SnapShot support
- DFSMSdss storage group filtering
- DFSMShsm secondary host promotion
- DFSMShsm control data set extended addressability
- DFSMShsm overhead reduction
- DFSMShsm GRSplex serialization
- DFSMShsm general function and usability improvements
- DFSMSrmm application programming interface
- DFSMSrmm catalog status tracking
- DFSMSrmm disposition control
- DFSMSrmm subsystem processing enhancements
- DFSMSrmm TSO subcommand enhancements
- DFSMSrmm vital record specification enhancements
- Extended remote copy enhancements
- Hierarchical file system performance and other enhancements
- ISO/ANSI Version 4 tape support
- NaviQuest batch support
- OAM parallel sysplex support
- OAM pseudo library concept
- OAM support for the 3995–SW4 optical drive
- OAM new and changed operator commands
- Virtual tape server import/export support
- VSAM extended format data set enhancements

For each new type of support, this chapter outlines environmental dependencies and migration considerations and, where appropriate, refers to other books in the DFSMS/MVS and OS/390 libraries where you can find further implementation assistance.

Note: The APARs referenced in this chapter are provided as a courtesy to help you correlate functional enhancements with corresponding toleration maintenance or other programming requisites listed in the *DFSMS/MVS Program Directory* or PSP bucket. The APARs listed here are not considered all-inclusive and are not intended to replace the information you obtain from these other sources.

© Copyright IBM Corp. 1979, 1999

Supplemental Planning Information

Please check the following up-to-date information sources in conjunction with the planning details presented in this chapter:

APAR II11474	This informational APAR contains a repository of DFSMS/MVS 1.5 information that was not available at the time DFSMS/MVS books were published for general availability. Included are any updates to this book.
www.ibm.com/s390/os390/	From the OS/390 web site, you can get an advance copy of the *DFSMS/MVS Program Directory* for this release. Included is a list of toleration and compatibility maintenance along with other DFSMS/MVS 1.5 installation requirements. From the OS/390 home page, click on the *Installation* bullet and then scroll down to find Program Directories under *Other Useful Resources*.

Catalog Alias and Usability Improvements

Catalog usability and performance is improved in the following ways:

- **Extended Alias Entries:** Catalog alias entries can now be defined using system symbols. This enables each system in a sysplex to resolve an alias with a different user catalog or different non-VSAM data set name. For example, the alias name IMSCAT can identify user catalog IMS.PRODCAT on one system and user catalog IMS.TESTCAT on another system.

  Symbolic substitution also allows a non-VSAM data set to be resolved at the time the alias is referenced (and differently on each system), rather than when the alias is defined. For example, SYS1.PLILIB can resolve to SYS1.V4.PLILIB on one system and resolve to SYS1.V5.PLILIB on a another system.

- **MLA enhancements:** In a sysplex environment, virtual storage requirements of the multi-level alias (MLA) table have been reduced, thus improving search performance.

- **Operator command for tracking performance:** The command MODIFY CATALOG,REPORT,PERFORMANCE, is now available to track the performance of key events occurring in the catalog address space.

Other changes to catalog support include the following:

- **IMBED option no longer supported:** IDCAMS will ignore the IMBED parameter on a DEFINE statement. SVC 26 calls specifying a DEFINE with IMBED are still honored, allowing users of this interface to create data sets as necessary; however, users creating new data sets through the documented interfaces can no longer specify the IMBED option.

- **Passwords within ICF catalogs ignored:** Passwords will not be checked when access is authorized for a data set cataloged in an ICF catalog. A security product should be used to control access to catalogs and data sets.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
Attempts to use an alias defined with a symbolic value will fail on systems prior to DFSMS/MVS 1.5.

Although the system no longer checks passwords for ICF-cataloged data sets, it will allow IDCAMS functions that operate on passwords (for example, DEFINE and ALTER) to continue to work in the event a catalog is shared with a pre-DFSMS/MVS 1.5 system that still uses passwords.

Toleration maintenance is needed on lower-level systems to prevent unnecessary IDCAMS errors from occurring as a result of some catalog functions.

APAR Reference

OW32335

System Interfaces Affected
The names for symbolic substitution are defined on the SYMDEF parameter in SYS1.PARMLIB member LOADXX.

Symbolic substitution for non-VSAM data sets occurs at time of use; for catalogs, substitution occurs at IPL, catalog address space restart, or when the master catalog is updated with user catalog names on another shared system.

Operator Commands Modified
Operator command, MODIFY CATALOG,REPORT,PERFORMANCE, extracts information on internal catalog performance. This command tracks and displays the number of key performance-related events in the catalog address space and calculates the average time that elapses when these events occur.

Commands or Control Statements Modified
With the DEFINE ALIAS command, users typically specify a user catalog with the RELATE parameter to indicate that all references to data sets beginning with that alias should use the catalog specified. In the example below, all references to data sets starting with “MYALIAS” will find the data set by way of the user catalog “SAMPLE.USERCAT.”

```
DEFINE ALIAS (NAME(MYALIAS) RELATE(SAMPLE.USERCAT))
```

A new DEFINE ALIAS keyword, SYMBOLICRELATE (or SYMRELATE), is provided for symbolic substitution and is mutually exclusive with the RELATE keyword. In the following example, the value of system symbol “&USERCAT1” can be set to a different user catalog for each system that has the alias defined.

```
DEFINE ALIAS (NAME(MYALIAS) SYMBOLICRELATE('&USERCAT1'))
```

Processing Restrictions
Products that extract information from a catalog on one system and send it to another system might not work properly when a symbolic value is used in an alias reference because the true value of the alias could vary from system to system.

In a JES3 environment, in particular, JCL procedures and INCLUDE segments are expanded from data sets found by LOCATEs occurring on the conversion system. Conversely, problem program data, including STEPLIB, JOBLIB, and DD statements specified in the JCL, and DD statements opened dynamically by a program, are found by LOCATEs occurring on the execution system.

Where to Find More Information

- *DFSMS/MVS General Information*
- *DFSMS/MVS Using Data Sets*
- *DFSMS/MVS DFSMSdfp Storage Administration Reference*
- *DFSMS/MVS Access Method Services for ICF*
- *DFSMS/MVS Managing Catalogs*

Catalog Sharing Enhancements for Sysplex Environments

By using the S/390 coupling facility cache structure instead of DASD to store catalog sharing control information, shared catalog performance in a sysplex environment can be improved considerably. This new sharing method, *enhanced catalog sharing* (ECS), eliminates a reserve, dequeue, and I/O request to the VVDS on most catalog calls.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
A coupling facility is required for ECS. A new cache structure, SYSIGGCAS_ECS is used in the coupling facility for this support. All systems in the sysplex that share catalogs through ECS must have connectivity to SYSIGGCAS_ECS in the coupling facility.

Coexistence Issues
All systems must be at the DFSMS/MVS 1.5 level to share catalogs through ECS. The systems sharing a catalog must use the same method for sharing; mixed-mode (ECS and non-ECS) sharing is not permitted. Explicit action must be taken to enable a catalog for ECS; by default, catalogs are not ECS-eligible.

Toleration PTFs are available for lower-level systems that share catalogs with DFSMS/MVS 1.5 systems to ensure data integrity in this mixed environment. On a lower-level system, or on a DFSMS/MVS 1.5 system in which ECS in not activated, access to a catalog is blocked if that catalog is currently shared through ECS. The systems can continue sharing catalogs that are not using ECS.

If, during testing, it becomes necessary to back out of DFSMS/MVS 1.5, you should deactivate ECS for all catalogs affected, *before* the back-out takes place.

In the case of catalogs that are ECS-enabled, IDCAMS EXPORT or DFSMSdss DUMP/RESTORE or COPY will preserve the ECS indicator that was specified when the catalog was defined or altered. If the catalog is then imported, restored, or copied to a lower-level system, that system will ignore the ECS indicator.

APAR Reference

OW32576

Performance Characteristics

Overall, CPU usage and elapsed time could decrease considerably with this support.

System Interfaces Affected

A Coupling Facility Resource Management (CFRM) policy must be established in setting up the sysplex for enhanced catalog sharing.

All systems in the sysplex sharing catalogs through ECS must be in the same GRS complex. ECS uses a new GRS major name, SYSZCATS, to preserve the integrity of the catalog by preventing mixed-mode (ECS and non-ECS) sharing. This GRS enqueue specifies RNL=NO to ensure that an integrity exposure is not inadvertently created by someone including it in an inappropriate RNL.

Operator Commands Modified

A new parameter, ECSHR, on the operator command, MODIFY CATALOG, activates or deactivates ECS for the entire system or for an individual catalog.

Commands or Control Statements Modified

A new parameter, ECSHARING or NOECSHARING, on the IDCAMS DEFINE USERCATALOG and ALTER commands, determines whether ECS is enabled or disabled for an individual catalog. This allows you to easily stage in the use of this support—even catalog by catalog.

Where to Find More Information

- *DFSMS/MVS General Information*
- *DFSMS/MVS Access Method Services for ICF*
- *DFSMS/MVS Managing Catalogs*
- *OS/390 MVS System Commands*
- *OS/390 MVS Setting Up a Sysplex*

Control Data Set Structure Changes for SMS

The SMS control data set structure has changed significantly with the functions introduced by DFSMS/MVS 1.5.. If you intend to share SMS control data sets (SCDS, ACDS and COMMDS) with lower-level releases, compatibility maintenance must be applied to preserve data integrity in this cross-system sharing environment.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no additional software dependencies.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

Compatibility maintenance is provided for DFSMS/MVS 1.2, and above, to allow DFSMS/MVS 1.5 to share SMS control data sets with lower level systems in the SMS complex. MVS/DFP V3 and DFSMS/MVS 1.1 systems cannot share SMS

control data sets with systems at the DFSMS/MVS 1.5 level; however, they can share DASD data sets and volumes that are not SMS-managed.

APAR Reference

OW31930

Where to Find More Information

- *DFSMS/MVS DFSMSdfp Storage Administration Reference*

DEB Table Expansion

Prior to this release, the maximum size of the DEB table was approximately 32K. This allowed for up to 8191 DEB entries per address space. Each entry represents a DEB for a DCB or ACB, opened for a data set.

The DEB table has moved above the 16MB line of storage, providing for greater flexibility in sizing. The DEB table can now accommodate up to 65532 DEB entries per address space.

This enhancement allows installations to fully exploit the 10K DD limit available with IMS 6.1, enabling IMS to allocate and open a large number of OSAM or VSAM data sets.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

OS/390 Release 5 and MVS/ESA SP 5.2 systems require maintenance to allow EXCP purge processing to accept the new DEB table format. These changes were incorporated into OS/390 Release 6.

Additional maintenance is needed for the Language Environment for MVS and VM product.

APAR Reference

OW31912, PQ16014 (LE)

Hardware Dependencies

There are no hardware dependencies for this support.

Coexistence Issues

No toleration PTFs are needed for previous releases.

Programming Interfaces Affected

There are no external programming interfaces introduced or changed by this support. Applications using the EXCP macro or any of the standard data management macros, such as OPEN or CLOSE, do not require modification. The larger DEB table is obtained above the 16MB line automatically during the first DEBCHK TYPE=ADD processing for an address space.

Any advanced applications that access the DEBTBLOF field in the DEB, or process the DEB table directly without using the DEBCHK macro, must be modified to handle the new DEB table format and storage location. Applications can check a flag in the DEB table header to determine whether the new or old format of the DEB table is active. As an extra precaution, the new DEB table

header was designed in such a way that a failure will occur if any program does access the old DEB table directly.

Other Migration Considerations
Data sets accessed by the media manager do not have a DEB data area. Therefore, they are not impacted by the maximum size of the DEB Table and are not affected by this change.

Where to Find More Information

- *DFSMS/MVS General Information*
- *DFSMS/MVS DFSMSdfp Advanced Services*

DFSMSdss Logical Release Support

Logical release processing is now available along with physical release processing. Logical release does not require input volumes to be specified. This type of processing can be used for VSAM extended-format data sets. Both logical and physical release can be used to process sequential or partitioned data sets.

Functional Components Enhanced
DFSMSdss

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence issues for this support.

Commands or Control Statements Modified
The DFSMSdss command RELEASE has been modified to add optional keywords LOGDDNAME, LOGDYNAM, INCAT, ONLYINCAT, SPHERE, and SELECTMULTI.

Where to Find More Information

- *DFSMS/MVS DFSMSdss Storage Administration Reference*

DFSMSdss SnapShot Support

SnapShot is a function of the RAMAC Virtual Array (RVA) which allows you to make a very quick copy of a set of tracks (an entire volume, a data set, or just a random set of tracks). The copy operation completes with only a few I/O requests to the device.

DFSMSdss offers two levels of SnapShot support:

Native SnapShot: Data is quickly copied, or “snapped,” directly from the source location to the target location. This capture occurs when you issue a DFSMSdss COPY command to copy volumes, tracks, or data sets from one DASD volume to another. DFSMSdss uses this method whenever the source and target data are on like devices in the same partition on the same RVA subsystem. No reblocking is required and a utility is not required. DFSMSdss can use this method whether or not the CONCURRENT keyword is specified. With native

SnapShot, the copy of the data is logically and physically complete as soon as the snap is complete.

CC-compatible SnapShot: Data is “snapped” from the source location to an intermediate location and then gradually copied to the target location using normal I/O methods. This capture occurs when you issue either the DFSMSdss COPY or DUMP command and specify the CONCURRENT keyword. As the name implies, this method operates in a fashion almost identical to existing concurrent copy (CC) operations. All DFSMSdss users and callers of the DFSMSdss API (such as DFSMShsm, DB2, and IMS) can continue to use the CONCURRENT keyword and receive functionally-identical CC support, but on a wider range of devices. Using CC-compatible SnapShot, the copy or dump of the data is logically complete after the source data is snapped to the intermediate location, and then physically complete after the data has been moved to the target media. It is also possible to perform concurrent copy on VM minivolumes using CC-compatible SnapShot support. CC-compatible SnapShot is also known as “virtual concurrent copy.”

Functional Components Enhanced
DFSMSdss, DFSMSdfp

Software Dependencies
This support requires IXFP 2.1 with a minimum PTF level of L170019.

Hardware Dependencies
The SnapShot function requires SnapShot-capable storage devices. For RAMAC Virtual Array, microcode must be at 4.3.26 or higher, and must have SnapShot enablement feature 6001.

Coexistence Issues
There are no coexistence issues related to this support.

Where to Find More Information

- *DFSMS/MVS DFSMSdss Storage Administration Reference*
- *DFSMS/MVS DFSMSdss Storage Administration Guide*
- *Implementing DFSMSdss SnapShot and Virtual Concurrent Copy (Redbook)*

DFSMSdss Storage Group Filtering

The DFSMSdss STORGRP parameter is a new source-volume-group keyword that allows selection of all volumes in a storage group, with a single parameter. The new volume selection keyword can be specified during logical data set dump, data set copy, and logical release. The STORGRP keyword with a storage group name is the functional equivalent of the LOGINDYNAM keyword (LOGDYNAM for RELEASE) listing all online volumes in the storage group.

Functional Components Enhanced
DFSMSdss

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence issues for this support.

Commands or Control Statements Modified

The DFSMSdss commands COPY, DUMP, and RELEASE have been modified to add the new optional keyword STORGRP.

Where to Find More Information

- *DFSMS/MVS DFSMSdss Storage Administration Reference*

DFSMShsm Secondary Host Promotion

Using the Cross-System Coupling Facility (XCF), which is available in both basic and parallel sysplex environments, DFSMShsm enables a secondary host to take over the unique functions performed by a failed primary host. This failure can be either an address space failure or an entire MVS image failure. Support is also provided to enable a DFSMShsm host to take over secondary space management (SSM) from a failed host which can be either the primary or a secondary host.

Functional Components Enhanced

DFSMShsm

Software Dependencies

XCF must be configured and running in Multisystem mode.

Hardware Dependencies

There are no additional hardware dependencies.

Coexistence Issues

There are no coexistence issues.

Performance Characteristics

Typically an address space or processor failure is detected and corrected within 12 hours. DFSMShsm's goal is that secondary host promotion occurs within minutes of the time of failure of the primary or SSM host.

Commands or Control Statements Modified

Two parameters have been added to the DFSMShsm SETSYS command. The first new parameter is PLEXNAME(hsmplex_suffix). This parameter allows you to name multiple HSMplexes within a sysplex. This parameter is used during the startup of DFSMShsm and must be specified if there are multiple HSMplexes in a sysplex because this HSMplex name is used to distinguish between the separate HSMplexes. The prefix of ARC is used by DFSMShsm and you can specify a maximum of five characters as the suffix for the HSMplex name. The default suffix used by DFSMShsm is PLEX0.

The second new parameter is PROMOTE(PRIMARYHOST(Y|N) SSM(Y|N)). This parameter allows the user to specify which hosts in an HSMplex can be promoted and which cannot. Currently, one host in an HSMplex is designated as the primary host and all other hosts are considered secondary. You should specify the PROMOTE parameter on only those hosts that are actually eligible for promotion. Order is not assigned for which host takes over for a failed host.

The PROMOTE parameter has been added to the DFSMShsm STOP command. This parameter is used to request that this host in a HSMplex is eligible to have its primary functions or secondary space management functions taken over by another host.

Migration Considerations

The following information applies to configuring XCF for DFSMShsm:

- There is one DFSMShsm XCF group per HSMplex. The XCF group name is the HSMplex name. The default HSMplex name is ARCPLEX0.
- There is one XCF group member for each DFSMShsm host in the HSMplex.
- DFSMShsm does not use the XCF messaging facilities.

Where to Find More Information

- *OS/390 MVS Setting Up a Sysplex*
- *OS/390 MVS Sysplex Services Guide*
- *OS/390 MVS Sysplex Services Reference*
- *OS/390 MVS JES Common Coupling Services*
- *OS/390 MVS System Commands*
- *DFSMS/MVS DFSMShsm Storage Administration Guide*
- *DFSMS/MVS DFSMShsm Implementation and Customization Guide*
- *DFSMS/MVS DFSMShsm Storage Administration Reference*

DFSMShsm Control Data Set Extended Addressability

With the possibility of combining even more HSMplexes into a single HSMplex, it becomes more likely that the CDS clusters will grow beyond 4GB. Currently, the MCDS and BCDS can increase up to 16GB but the OCDS is limited to 4GB. The MCDS and BCDS can achieve 16GB when they are comprised of four 4GB clusters.

DFSMShsm now supports VSAM KSDS extended addressability (EA) using record-level sharing (RLS) access mode for its control data sets. This allows the MCDS and BCDS to exceed the 16GB limitation and the OCDS to exceed the 4GB limitation. DFSMShsm does not support EA control data sets in CDSQ or CDSR serialization modes.

Since keyrange values are not allowed for EA KSDS data sets, DFSMShsm supports EA multicluster configurations with implicit keyranges. Multicluster support allows the MCDS and BCDS to be represented by up to 4 KSDS EA data sets. Having separate clusters reduces backup processing time by allowing parallel operations. Potential recovery time is also reduced when only one of the clusters is damaged or lost and needs to be forward recovered.

Functional Components Enhanced

DFSMShsm

Software Dependencies

The use of DFSMShsm EA CDS data sets requires that the control data sets are defined as EA and then accessed in RLS mode.

Hardware Dependencies

The use of DFSMShsm EA requires a coupling facility.

Coexistence Issues

In order to use EA control data sets, all DFSMShsm systems sharing the same control data sets must be at DFSMS/MVS 1.5. DFSMShsm 1.5 can coexist with

DFSMShsm 1.4 when toleration maintenance is applied. The toleration APAR addresses the support of EA control data sets that are accessed with RLS.

APAR Reference

OW33226

Performance Characteristics

There are no additional performance characteristics.

Processing Restrictions

With this new support, extended format (EF) data sets, EA or not, are accepted when CDSSHR=RLS is specified as a startup procedure keyword. Mixing EF and non-EF clusters is permissible since each is treated as a separate entity. However, if any cluster is accessed in RLS mode, all clusters must be accessed in RLS mode. This RLS requirement was established by DFSMShsm V1R4.

Other Migration Considerations

Migration to DFSMShsm 1.5 requires that all DFSMShsm systems sharing the same EA control data sets accessed with RLS, move to DFSMShsm 1.5 or they must be DFSMShsm 1.4 with the coexistence APAR applied. All DFSMShsm systems in the sysplex need to be shutdown while the control data sets are REPRO'd to new EA control data sets accessed with RLS and then all DFSMShsm systems restarted.

Where to Find More Information

- *DFSMS/MVS DFSMShsm Storage Administration Guide*
- *DFSMS/MVS DFSMShsm Implementation and Customization Guide*

DFSMShsm Overhead Reduction

During volume migration or volume backup, one task manages the volume and a different task processes data sets. This data set control task performs setup, data movement, and post processing. The setup and post processing are considered overhead.

With this enhancement, DFSMShsm overhead is overlapped with data movement for volume backup functions, thus improving performance (throughput) for the data stream to the output device.

Functional Components Enhanced

DFSMShsm

Software Dependencies

There are no additional software dependencies.

Hardware Dependencies

There are no additional hardware dependencies.

Coexistence Issues

There are no specific coexistence considerations introduced by this enhancement.

Performance Characteristics

For volume backup processing, this enhancement produces a better throughput to the output device when the SETSYS USERDATASETSERIALIZATION option is used. The amount of improvement depends on the size of the data set, the order of data sets processed, and the configuration.

Where to Find More Information

- *DFSMS/MVS DFSMShsm Storage Administration Guide*
- *DFSMS/MVS DFSMShsm Implementation and Customization Guide*

DFSMShsm GRSplex Serialization

During migration to a parallel sysplex, you can place multiple HSMplexes into a single GRSplex. An HSMplex consists of one or more processors running DFSMShsm that share common control data sets and a journal.

Previously, if two HSMplexes existed within a single GRSplex environment, one HSMplex interfered with the other HSMplex. Although the HSMplexes had unique resources, each HSMplex used the same resource names for global serialization, thus interfering with each other. To prevent this interference within a single GRSplex, you must change DFSMShsm serialization methods.

Note: All DFSMShsm systems sharing the same control data sets must use the same serialization method.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues

All hosts within an HSMplex MUST be using the same serialization technique. Any time DFSMShsm detects an inconsistency in the method of serialization, the detecting DFSMShsm shuts down immediately.

Consider the coexistence requirements for each of the following HSMplex scenarios:

1. All DFSMShsm hosts within one HSMplex are running at DFSMS/MVS 1.5.

   All DFSMShsm hosts must use the same serialization method. If not, at least one of the hosts will shut down (that is, each host detecting a mismatch will shut down). Therefore, all hosts within the HSMplex must specify RNAMEDSN=Y to use the new serialization method. Or, they must all specify RNAMEDSN=N (also the default) to continue using the old serialization method.

2. Not all DFSMShsm hosts within one HSMplex are running at DFSMS/MVS 1.5.

   If an HSMplex has both 1.5 and pre-1.5 running concurrently, then the 1.5 hosts cannot specify RNAMEDSN=Y. If RNAMEDSN=Y is specified, hosts that detect the mismatched serialization method will shut down.

3. Two or more HSMplexes are running concurrently.

   Each HSMplex using an old serialization method will interfere with other HSMplexes using the old method. HSMplexes using the new serialization method will not interfere with other HSMplexes. However, in a two HSMplex environment, one can use the old method and the other the new method; neither will interfere with the other.

Toleration maintenance is available.

APAR Reference

OW32531/OW32532, OW34475/OW34476

Performance Characteristics

There are no additional performance characteristics. Users can avoid one HSMplex affecting another by selecting different time windows for periodic functions such that no overlaps occur. This may not be possible or practical in all cases, and so this enhancement will resolve conflicts between HSMplexes with overlapping processing windows.

Commands or Control Statements Modified

A new keyword has been added to help you convert your systems to this new serialization method. This keyword, RNAMEDSN=Y | N, can be specified in the startup procedure and has the following values:

- The value Y means: serialization which includes the data set names of the control data sets and journal in the resource names.
- The value N means: continue to use the old method of serialization. In a sysplex with multiple HSMplexes, one HSMplex will interfere with another. This option provides compatibility with down-level releases.

The default value is N.

Processing Restrictions

All hosts within an HSMplex MUST be using the same serialization technique. Any time DFSMShsm detects an inconsistency in the method of serialization, the detecting DFSMShsm shuts down immediately.

Where to Find More Information

- *DFSMS/MVS DFSMShsm Implementation and Customization Guide*

DFSMShsm General Function and Usability Improvements

This section provides a summary of the general improvements to DFSMShsm functions. Changes include:

- **Improved filling of ML2 Partial Tapes:** Utilizing the complete capacity of a large tape is not a simple matter especially when you need to recall data already written to a tape before DFSMShsm completes its continuous writing to that tape. Another problem associated with a computing environment that uses a single storage group spanning several libraries, is that you can have too many ML2 partials. DFSMShsm has made significant changes that increase the average amount of data placed on your ML2 tapes. These changes result in the use of fewer tapes and in recycling fewer tapes than when DFSMShsm is running with the mark-full-on-take-away capability added with APAR OW23604. This new enhancement is accomplished without any degradation to recall performance and includes some new options for the LIST command.

  DFSMShsm has added one new subparameter to the LIST SELECT command: ASSOCIATED or NOTASSOCIATED. With this subparameter, you can specify a listing of only those ML2 partial tape volumes currently associated or not associated for output by specific migration or recycle tasks.

DFSMShsm has also added one new subparameter to the LIBRARY and NOLIBRARY parameters of the LIST SELECT command: ALTERNATE. With the ALTERNATE subparameter, you can list only the tape volumes having their alternate tape volumes in a library or not in a library.

A new ML2PARTIALSNOTASSOCIATEDGOAL parameter has been added to the SETSYS command. This parameter allows you to specify the maximum number of allowed partial ML2 tapes beyond those currently associated with a specific ML2 output task. The rest of the partial ML2 tapes will now be eligible for recycle.

- **Tape Conversion to 3590s:** The expected way to convert DFSMShsm migration and backup tapes to new technologies is with the DFSMShsm generic RECYCLE command. As an aid to processing only the old technology cartridges, DFSMShsm has added a new SELECT parameter to the RECYCLE command. The SELECT parameter has two subparameters, INCLUDE and EXCLUDE. You can now include or exclude ranges of tape volumes.

- **Volume Dump Stacking:** This improvement adds the capability to stack a specified number of DFSMShsm invoked DFSMSdss physical volume dumps to a single cartridge (or stream of cartridges) as separate files during the auto dump processing. A single MAGSTAR cartridge will hold the dumps on the order of 14 different 3390–3 DASD volumes.

  DFSMShsm still retains the same ease of use for volume and/or data set recovers and cartridge management. The stacking value is specified in the dump class provided by DFSMShsm. The capability to concurrently dump to multiple output streams is retained with each stream capable of having its own stack value. During recovers from dump copies, the high speed positioning with the locate record order is used to position to the beginning of the appropriate dump copy.

  DFSMShsm has added a new STACK subparameter to the DEFINE DUMPCLASS command. This subparameter is used to specify to DFSMShsm how many dump copies should be placed on a dump volume assigned to this dump class during one invocation of automatic dump.

  A new subparameter has been added to the DUMPCONTENTS subparameter of the LIST DUMPVOLUME command. This *srcvol* subparameter allows you to specify which dump contents are desired in the LIST output. The LIST DUMPCLASS outputs have been changed to include a new STACK field.

  DFSMShsm has also added a new SOURCEVOLUME(volser) to the RECOVER DUMPVOLUME command. This subparameter allows you to identify which dump copy on a dump volume should be used to restore a data set.

- **Recall Take-away from Recall:** DFSMShsm has an internal tape mount optimization capability that minimizes the number of tape mounts when there are multiple requests on one host to use the same tape. However, if this list of data sets to be processed gets excessively long it can prevent service to requests on other hosts needing the same tape or for requests on the same host needing a different tape.

  This problem could be addressed by an operator if it was brought to his or her attention. This enhancement enables the storage administrator to set a response time so that after a tape has been in continuous use by recall for that amount of time, DFSMShsm detects the existence of higher priority requests and conditionally terminates the mount optimization process. This makes the tape and the recall task available for doing higher priority work.

DFSMShsm has added a new TAPERECALLLIMITS parameter to the SETSYS command. The TAPERECALLLIMITS parameter has two subparameters, TASK and TAPE. With these subparameters you can specify limits to continuously mounted ML2 tapes during multiple recalls.

- **Remove Reel-type Tape Support:** DFSMShsm has discontinued the use of reel-type tapes for migration and backup output. Consequently, the following tape units are no longer supported for the creation of backup or migration tapes: 3400–3, 3400–4, 3400–5, 3400–6, and 3400–9.

  The following notes relate to the removal of reel-type tape support:

  – 3400–9 are cartridge-type tapes that emulate reel-type tapes and are no longer supported for backup or migration output.

  – Multi-file tape format is no longer supported. This format was used for reel-type tapes.

  Note: Many DFSMShsm commands have defaults that are used when a parameter is not specified. **All unit type defaults have been changed to 3590–1.**

  Data sets that are already backed up or migrated to reel-type tapes can still be recovered or recalled from these tapes. Reel-type tapes can be recycled, but the new tapes created will only be cartridge-type tapes.

  CDS backup and dump still support reel-type tapes as output. ABARS still supports reel-type tapes as output (including ARECOVER to ML2) as well as supporting reel-type user tapes.

- **VSAM Extended Format Data Set Support:** DFSMShsm can be used to process any of the new VSAM extended format data set organizations. See "VSAM Extended Format Data Set Enhancements" on page 72 for more information.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no additional coexistence considerations introduced by these enhancements.

Where to Find More Information

- *DFSMS/MVS DFSMShsm Storage Administration Guide*
- *DFSMS/MVS DFSMShsm Implementation and Customization Guide*

DFSMSrmm Application Programming Interface

DFSMSrmm provides an application programming interface that can be used to obtain information about resources defined to DFSMSrmm. Using the DFSMSrmm API, you can read, extract, and update data in the DFSMSrmm control data set. You can use the data to create reports or implement automation.

Functional Components Enhanced
DFSMSrmm

Software Dependencies
High level assembler language is required to use the DFSMSrmm application interface. See “Program Products Used for Installation” on page 13 for the level of high level assembler language required.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Programming Interfaces Affected
These structured field introducers have been added or changed:

- CATS — CATSYSID value
- CSDT — Catalog synchronize date
- CSTM — Catalog synchronize time
- CTLG — Catalog status
- DSPD — Disposition DD name
- DSPM — Disposition message prefix
- FCSP — Catalog synchronize in progress
- KEYF — Key from
- KEYT — Key to
- LVC — Current label version
- LVN — Required label version
- TYPF — Type from
- TYPT — Type to

Where to Find More Information

- *DFSMS/MVS DFSMSrmm Application Programming Interface*
- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*

DFSMSrmm Catalog Status Tracking

DFSMSrmm inventory management has been improved by dynamically tracking changes in catalog status of tape data sets. This removes the need to check catalog status of all recorded data sets other than on an exception basis. As long as the DFSMSrmm control data set is shared and accessible, this allows DFSMSrmm environments where catalogs are not shared. DFSMSrmm provides an option where you can identify a set of shared catalogs. Then you can run DFSMSrmm inventory management once to synchronize the set of catalogs you identified. DFSMSrmm also provides a utility you can use to synchronize DFSMSrmm with user catalogs which has the same capability to support both shared catalog or non-shared catalog environments. Run the utility periodically as a safeguard or during recovery activities.

Functional Components Enhanced
DFSMSrmm

Software Dependencies
DFSMSrmm catalog status tracking only supports ICF catalogs. Also all systems sharing the DFSMSrmm control data set must be at the 1.5.0 level before you can specify the DFSMSrmm EDGRMMxx parmlib member CATSYSID operand.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

System Interfaces Affected
You must enable this function by specifying the DFSMSrmm EDGRMMxx parmlib member OPTION command CATSYSID option.

Programming Interfaces Affected
The following DFSMSrmm macros have been modified for this function: EDGSDREC and EDGACTRC.

The DFSMSrmm ISPF panels have been modified for this function.

The CATSYNCH parameter is been added to the EXEC parameters for the DFSMSrmm EDGHSKP utility.

The CATSYNCH operand has been added to the DFSMSrmm EDGUTIL utility SYSIN CONTROL statement.

The CATSYSID operand has been added for the DFSMSrmm EDGRMMxx parmlib member OPTION command.

Processing Restrictions
When catalogs are not fully shared, you might have data sets with the same name or GDGs with the same base name that are cataloged in different, unshared catalogs. If you specify catalog retention to retain these data sets, DFSMSrmm retains these data sets without any problems. If, however, you combine cycle retention and catalog control, you can get unexpected results because DFSMSrmm does no special processing for this case. DFSMSrmm uses normal cycles processing which might retain a mix of data sets from both catalogs.

For DFSMSrmm to group cycles correctly we recommend that you use the jobname to further qualify the data sets, using a different jobname for each 'set' of cataloged data sets. For example, use JOBNAME(A) and JOBNAME(B) to differentiate between data sets created on system A and system B.

Where to Find More Information

- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*

DFSMSrmm Disposition Control

DFSMSrmm now supports use of a special dd statement to point to a data set containing special handling information needed by the tape operator. This can be used to identify tapes needed for further processing or for immediate removal to a disaster recovery site. It can also be used to create sticky labels containing data set and volume information.

Functional Components Enhanced
DFSMSrmm

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Programming Interfaces Affected
The DFSMSrmm EDGUX100 installation exit and parameter list has been updated with new fields.

The DFSMSrmm ISPF dialog panels have been modified for this function.

The DISPDDNAME and DISPMSGID operands have been added to the DFSMSrmm EDGRMMxx parmlib member OPTION command.

The EDGSLAB macro has been added to DFSMSrmm.

Other Migration Considerations

You might need to modify your version of the EDGUX100 installation exit to use some of the functions provided by this new function. If you plan on updating EDGUX100 to use it on older software versions, be sure to check the parameter list version number before using fields that are only present in a higher version. If you want to continue to use your current exit, you can do so without reassembling the exit if your version of the exit only validates the parameter list header name.

Where to Find More Information

- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*

DFSMSrmm Subsystem Enhancements

DFSMSrmm has been enhanced to improve the handling of requests when DFSMSrmm is stopped, modified, or quiesced. When DFSMSrmm is stopped, outstanding requests are completed. When DFSMSrmm is modified or quiesced, only current requests are completed and outstanding requests are processed only after the modify or quiesce is completed.

Functional Components Enhanced
DFSMSrmm

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Programming Interfaces Affected
DFSMSrmm issues WTOR EDG1107D REQUESTS WAIT TO BE PROCESSED = REPLY "STOP", "QUIESCE", "RESTART", OR "M=xx"

Other Migration Considerations
None.

Where to Find More Information

- *DFSMS/MVS DFSMSrmm Application Programming Interface*
- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*
- *OS/390 MVS System Messages, Vol 2 (ASB-EWX)*

DFSMSrmm TSO Subcommand Enhancements

DFSMSrmm provides new operands for DFSMSrmm TSO subcommands so you can change data set name and volume serial number recorded in the DFSMSrmm control data set.

Functional Components Enhanced
DFSMSrmm

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Commands or Control Statements Modified
The following DFSMSrmm TSO subcommands have been modified.

- CHANGEDATASET — The NEWDSNAME operand has been added so you can change the name of a data set in the DFSMSrmm control data set.
- CHANGEVOLUME — The NEWVOLUME operand has been added so you can change the volume serial number of a volume already defined in the DFSMSrmm control data set.
- SEARCHDATASET — The CHAIN operand has been added so you can search for all data set information for all physical files in the same multi-volume data set.
- SEARCHVOLUME — The CHAIN operand has been added so you can search for all volumes in the same multi-volume set.

Where to Find More Information

- *DFSMS/MVS DFSMSrmm Application Programming Interface*
- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*

DFSMSrmm Vital Record Specification Enhancements

The DFSMSrmm vital record specification enhancements improve the way DFSMSrmm processes the policies which control expiration and vaulting characteristics of a tape data set. These enhancements allow expiration date and vaulting specifications to be assigned through separate policies or a single policy, as the

situation requires. These enhancements increase the flexibility of tape management policy definitions and simplify the migration from other solutions to DFSMSrmm.

Here is a summary of the capabilities available in the DFSMSrmm vital record specification enhancements.

- Data sets can be managed using more than one retention policy.
- All the copies of a data set created on the same day can be retained as a single cycle.
- Extra days can be defined so data sets can be retained beyond the date specified in the DFSMSrmm retention policy for the data sets.
- Volumes can remain in their current locations so that it is not necessary to know where volumes are located.
- Data sets or volumes can be managed using a combination of retention and movement policies.
- Enhanced DFSMSrmm inventory management processing allows the release of volumes and their return to scratch status in one run of inventory management by using the RELEASE(SCRATCHIMMEDIATE) operand on the RMM ADDVRS subcommand. The operand is honored for volumes that have no release actions pending other than return to scratch.
- VRS policies can control total retention for a data set by overriding the volume expiration date that is used to retain a data set when the data set is no longer VRS retained.

Functional Components Enhanced
DFSMSrmm

Software Dependencies
None.

Hardware Dependencies
None.

Coexistence Issues
A compatibility PTF is required.

APAR Reference

OW28155

Programming Interfaces Affected
New messages, return codes, and reason codes have been added.

The DFSMSrmm EDGHSKP inventory management utility has been changed.

The DFSMSrmm ISPF dialog has been changed.

Commands or Control Statements Modified
The following DFSMSrmm TSO subcommands been modified: ADDBIN, ADDDATASET, ADDRACK, ADDVOLUME, ADDVRS, DELETEBIN, DELETERACK, and SEARCHVRS.

Where to Find More Information

- *DFSMS/MVS DFSMSrmm Application Programming Interface*
- *DFSMS/MVS DFSMSrmm Diagnosis Guide*

- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*
- *DFSMS/MVS DFSMSrmm Command Reference Summary*
- *OS/390 MVS System Messages, Vol 2 (ASB-EWX)*

Extended Remote Copy Enhancements

The extended remote copy (XRC) suspend/resume function provides support for planned outages. A planned outage is an event that can be scheduled in advance and has a known duration. Examples include physical configuration changes and disruptive updates to system code.

For more information about using this disaster recovery and workload migration solution, see “Extended Remote Copy Support” on page 134.

Functional Components Enhanced
DFSMSdss, DFSMSdfp

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence issues related to this support.

Command Changes
Two new XRC commands, XSET and XSUSPEND, allow you to modify session characteristics and suspend a session, respectively. Parameters are added to new and existing commands to allow control over the timing of delete, suspend, and end functions.

A new PPRC command, CGROUP, gives you an easy way to control all PPRC activity on any storage control.

Where to Find More Information

- *DFSMS/MVS Remote Copy Administrator's Guide and Reference*

Hierarchical File System Performance and Other Enhancements

The hierarchical file system (HFS) includes the following functional enhancements:

- Performance is improved. In conjunction with OS/390 UNIX System Services, DFSMS/MVS 1.5 includes fundamental changes in the way it writes HFS data to disk. Multiple updates are now batched into a single I/O operation, reducing both I/O and path length and greatly decreasing file access contention. HFS now maintains its own buffer pools and also takes advantage of caching for file data, file attributes, and HFS indexes.
- File system integrity is improved in a multisystem shared environment. Integrity of volume mounts is assured with a new serialization protocol used when HFS data sets are shared among systems in a sysplex.
- HFS data sets can now span multiple volumes.

Functional Components Enhanced

DFSMSdfp, DFSMSdss

Software Dependencies

OS/390 Release 5, or higher, is required for installations to realize the HFS performance benefits associated with these changes.

A new FMID is included on the DFSMS/MVS product tape to install DFSMS/USS—an extension to the OS/390 Unix System Services provided for this new support. A separate FMID is required in this case because it must be applied to the system as part of Wave 2 of the OS/390 installation rather than in Wave 1, which includes only the basic elements. After DFSMS/USS is installed, no additional user action is required, as the function becomes integrated with OS/390 UNIX System Services.

Maintenance is needed for DFSMSdss 1.5 to dump and restore HFS data sets.

APAR Reference

OW35716

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

Previous releases of DFSMS/MVS cannot mount HFS data sets that span more than one volume. Toleration PTFs must be installed on lower-level systems that share HFS data with DFSMS/MVS 1.5 systems to ensure the integrity of multi-volume HFS data sets.

Single volume HFS data sets created on a DFSMS/MVS 1.5 system are still accessible on previous releases.

APAR Reference

OW30999, OW31002, OW34335, OW34343

Processing Restrictions

A colony address space is one that OS/390 UNIX starts in order to run certain types of physical file systems. This is indicated by the presence of the ASNAME() parameter on the FILESYSTYPE statement that defines the PFS. It is rare for HFS to get started in a colony address space. You should be aware, however, that with the changes introduced by this support, HFS should not be allowed to run in a colony address space.

Command or Control Statements Modified

Users can extend the HFS using the new OS/390 UNIX System Services shell command *confighfs*.

Migration Considerations

Refer to *OS/390 UNIX System Services Planning* and *OS/390 Planning for Installation*for migration planning information related to these HFS changes.

Where to Find More Information

- *OS/390 UNIX System Services Planning*
- *OS/390 UNIX System Services File System Interface Reference*
- *OS/390 UNIX System Services Command Reference*
- *OS/390 Planning for Installation*

- *OS/390 Introduction and Release Guide*

ISO/ANSI Version 4 Tape Support

ISO and ANSI Version 4 tape labels are now supported for data interchange with non-S/390 platforms. Version 4 tape labels follow industry standards for ISO 1001-1986(E) and ANSI X3.27-1987, enabling data to be written in any national or user-defined character standard supported by the conversion utility. The 2K block size limit for data has been removed with this support.

Support continues for users reading Version 1 tape volumes (following ISO 1001-1967 and ANSI X3.27-1969 standards) and users reading or creating Version 3 tape volumes (ISO 1001-1979, ANSI X.27-1978, and FIPS 79 standards).

You can specify a system-wide default for ISO/ANSI labeling with the new ALVERSION parameter in SYS1.PARMLIB member DEVSUPxx. You can also override this default with labeling utilities IEHINITT or DFSMSrmm EDGINERS.

As with Version 3 tapes, Version 4 labels must be in 7-bit American Standard Code for Information Interchange (ASCII); however, the actual data can be written in any code page. For ease of migration, the data written to Version 4 tapes will continue to be translated to ASCII; however, you can choose to convert the data to any of the supported code pages.

You can request a national or user-defined character standard with the new JCL parameter, CCSID, on a DD, EXEC or JOB statement. During output processing, the data is converted from the JOB/EXEC statement CCSID, where the system default is a CCSID of 500 (International EBCDIC) to the CCSID specified on the DD statement, where the system default is a CCSID of 367 (ANSI X3.4 ASCII). The CCSID is recorded in the HDR2 tape label and is used for conversion on read operations. A CCSID of 65535 has a special meaning and results in bypassing conversion.

Functional Components Enhanced
DFSMSdfp, DFSMSrmm

Software Dependencies
There are no specific software requirements to exploit the basic ISO/ANSI Version 4 support; however, if you plan to use BSAM or QSAM to write data in formats other then ASCII or EBCDIC, OS/390 Release 5 (or higher) is required to do the conversion. OS/390 maintenance is required if you intend to use the new JCL CCSID parameter.

APAR Reference

OW32524

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There is no requirement to have all systems in a sysplex at DFSMS/MVS 1.5 or to have the same system-wide ALVERSION specified for all DFSMS/MVS 1.5 systems in a sysplex. You can have a default of Version 3 on one system and Version 4 on another system. Refer to “Other Migration Considerations” for a

suggested method of converting from Version 3 to Version 4 tape processing across your installation.

Toleration PTFs must be applied to any lower level systems that share data with a DFSMS/MVS 1.5 systems if the new ALVERSION parameter is specified in SYS1.PARMLIB member DEVSUPxx, and the DEVSUPxx member is shared among the systems of mixed releases. Toleration PTFs allow the lower level systems to ignore the ALVERSION parameter.

Toleration PTFs are also needed if Version 4 tapes created on a DFSMS/MVS 1.5 system are read by the lower-level systems. The Version 4 tapes must remain in ASCII format. If the data is converted to any other code page, unpredictable results will occur on the lower-level systems.

You should take precautions to ensure that a lower level system does not convert Version 4 tapes back to Version 3 by either modifying the system's associated tape or installation exits or by preventing Version 4 tapes from being mounted on that system.

APAR Reference

OW31927/OW31944, OW26078

System Interfaces Affected

You can update SYS1.PARMLIB member DEVSUPxx to include the ALVERSION parameter to specify a system-wide default for the tape labeling version and to indicate whether the version level should be forced when tape labels are rewritten. Version 3 is the IBM-supplied default if DEVSUPxx is not updated.

Independent of what conversion is done for the user data, tape labels are always created in ASCII, and are translated through the IBM-supplied translation routine IGC0010C. As with Version 3 tape processing, you can replace IGC0010C with an installation-written routine to translate ISCII/ASCII 8-bit code.

Programming Interfaces Affected

No changes to existing applications are required; however, if you want to override the system-wide default for the labeling version used, you need to update the appropriate applications.

You can take advantage of the new options available through DFSMSrmm EDGINERS or the IEHINITT utility to specify whether tapes should be initialized as Version 3 or Version 4. This could require a modification to existing jobs.

All currently-supported tape exits and installation exits that get control for Version 3 tapes also get control for Version 4 tapes. You can override the system-defined version by changing the volume mount exit. By modifying the label anomaly exit, you can reject volumes that have a version discrepancy between what is currently used on the tape and what will be written.

The following general-use programming macros are affected by these changes:

IFGTEP label exit parameter list: The volume mount (VOLMT) section is changed to include an option to override the version number specified in the DEVSUPxx PARMLIB member. The label anomaly (LABAN) section is changed to indicate if there is a discrepancy between the current version of a tape and the version in which the tape will be written.

IEZDEB: User application and tape CCSIDs are saved in the second DEB extension.

IFGSMF14: User, tape, and label CCSID values and other information are saved in a new ISO/ANSI Version 4 CCSID Extended Information Section defined in SMF record types 14 and 15.

If you have any OEM applications, tape exits, or installations exits tailored to process only ISO/ANSI Version 3 tape labels, the code might require an update to support ISO/ANSI Version 4 labels. Changes might include modifying an installation exit to support an accessibility value that is acceptable to Version 4 but not acceptable to Version 3; or modifying an exit to specify a labeling version other than the system-wide default.

If you intend to use a character standard other than ASCII or EBCDIC, user applications could require changes to recognize the new format.

The DFSMSrmm ISPF dialog has been changed for ISO/ANSI support.

Commands or Control Statements Modified

If you have applications for which you need to specify a national or user-defined character set other than ASCII or EBCDIC, you need to modify the associated application JCL to include the new CCSID parameter on the DD, EXEC or JOB statement.

Processing Restrictions

Character set conversion is supported only when BSAM or QSAM are used and the conversion does not result in a change to the length of user data.

Other Migration Considerations

It is recommended that you take a staged approach to exploiting ISO/ANSI Version 4 tape support. Following is one method for migrating from Version 3 to Version 4 tape processing:

1. After installing DFSMS/MVS 1.5, either specify ALVERSION=3 in DEVSUPxx or let it default to Version 3. Allow individual users to label Version 4 tapes using IEHINITT or DFSMSrmm to test Version 4 processing until results are satisfactory.
2. After consulting with the owners of tape applications, change the installation version level to Version 4 (ALVERSION=4). This allows current Version 3 tapes to remain unchanged, but will create new tapes as Version 4.
3. When there is no longer any need to process Version 3 tapes, or when it is determined that all receiving systems can process Version 4 tapes, change the DEVSUPxx member to force Version 4 (ALVERSION=FORCE4). This causes any remaining Version 3 tapes to be converted to Version 4 tapes.

Where to Find More Information

- *DFSMS/MVS General Information*
- *DFSMS/MVS Macro Instructions for Data Sets*
- *DFSMS/MVS Using Data Sets*
- *DFSMS/MVS Access Method Services for ICF*
- *DFSMS/MVS DFSMSdfp Advanced Services*
- *DFSMS/MVS Installation Exits*
- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*
- *DFSMS/MVS Utilities*

- *DFSMS/MVS Using Magnetic Tapes*
- *OS/390 MVS Initialization and Tuning Reference*
- *OS/390 MVS JCL Reference*
- *OS/390 MVS JCL User's Guide*

NaviQuest Batch Support

With DFSMS/MVS 1.3, NaviQuest allowed you to test SMS ACS routines in a batch environment. You could also define and alter a management class, pool storage group, or tape storage group; and add or delete a volume in a pool storage group in batch mode. The NaviQuest interface is accessible from an application program or a REXX EXEC and does not require hands-on operation as ISMF does.

NaviQuest is further enhanced in this release to provide batch support for additional storage administrator tasks related to setting up and maintaining an SMS configuration. Batch functions now include:

- SMS base configuration define, alter and display
- Data class define, alter and display
- Storage class define, alter and display
- Management class define, alter and display
- VIO, dummy, object, and object backup storage group define and alter
- Aggregate group define, alter, and display

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
Coexistence with ISMF/E is not feasible. ISMF/E was a program offering (5764-033) available through IBM branch offices. It was a predecessor to the first release of NaviQuest.

Installations using either ISMF/E or NaviQuest Version 1 Release 1 (5655-ACS, FIND HACS110) need to delete the earlier programs before applying this version.

ISMF Changes
NaviQuest runs under ISMF and is invoked as “Enhanced ACS Management” from the ISMF Primary Option Menu for storage administrators. This has not changed from the previous release; however, there are changes to the ISMF Primary Option Menu for end users to invoke the aggregate group application.

Control Statements Modified
The NaviQuest batch program interface includes new or modified EXECs and keywords for the following:

Data class DEFINE/ALTER/DISPLAY

Storage class DEFINE/ALTER/DISPLAY

Management class DEFINE/ALTER/DISPLAY

VIO storage group DEFINE/ALTER

POOL storage group VOLUME DEFINE/ALTER

Dummy storage group DEFINE/ALTER

Object storage group DEFINE/ALTER

Object backup storage group DEFINE/ALTER

Aggregate group DEFINE/ALTER/DISPLAY

Base configuration DEFINE/ALTER/DISPLAY

The functions that NaviQuest provides online are also available in a library of sample jobs which you can modify and submit to run on the system. These JCL members reside in the library SYS1.SACBCNTL which was introduced in NaviQuest Version 1 Release 1.

Other Migration Considerations

Users who defined test cases with earlier releases of NaviQuest might have filled in both lines of the test case description field. With DFSMS/MVS 1.5, the second line of the description field is replaced with the expected results after testing with NaviQuest.

Where to Find More Information

- *DFSMS/MVS NaviQuest User's Guide*
- *DFSMS/MVS DFSMSdfp Storage Administration Reference*
- *DFSMS/MVS Implementing System-Managed Storage*

OAM Parallel Sysplex Support

OAM supports the parallel sysplex environment. Each OAM running on an OS/390 system in a parallel sysplex, at the DFSMS/MVS 1.5.0 level, will be in communication with all other instances of OAM on other DFSMS/MVS systems in the sysplex, which belong to the same XCF group, using services of the cross-system coupling facility (XCF) component of OS/390.

Each instance of OAM in a single parallel sysplex, sharing the same database for OAM, must belong to the same XCF group and the DB2 subsystem must belong to the same DB2 data sharing group. This collection of address spaces in a sysplex is known as an OAMplex. An OAMplex consists of one or more instances of OAM, running on systems which are part of a parallel sysplex. Each instance of OAM in an OAMplex is connected to a subsystem, each of which belongs to the same DB2 data sharing group, so each OAM has the same DB2 database. Also, each instance of OAM in an OAMplex is a member of a single XCF group.

Any OAM object may be retrieved from any OS/390 system in a parallel sysplex regardless of which OAM in the sysplex stored the object or on which media (DASD, 3995 optical, or tape) the object resided on as long as each instance of OAM is part of the same OAMplex.

OAM no longer has a 100 object storage group restriction. It is now possible to define more than 100 object storage groups in your configuration, or if fewer than 100 object storage groups are needed, views are no longer required to define all 100 previously required.

Functional Components Enhanced
DFSMSdfp(OAM)

Software Dependencies
Cross-system coupling facility (XCF) software.

Hardware Dependencies
Cross-system coupling facility (XCF) hardware.

Coexistence Issues

APAR Reference

OW31930, OW34385

New Operator Command

- D SMS,OAMXCF

Operator Commands Modified

- VARY SMS,DRIVE
- VARY SMS,LIBRARY
- LIBRARY EJECT
- MODIFY OAM,LABEL
- MODIFY OAM,QUERY
- MODIFY OAM,START,MOVEVOL
- MODIFY OAM,STOP,MOVEVOL
- MODIFY OAM,START,LIBMGT
- DISPLAY SMS,OAM
- DISPLAY SMS,LIBRARY
- DISPLAY SMS,DRIVE
- DISPLAY SMS,VOLUME
- MODIFY OAM,UPDATE

Where to Find More Information

- *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support*

OAM Pseudo Library Concept

In addition to real optical libraries, OAM also supports the concept of pseudo optical libraries. With the support of parallel sysplex, the concept of pseudo libraries has changed from previous releases; however, pseudo libraries defined in previous releases are still supported. There are no longer restrictions on the number of pseudo optical libraries that can be defined in an SCDS. Also, there is no longer a device type associated with pseudo optical libraries.

A pseudo library can now be a collection of stand-alone or operator-accessible drives, or both, and shelf-resident volumes, defined by the installation and not necessarily of the same device and media type. Pseudo libraries can be defined without a device type. When operator-accessible drives are defined, they can be

assigned to a pseudo library chosen by the person defining the devices. These devices can be grouped in a manner that best fits the needs of the installation (for example, physical location, device and media affinity, backup and primary objects stored together, and so forth). Additionally, there is no limit on the number of pseudo libraries that can be defined within an active SMS configuration.

It is the responsibility of the installation to determine which pseudo library an optical volume is to be associated with. This determination is made either when the volume is ejected from a real library or when a volume is labeled on an operator-accessible drive.

Pseudo libraries defined prior to DFSMS/MVS 1.5.0 are still supported. These pseudo libraries were defined as a collection of one or more stand-alone or operator-accessible drives of the same device type, and one or more shelf-resident optical volumes of a like media type. OAM continues to honor these old pseudo library definitions. Support for both concepts allows installations the option of deciding which method would be more beneficial or efficient in their environments. It also allows installations wishing to convert their environments to the new pseudo library concept over a period of time, the ability to use their previously defined pseudo libraries during the transition period.

Functional Components Enhanced
DFSMSdfp(OAM)

Software Dependencies
No additional software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

ISMF Changes
The following ISMF panels are either new or have been modified for this support:

- 9246 Library Define (modified)
- 3995 Library Define (modified)
- Pseudo Library Define (new)
- 9246 Library Alter (modified)
- 3995 Library Alter (modified)
- Object Storage Group Define/Alter (modified)

Operator Commands Modified
There are no operator commands modified.

Where to Find More Information

- *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support*

OAM Support for the 3995-SW4 Optical Drive

OAM supports an new optical disk drive, the 3995-SW4, which is a drive device type for the 3995-Cxx libraries, both internal and external. This drive supports 8x-density WORM, rewritable, and CCW optical disk media. This media can support up to 5.2 gigabytes of data.

Functional Components Enhanced
DFSMSdfp(OAM)

Software Dependencies
No addtional software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
The 3995-SW4 drive cannot coexistence in a 3995-Cxx library with 3995-SW3 drives.

ISMF Changes

- 3995 Drive Define (modified)

Where to Find More Information

- *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support*

OAM New and Changed Operator Commands for Object Support

The following OAM operator commands are either new or changed from previous releases to support various functions.

Reformatting a 3995 Optical Disk

The OAM reformat utility (the TSO command OAMUTIL) is used to perform various tasks against a 3995 optical disk cartridge to reclaim usable space on the cartridge. This utility can be run either by running the CBRSAMUT SAMPLIB job or by issuing a TSO command to start the utility. This utility allows you to perform the following tasks:

- Reformat one or both sides of a 3995 optical disk cartridge
- Reformat and rename the volume serial number of one or both sides of an optical disk cartridge.
- Return the volumes back to the SCRATCH storage group upon completion of the REFORMAT utility (only when there is a request for both sides of the optical disk to be reformatted) to be used for subsequent write requests.

The reformat utility can be run regardless of whether the volume is inside or outside a 3995 optical library, the volume belongs to an OBJECT or an OBJECT BACKUP storage group, or the media is rewritable or WORM. Reformatting rewritable media reclaims the used space; however, reformatting WORM media cannot reclaim the used space.

Relabeling a 3995 Optical Disk Volume

The relabel command is used to allow the operator to rename a volume serial number on a previously defined 3995 optical disk volume. As a preventative measure to keep from losing all active primary or backup copies of objects on the 3995 disk volume that is being relabeled, the following conditions apply:

- The optical disk volume being relabeled cannot be an OBJECT BACKUP volume.
- There can be no active primary copy of an object on the volume.
- There can be no write request currently scheduled for the volume.
- In an OAMplex environment, the requested volume must be currently managed and controlled by the OAM on which the command was entered.

To erase all copies of objects on the 3995 optical disk volume, submit the OAM utility job to reformat the volume. See “Reformatting a 3995 Optical Disk” on page 64 for more information on this utility.

When the command is accepted, if the requested volume is shelf-resident and not currently mounted on the selected drive, OAM asks the operator to mount the requested volume on the selected optical drive. If the requested volume is library-resident, OAM mounts the volume on a library drive. Following successful completion of this processing, OAM then performs the following functions:

- deletes the row in the DB2 Volume Table row for the old volume serial number
- inserts a new row in the DB2 Volume Table for the new volume serial number
- updates the row of the opposite volume in the DB2 Volume Table with the new volume serial number
- issues message CBR4460 to inform the operator that the relabeling of the 3995 optical disk volume has completed

Displaying OAM XCF Status

This command displays system status for an instance of OAM in relation to the sysplex and XCF.

To display OAM XCF status, enter the following command:

```
DISPLAY SMS,OAMXCF
```

Displaying SETOPT and SETOAM Parameters

This command displays the current settings of the SETOPT and SETOAM parameters for the OAM address space.

The following is a sample of DISPLAY SMS,SETOAM,*storgrp* command:

```
F OAM,DISPLAY,SETOAM,GROUP22
```

The following is a sample of DISPLAY SMS,SETOAM,ALL command:

```
F OAM,DISPLAY,SETOAM,ALL
```

The following is a sample of DISPLAY SMS,SETOPT,*storgrp* command:

```
F OAM,DISPLAY,SETOPT,GROUP22
```

The following is a sample of DISPLAY SMS,SETOPT,ALL command:

```
F OAM,DISPLAY,SETOAM,ALL
```

Using the Update Command to Set SETOAM and SETOPT Values

Most of the current values of the SETOAM and SETOPT commands can be changed using the F OAM,UPDATE command without having to restart OAM. These commands can also define settings of most of the SETOAM and SETOPT values if a CBROAMxx PARMLIB member was not used at OAM startup. This command provides dynamic processing that provides another method (other than using the CBROAMxx PARMLIB member) for customizing your object tape and optical support. For more information, refer to *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support*.

The following are some examples of these various UPDATE commands and explanation of the modification:

```
F OAM,UPDATE,SETOAM,GROUP22,TAPEUNIT,TAPEESO
```

Assuming TAPEESO is a valid tape esoteric name defined to the system, this sets the TAPEUNITNAME setting for only GROUP22 to the tape esoteric, TAPEESO.

```
F OAM,UPDATE,SETOAM,ALL,TFULLPER,90,TFULLTHR,100
```

This updates all global values and every storage group name in the active SMS configuration to a TAPEPERCENTFULL value of 90 and a TAPEFULLTHRESHOLD value of 100.

```
F OAM,UPDATE,SETOAM,GROUP22,MWT,2,DMWT,4,TFULLPER,95
```

This updates the values for MOUNTWAITTIME, DEMOUNTWAITTIME, and TAPEFULLPERCENT for only the GROUP22 storage group.

```
F OAM,UPDATE,SETOPT,ALL,OPREINIT,OAMSCR,OPDISDLY,50
```

This updates the values for OPTICALREINITMODE and OPTICALDISPATCHERDELAY for all global values and for each storage group in the active SMS configuration.

```
F OAM,UPDATE,SETOPT,GROUP22,OPREINIT,GROUP,OPDISDLY,45
```

This updates the values for OPTICALREINITMODE and OPTICALDISPATCHERDELAY only for storage group GROUP22.

```
F OAM,UPDATE,SETOPT,GROUP22,OPREINIT,GROUP,OPREINIT,OAMSCR
```

This updates the values for OPTICALREINITMODE twice for storage group GROUP22. The first update takes place followed by the second. The final result is OAMSCR.

Functional Components Enhanced
DFSMSdfp(OAM)

Software Dependencies
There are no additional software requirements

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Where to Find More Information

- *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support*
- *OS/390 MVS System Messages, Vol 2 (ASB-EWX)*

Virtual Tape Server Import/Export Support

OAM provides the ability to physically import and export logical volumes within the virtual tape server (VTS) subsystem[3]. This enhancement to existing support includes managing the physical removal of the 3590 cartridges containing stacked logical volumes from a VTS and the corresponding function for entering these cartridges into a VTS. These functions require interactions with the host and the tape management system.

DFSMSrmm also provides support for import and export. You can define volumes that reside in a virtual tape server subsystem to DFSMSrmm; then use DFSMSrmm to manage the retention and movement of these volumes. Using DFSMSrmm TSO subcommands, you can keep your inventory accurate and up-to-date. You can also use the DFSMSrmm TSO subcommands or the DFSMSrmm application programming interface and the DFSMSrmm EDGRPTD utility to create reports that your operators can use to move volumes.

[3] This support is available with APARs OW36342 or OW36343 and OW36349 or OW36350. If you are planning to use the new pre-ACS exit routine, APAR OW36351 or OW36352 should be installed.

Also included in this support is a DFSMS/MVS pre-ACS interface routine that enables an external source, such as a tape management system, to provide input (through read-only variables) to the ACS routine to influence construct selection and assignment. The variables provided by the tape management systems can be used to direct allocations to a particular tape library, thereby coordinating vaulting runs for backups, off-site storage, and so forth.

Exporting Volumes

To remove a set of logical volumes from a VTS, the user or the tape management system must first perform an operation at the host to write the list of logical volumes to be exported, in pairs of volume serial number and destination, on a logical volume in the library. Two other files are also written on this export list volume. One is reserved for future use and the other is used by the library to record the export status of each logical volume listed in the export list. Once the export list is written, the LIBRARY EXPORT command or the CBRXLCS FUNC=EXPORT programming interface can be used to identify the export list volume and to initiate the export operation at the library. The DFSMSrmm SEARCHVOLUME command with the CLIST option can be used to create the list of volumes to be exported. For more information on DFSMSrmm, see “Preparing to Implement DFSMSrmm” on page 247.

The library mounts the export list volume and determines the list of volumes to export and the destination for the logical volumes. Logical volumes with the same destination are grouped together on the same stacked media. The library then begins to write the export logical volumes onto the stacked volumes. Each exported stacked volume contains a volume map identifying the contents of the stacked volume, logical volume data fragment files for each logical volume successfully copied, and the logical volumes themselves. A single export operation may result in many stacked volumes being exported with multiple volumes per destination.

As the library manager finishes with a stacked volume, the logical volumes are assigned to the exported category and are then processed by OAM. Volume export processing is similar to eject processing; however, in addition, the cartridge eject installation exit (CBRUXEJC) is passed the container volume serial number of the stacked cartridge on which the logical volume resides. The tape management system acceptance of the export is implicit (no opportunity to fail the export). Each logical volume is then purged from the library manager inventory and the host TCDB volume record is kept or purged depending on the disposition specified by the installation exit or defaults set for the library. At this time, DFSMSrmm records the volume serial number of the exported stacked volume as the container for each exported logical volume.

Once all logical volumes residing on a stacked cartridge have been successfully purged from the library manager inventory, the stacked volume is ejected from the library and the export operation continues at the library, filling another stacked volume for the destination currently being processed. For the export operation to continue with minimal host delays, it is important for a host that owns the TCDB records for the logical volumes being exported to have the OAM address space available to process the volumes in the exported category. This enables the exported category to be processed without delays and allows the library to continue with the next stacked volume.

Once the export operation is complete, all attached hosts are notified of this completion. This enables export completion status to be independent of the status of

the host that initiated the export operation. If the attached host has the export list volume in its TCDB, messages are issued with the completion results of the export operation. To move volumes, the DFSMSrmm EDGRPTD utility can be used to create a report extract file from which movement reports can be generated.

The LIBRARY EXPORT,*volser*,CANCEL command or the CBRXLCS FUNC=EXPORT programming interface with the cancel option can be used to cancel the export operation. When the library manager receives a cancel for an executing export operation, it completes writing the logical volume to the stacked volume it is currently processing and then terminates the export operation; therefore, the cancel may not be immediate.

Importing Volumes

To introduce logical volumes with data into a VTS library or to reintroduce logical volumes back into a VTS to reuse the volume serial numbers, the volumes must be imported into the library.

An import operation begins with the user or the tape management system writing a list of volumes to be imported into a library on a logical volume in the library. This logical volume then becomes the import list volume. This import list volume can indicate:

- Import all logical volumes

  All logical volumes residing on the stacked volumes specified will be imported.

- Import specific logical volumes

  In this case, the list of volumes is a list of stacked/logical pairs; each stacked volume is followed by the logical volume to import.

The required stacked volumes containing logical volumes to be imported must be entered into the library prior to initiating the import operation at the library. The DFSMSrmm SEARCHVOLUME subcommand with the CLIST operand can be used to create a list of the volumes to be imported.

Once the import list volume has been written and the stacked volumes are entered into the library, the host needs to notify the library of the logical volume being used for the import operation and to initiate the import operation at the library. The CBRXLCS external services programming interface FUNC=IMPORT or the LIBRARY IMPORT command can be used to initiate the import operation and to identify the import list volume.

When all of the requested logical volumes on a stacked volume have been imported, the library manager places the logical volumes into the insert category to be processed as part of normal cartridge entry processing. Indicators are assigned to the volume to indicate whether the volume is logical or not and whether it has been imported into the VTS. The CBRUXENT installation exit is called for each logical volume. If the logical volume being imported is known to DFSMSrmm as having been exported, DFSMSrmm clears the container information. For more information on DFSMSrmm, see “Preparing to Implement DFSMSrmm” on page 247.

When all of the requested logical volumes on a stacked volume have been processed at the host (no longer remain in the insert category at the library manager), the library continues with the next stacked volume referenced in the import list

volume. The stacked volume that the library just processed is left in the import category pending further operator instruction.

Once the import operation is complete, all attached hosts are notified of this completion. This enables import completion status to be independent of the status of the host that initiated the import operation. If the attached host has the import list volume in its TCDB, messages are issued with the completion results of the import operation.

There may be instances in which an exported logical volume is needed during job processing and the logical volume has not previously been imported. The library allows an operator to initiate a single volume import operation at the library manager. The operator enters the stacked volume containing the logical volume to import and indicates which logical volume is being imported. This eliminates the need for the host to write the logical list volume.

Functional Components Enhanced
DFSMSdfp(OAM, ISMF, and SMS), and DFSMSrmm

Software Dependencies
This support is available as a DFSMS/MVS 1.4 or DFSMS/MVS 1.5 SPE

APAR Reference

OW36342/OW36343 (OAM), OW36349/OW36350 (DFSMSrmm), OW36351/OW363 (ISMF)

Hardware Dependencies
A model B18 VTS library including:

- the extended high performance option (EHPO) feature
- a minimum of four 3590 native devices
- the convenience I/O station installed

Coexistence Issues
All LCS-related toleration support will be provided at DFSMS/MVS 1.1.0, 1.2.0, 1.3.0, 1.4.0., and 1.5.0.

Modified Operator Commands

DISPLAY SMS,VOLUME(*volser*)
DISPLAY SMS,LIBRARY(*library-name*),DETAIL
DISPLAY SMS,OAM

New Operator Commands

LIBRARY IMPORT,*volser*
LIBRARY EXPORT,*volser*

Programming Interfaces Affected

- Pre-ACS interface
- CBRXLCS
  - FUNC=EXPORT allows an installation to identify the export list volume and initiate or cancel an export operation. This is a new function.
  - FUNC=IMPORT allows an installation to identify the import list volume and initiate or cancel an import operation. This is a new function.

- FUNC=QVR allows an installation to query the library residency of a volume that does not have a volume record in the tape configuration database (TCDB). This is a modification of this function.

- CBRLCSPL is updated with new function reason codes.
- CBRUXENT/CBRUXEPL is updated with indicators to indicate whether the volume is a logical or physical volume and whether the logical volume was imported.
- CBRUXEJC/CBRUXJPL is updated with an indicator to indicate if the logical volume was exported and it is updated with a new field for the container volume.
- CBRTVI is updated to provide additional information for volumes associated with a VTS.
- DFSMSrmm programming interfaces:
  - The DFSMSrmm ISPF dialog has been changed.
  - New reason codes have been added for the EDGLCSUX programming interface.
  - DFSMSrmm macros EDGLCSUP, EDSVREC, and EDGRVEXT have been modified.
  - Additional reason codes have been added for DFSMSrmm.
  - The DFSMSrmm TSO ADDVOLUME, CHANGEVOLUME, and SEARCHVOLUME subcommands have been modified.

Other Migration Considerations

To implement import/export in a VTS, an installation and/or its tape management system will need to enhance its current code and procedures. First, an export operation will need to be initiated at the host as part of the vaulting run and then the installation and/or its tape management system will need to track which container volume the exported logical volume resides on. The container volume will be reported through the cartridge eject installation exit (CBRUXEJC). When the logical volumes are needed in a library, the installation and/or its tape management system will need to initiate an import operation at the host to get the logical volumes entered into a library or for a single volume being imported, the operator can initiate a single volume import at the library itself.

For full functional capability, you must use the DFSMSrmm subcommands to ensure that the virtual tape server logical volumes are identified to DFSMSrmm as logical volumes.

Where to Find More Information

- *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*
- *OS/390 MVS System Messages, Vol 2 (ASB-EWX)*
- *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*
- *DFSMS/MVS DFSMSrmm Diagnosis Guide*
- *DFSMS/MVS DFSMSrmm Guide and Reference*
- *DFSMS/MVS DFSMSrmm Command Reference Summary*
- *DFSMS/MVS DFSMSrmm Application Programming Interface*

- *DFSMS/MVS DFSMSdfp Storage Administration Reference*

VSAM Extended Format Data Set Enhancements

Many advanced VSAM functions previously available only to key-sequenced data sets (KSDS) are now extended to the other VSAM data set organizations: entry-sequenced data sets (ESDS), relative-record data sets (RRDS), variable-length relative-record data sets (VRRDS) and linear data sets (LDS).

All VSAM data sets can now be defined in extended format, which provides the basis for other enhancements as follows:

- **Extended addressability:** Extended format data sets larger than 4 GB are now supported for all VSAM record organizations. Use data class parameter `DSNTYPE=EXT` to define a data set as extended format. Use data class parameter `EXTENDED ADDRESSABILITY` to specify this additional option.
- **Partial release:** Space beyond the high-used control area can now be released to the system for all VSAM extended format data sets. You can specify partial release through the management class, through JCL using `SPACE=(,,RLSE)` or with the DFSMSdss RELEASE command.
- **Space allocation on candidate volumes:** An extension to a new volume can now use either the primary or the secondary space quantity for any VSAM extended format data set. Use the data class parameter `ADDITIONAL VOLUME AMOUNT` to identify which one is preferred.
- **System-managed buffering:** For non-shared resource (NSR) access, system-managed buffering is available to all VSAM extended format data set types. You can invoke system-managed buffering by specifying a `Record_Access_Bias` in the data class or by using the JCL `ACCBIAS` subparameter of the `AMP` parameter.

Table 5 provides a summary of advanced functions available to each type of extended format data set.

Table 5. Summary of Advanced Functions By Data Set Organization

Function	KSDS	ESDS	RRDS	VRRDS	LDS
Extended addressability	Yes	Yes	Yes	Yes	Yes
Partial release	Yes	Yes	Yes	Yes	Yes
Candidate volume space	Yes	Yes	Yes	Yes	Yes
System-managed buffering for NSR	Yes	Yes	Yes	Yes	Yes
Compression	Yes	No	No	No	No

Notes:

- All data set types listed assume an extended format definition.
- Linear data sets processed using control interval access can use system-managed buffering.
- Access method compression is restricted to a KSDS.

Functional Components Enhanced

DFSMSdfp, DFSMSdss, DFSMShsm

Software Dependencies

There are no software dependencies.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

Extended format data sets other than the KSDS cannot be accessed on DFSMS/MVS releases prior to 1.5. Compatibility PTFs are available for DFSMSdfp and DFSMSdss at the DFSMS/MVS 1.2, 1.3 and 1.4 levels to prevent data integrity problems on these earlier releases that already support the extended format KSDS. Toleration maintenance is also necessary for VSAM RLS at the DFSMS/MVS 1.3 and 1.4 levels.

DB2 support has PTFs available to recognize when linear data sets are defined for extended addressability.

APAR Reference

OW32778, OW32754, OW31473

ISMF Changes

The Define/Alter panel of the ISMF Data Class Application has changed to remove the restriction on defining extended format data sets (previously, only Recorg=KS was allowed).

Programming Interfaces Affected

An application that uses an RRDS or VVRDS should not require changes provided the application accesses the data using a relative record number and not a relative byte address (RBA) for control interval access.

An application that uses an ESDS or LDS will require changes if the data is defined for extended addressability. With addressed or control interval processing, the RBA is passed as the argument in the request parameter list (RPL). The RPL also contains an option code specifying whether the request is using a 4-byte or 8-byte RBA. The application must provide an 8-byte RBA when the data set is defined for extended addressability and the type of access uses the RBA specified.

Special provisions are allowed for certain types of requests (for example GET SEQ,ADR or GET ADR,DIR,LRD) to use either a 4-byte or 8-byte RBA because, although ADR or CNV is specified in the RPL, the RBA argument passed is not actually referenced.

There are no new programming macros for this support. VSAM macros used for an extended format KSDS are also now available for the other data set types. Control block manipulation macros have special keywords to retrieve the longer RBA values.

Commands or Control Statements Modified

DFSMSdss supports logical dump, logical restore, data set copy, and release operations for all types of VSAM extended format data sets. DFSMSdss does not support physical data set dump or physical data set restore for any VSAM extended format data set. No DFSMSdss commands, keywords, or parameters have changed for this support.

DFSMShsm processing is affected in the following ways:

- DFSMShsm supports VSAM extended addressability through the use of relative control intervals rather than relative byte addresses when performing space calculations.
- DFSMShsm invokes DFSMSdss as the data mover to migrate, backup/dump, recall, or recover/restore VSAM data sets defined with extended addressability.
- DFSMShsm ABARS processing invokes DFSMSdss to dump and restore VSAM data sets defined with extended addressability.

Processing Restrictions

Note the following restrictions associated with extended format data sets:

- An alternate index (AIX) cannot be defined over an extended format ESDS.
- For a KSDS, the data set cannot be defined with keyranges or with the index sequence set imbedded in the data (using IMBED)
- Data sets defined for VSAM RLS access cannot use system-managed buffering.
- VSAM RLS does not support the use of linear data sets.
- Hiperbatch support is not available.
- DFSMShsm will not migrate to DASD (or allow a BACKDS) any VSAM extended format data set that is larger than 64K tracks.

It is also important to note that you must be careful when performing a REPRO on a catalog that contains extended format data sets. If you run REPRO while data sets are open in the source catalog, and any of those data sets extend, changes might not get copied to the target catalog, resulting in a mismatch between the information contained in the VVDS and the new target BCS. This could result in a loss of information or make the data sets inaccessible.

Other Migration Considerations

This support also enables applications that access very large BDAM data sets (with fixed block records) to convert to using VSAM RRDS to exploit sharing and data extension.

When converting an existing VSAM data set to an extended format VSAM data set, you can use IDCAMS REPRO to move data from its current location.

Other migration considerations for VSAM extended format data sets are described under:

- “VSAM Extended Addressability” on page 157
- “VSAM Load Enhancements” on page 112
- “VSAM System-Managed Buffering” on page 113
- “VSAM Partial Release” on page 200
- “VSAM Compression” on page 198

Where to Find More Information

- *DFSMS/MVS General Information*
- *DFSMS/MVS Using Data Sets*
- *DFSMS/MVS Macro Instructions for Data Sets*
- *DFSMS/MVS Access Method Services for ICF*

- *DFSMS/MVS Managing Catalogs*
- *DFSMS/MVS Implementing System-Managed Storage*
- *DFSMS/MVS DFSMSdfp Storage Administration Reference*
- *DFSMS/MVS DFSMShsm Storage Administration Guide*
- *DFSMS/MVS DFSMSdss Storage Administration Guide*
- *DFSMS/MVS Using ISMF*
- *DFSMS/MVS Checkpoint/Restart*

Migration Planning for DFSMS/MVS Version 1 Release 4

This chapter presents a technical overview of all new or enhanced functions provided in DFSMS/MVS 1.4:

- DFM/MVS DataAgent
- DFSMSdss enhanced protection of checkpointed data sets
- DFSMSdss multivolume selection
- DFSMSdss stand-Alone services
- DFSMShsm ABARS activity log deletion
- DFSMShsm ABARS externalization of the TGTGDS and OPTIMIZE parameters
- DFSMShsm ABARS file stacking
- DFSMShsm ABARS generation data group base name in allocate
- DFSMShsm ABARS 64 tasks
- DFSMShsm CDS RLS serialization
- DFSMShsm CPU time in WWFSR and ABR records
- DFSMShsm dump analysis elimination
- DFSMShsm duplex tape function
- DFSMShsm enhanced protection of checkpointed data sets
- DFSMShsm extended invocation of the ARCBEEXT exit
- DFSMShsm general function and usability improvements
- DFSMSrmm enhancements
- IBM 3590 tape subsystem
- NaviQuest
- OAM DUMP/RESTART/QUERY/MODIFY/DISABLE command changes
- OAM move volume enhancement for supporting tape media
- OAM SMF recording
- OAM Optical Library 3995-Cxx Series
- OAM volume stacking and the virtual tape server
- OPEN/CLOSE/End-of-Volume serviceability improvements
- Program management extensions
- Sequential access method tailored compression
- Storage Management Subsystem out-of-space (X37 Abend) reduction
- VSAM catalog search interface
- VSAM DCOLLECT enhancements
- VSAM last-referenced date at close
- VSAM load enhancements
- VSAM LSR enhancements
- VSAM system-managed buffering

For each new type of support, this chapter outlines environmental dependencies and migration considerations and, where appropriate, refers to other books in the DFSMS/MVS and OS/390 libraries where you can find more implementation assistance.

© Copyright IBM Corp. 1979, 1999

APAR Information

You should review the following informational APARs in conjunction with the planning information presented in this chapter:

II08878	List of toleration and compatibility maintenance required for the installation of DFSMS/MVS 1.4. This list of APARs and PTFs is duplicated in the RETAIN PSP bucket and also still exists in the *DFSMS/MVS Program Directory*.
II08925	Repository of DFSMS/MVS 1.4 information that was not available at the time the DFSMS/MVS books were published for general availability. Included are any updates to this book.

DFM/MVS DataAgent

As an extension to the Distributed FileManager (DFM), the DFM DataAgent provides a technique for initiating remote procedures (i.e. DataAgents) from workstations that run on MVS. These routines can, for example, issue TSO commands, execute sorts, and perform functions not directly supported by the DFM component itself. DFM DataAgents allow the workstation application to execute routines that can operate on data or to execute functions such as CLISTs and REXX execs.

Functional Components Enhanced
DFSMSdfp, DFSMSdss

Software Dependencies
The Distributed FileManager requires APPC/MVS and VTAM for communicating with other systems on an SNA LU 6.2 network. RACF is required for checking the authorization of the remote systems to connect to MVS/ESA and to access specific data sets.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Installation Considerations
The system programmer will have to determine the appropriate setting for RESTRICT_START and should run the installation jobs for the sample application.

After installing DFSMS/MVS 1.4, the following steps are required to completely install DFM DataAgent:

- Browse the SAMPLIB member DFMREADM for DFM DataAgent highlights.
- Verify that the DFM member in PROCLIB is correct.
- Update the DFM00 member in PARMLIB. Set RESTRICT_START as appropriate for your installation.
- Update and submit the DFMXTSOI member in SAMPLIB to build the DFMQTSO and DFMXTSO agent procedures in PROCLIB and the DFMQTSO executable in LINKLIB.

- If your installation will write DataAgent routines in the C language, copy the DFMXAGNT member from SAMPLIB to a suitable working library and use your installation's compile and linkedit JCL to build the DFMXAGNT executable code. Refer to the prologue in DFMXAGNT source for instructions concerning how to build the DFMXAGNT member in PROCLIB.
- If you want to install the SORT sample, update and submit the SAMPLIB member DFMXSRTI.
- If you want to use DFMX0001 as a generic DataAgent procedure you can copy it to PROCLIB.
- To verify that the DataAgent is installed properly, install DFMACALL on a workstation connected to the mainframe and run the above DataAgent routines. Refer to the prologues in the source for the routines or look at DFMACALL help text for more details.
- If DataAgent routines can initiate long-running processes, APPC message limit and timeout settings should be adjusted accordingly.

Where to Find More Information

DFSMS/MVS DFM/MVS Guide and Reference
DFSMS/MVS General Information
OS/390 MVS System Messages, Vol 3 (GDE-IEB)

DFSMSdss Enhanced Protection of Checkpointed Data Sets

In this release, MVS Checkpoint/Restart has added an indicator that uniquely distinguishes between SMS physical sequential data sets checkpointed by MVS Checkpoint/Restart and SMS physical sequential data sets checkpointed by IMS.

DFSMSdss has added support to prevent the movement of both types of SMS checkpointed data sets by the DFSMSdss CONVERTV, logical DUMP, logical RESTORE, and data set COPY and RELEASE functions unless the FORCECP keyword is specified. Furthermore, the FORCECP keyword is necessary if MVS Checkpoint/Restart checkpointed data sets are to be processed by physical data set RESTORE and DEFRAG functions. When DFSMSdss processes a checkpointed data set of either type, the checkpointed indicators are removed from the target of the operation.

Functional Components Enhanced
DFSMSdss, DFSMShsm, DFSMSdfp

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
SMS physical sequential data sets that are indicated as checkpointed by MVS Checkpoint/Restart are not processed by previous levels of DFSMSdss.

Commands Modified

The DFSMSdss commands CONVERTV, COPY, DUMP, DEFRAG, RELEASE and RESTORE have been modified with the optional keyword FORCECP(*days*).

Where to Find More Information

DFSMS/MVS DFSMSdss Storage Administration Reference

DFSMSdss Multivolume Selection

DFSMSdss multivolume selection adds a new data set selection keyword that works with the user-supplied input volume list to control the selection of multi-volume data sets during logical data set dump and copy, and during conversion to or from SMS management. It is now possible to select a multivolume data set only if its first volume is included in the input volume list. The new SELECTMULTI keyword obsoletes the current ALLMULTI keyword.

Functional Components Enhanced

DFSMSdss

Software Dependencies

There are no additional software dependencies.

Hardware Dependencies

There are no additional hardware dependencies.

Coexistence Issues

There are no coexistence issues for this support.

Commands Modified

The DFSMSdss commands CONVERTV, COPY, and DUMP have been modified with the optional keyword SELECTMULTI(ALL | ANY | FIRST).

Where to Find More Information

DFSMS/MVS DFSMSdss Storage Administration Reference

DFSMSdss Stand-Alone Services

IBM offers this improved Stand-Alone data recovery solution to all users of the DFSMS/MVS Version 1 Release 4 DFSMSdss software package. This Stand-Alone recovery function was previously available as an enhancement to DFSMS/MVS Version 1 Release 3 DFSMSdss and previous releases of DFSMSdss and DFDSS.

The new Stand-Alone Services function completely replaces and makes obsolete the previous Stand-Alone Restore function, which is no longer available as part of DFSMSdss.

This new version of IBM's Stand-Alone restore function is a single-purpose program designed to allow the system programmer to restore vital system packs during disaster recovery without needing to rely on an MVS environment. Stand-Alone Services runs independently of a system environment either as a "true" stand-alone system or under a VM system.

The Stand-Alone Services program operates on an IBM System/370 processor in either ESA/370 mode, 370-XA mode, or System/370 (S/370) mode. It also runs on

an IBM System/390 processor in ESA/390 mode or S/370 mode. The Stand-Alone Services program can run on a machine that is in BASIC or LPAR mode, or in a virtual machine under VM.

Functional Components Enhanced
DFSMSdss

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Operator Commands Modified
The command syntax to perform a restore with the new Stand-Alone Services function is different than with the previous Stand-Alone Restore function. Refer to *DFSMS/MVS DFSMSdss Storage Administration Reference* for details.

Other Migration Considerations
The method to build an IPL-able core image for the Stand-Alone Services function is different than for the previous Stand-Alone Restore function. Refer to *DFSMS/MVS DFSMSdss Storage Administration Reference* for a complete description of the DFSMSdss Stand-Alone Services function, including a listing of supported devices and instructions for creating the IPL-able Stand-Alone Services core image.

Where to Find More Information

DFSMS/MVS DFSMSdss Storage Administration Reference
OS/390 MVS System Messages, Vol 1 (ABA-ASA).

DFSMShsm ABARS Activity Log Deletion

The ABARS activity log deletion enhancement allows users to specify whether or not they want DFSMShsm to automatically delete the ABARS activity log during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing. When automatic deletion is specified, DFSMShsm deletes older versions of the ABARS activity log, leaving only the versions of the log that correspond to existing aggregate backup and recovery versions. The ABARS activity logs no longer need to be manually managed.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Commands Modified
The ABARSDELETEACTIVITY(Y |N) parameter has been added to the DFSMShsm SETSYS command. This parameter allows users to specify

whether or not they want DFSMShsm to automatically delete the ABARS activity log during ABARS roll-off processing or EXPIREBV ABARSVERSIONS processing. The default is N, which specifies that there is no automatic deletion.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm ABARS Externalization of the TGTGDS and OPTIMIZE Parameters

DFSMShsm is externalizing the functions of the TGTGDS parameter used by ARECOVER processing, and the OPTIMIZE parameter used by ABACKUP processing, by adding two new SETSYS parameters.

The new SETSYS ARECOVERTGTGDS(*option*) parameter provides greater flexibility managing SMS-managed generation data sets that are being restored to level 0 DASD.

The new SETSYS ABARSOPTIMIZE(*n*) parameter allows installations to adjust performance when backing up level 0 DASD data sets that are specified in an ABARS INCLUDE data set list.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Commands Modified
The new SETSYS ARECOVERTGTGDS(*option*) command allows you to select from the following options:

DEFERRED Specifies that the target data set is to be assigned the DEFERRED status.

ACTIVE Specifies that the target data set is to be assigned the ACTIVE status, for example, rolled into the GDG base.

ROLLEDOFF Specifies that the target data set is to be assigned the ROLLEDOFF status.

SOURCE Specifies that the target data set is to be assigned the same status the data set had when it was backed up.

The default option is SOURCE, which is consistent with prior releases of DFSMShsm.

Installations can override the SETSYS ARECOVERTGTGDS(*option*) specification with a new ARECOVER command parameter, ARECOVER TGTGDS(*option*). The options for ARECOVER TGTGDS are the same options as for SETSYS ARECOVERTGTGDS.

The new SETSYS ABARSOPTIMIZE(*n*) command controls what ABACKUP will specify with the OPTIMIZE parameter when invoking DFSMSdss to back up level 0 DASD data sets.

Installations can override the SETSYS ABARSOPTIMIZE(*n*) specification with a new ABACKUP parameter, OPTIMIZE(*option*).

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm ABARS File Stacking

ABARS file stacking allows the output files from a single aggregate group to be stacked on a minimum of tape cartridges.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
On a DFSMS/MVS 1.4 system, you can use ARECOVER to recover aggregate backups created by a lower-level release of DFSMS/MVS. However, a lower-level release of DFSMS/MVS can only use ARECOVER to recover aggregate backups created by a DFSMS/MVS 1.4 system if the NOSTACK option was specified when the aggregate backup was created.

Performance Characteristics
During ARECOVER processing, the control file is always the first file read. Because it must be the last file written during ABACKUP processing, the control file will always be on file sequence number four. This requires that the tape cartridge be forward-spaced to file sequence number four before the control file data can be read. With new tape technologies, such as those found on the IBM 3490E, 3590-1, and 3591 tape drives, this performance impact should be minimal.

Using the stacking option with ABACKUP minimizes the number of tape cartridges used, thus reducing the number of tape mounts and tape allocations required.

ISMF Changes
The aggregate group backup and aggregate group recover ISMF panels for the aggregate group application have been changed to include several new fields, including a field for selecting the STACK or NOSTACK option. If the field is left blank, the value specified by the SETSYS ABARSTAPES command, or its default value of STACK, is used.

There are also new ISMF message help panels to support the new fields.

Commands Modified
The ABARSTAPES(STACK | NOSTACK) parameters have been added to the DFSMShsm SETSYS command, with STACK being the default.

STACK and NOSTACK parameters have been added to the ABACKUP command, allowing installations to override the SETSYS ABARSTAPES command setting as needed.

STACK and NOSTACK parameters have been added to the ARECOVER DATASETNAME command so you can indicate to ARECOVER processing whether the ABACKUP output files are stacked or not. The STACK and NOSTACK parameters are not used with the ARECOVER AGGREGATE command, as it obtains the information it needs from the ABR record and catalog.

The DFSMShsm QUERY SETSYS and QUERY ABARS commands have been enhanced to display the status of the ABARSTAPES parameter in message ARC6036I.

Other Migration Considerations

The ARECOVER DATASETNAME command used with the NOSTACK parameter allows a DFSMS/MVS 1.4 system to recover ABACKUP output created on lower-level releases of DFSMS/MVS.

If installations want to redirect the ABACKUP output file allocations using their ACS routines, they must use the NOSTACK option.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm ABARS Generation Data Group Base Name in Allocate

ABARS support has been enhanced to allow installations to specify a generation data group (GDG) base name in the ABARS ALLOCATE statement. ABARS defines the GDG base name, if one does not already exist, prior to restoring any generation data sets (GDSs) associated with the GDG. This can be done even if GDSs associated with the GDG have not been specified in the selection data set.

Functional Components Enhanced

DFSMShsm

Software Dependencies

There are no additional software dependencies.

Hardware Dependencies

There are no additional hardware dependencies.

Coexistence Issues

If an installation is using DFSMS/MVS 1.4 and includes GDG base names in the ABARS ALLOCATE statements, and they want to recover the aggregate backups on a system at a lower-level of DFSMS/MVS, a coexistence PTF is required.

Installations using DFSMS/MVS 1.4 are able to recover aggregate backups created on lower-level versions of DFSMS/MVS.

Processing Restrictions

GDG base names specified in the ALLOCATE statement have the following restrictions:

- They are only defined if they do not already exist at the recovery site.

- The DSCONFLICT parameter of the ARECOVER command does not apply to these GDG base names.
- The ARCCREXT installation exit does not get control for conflicts that occur with GDG base names in the ALLOCATE statement.
- RECOVERNEWNAMELEVEL and RECOVERNEWNAMEALL apply to these GDG base names.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide

DFSMShsm ABARS 64 Tasks

ABARS has been enhanced to allow up to 64 concurrent ABARS address spaces. The current restrictions remain that an installation cannot simultaneously back up the same aggregate group name, or recover using the same aggregate group name or control file data set name.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Performance Characteristics
Installations should closely monitor ABARS as they increase the number of ABARS tasks running concurrently to ensure that ABARS tasks don't over-utilize system resources. Some things to consider are channel path loads, data set contention, catalog contention, and contention for ML2 tape volumes.

Commands Modified
The DFSMShsm SETSYS MAXABARSADDRESSSPACE(*nn*) command has been enhanced to allow the value of *nn* to range from 1 to 64, with the default remaining as 1.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm CDS RLS Serialization

CDS RLS serialization support provides the option for the DFSMShsm control data sets to be accessed in record-level sharing (RLS) mode. This enables DFSMShsm to take advantage of the features of the coupling facility hardware for CDS access.

When RLS is used to access CDSs, they cannot be defined with key ranges. Multi-cluster support for the CDSs has been enhanced to enable DFSMShsm to dynamically calculate key boundaries for multicluster CDSs. DFSMShsm continues to

support key-ranges for non-RLS accessed CDSs. RLS is not a requirement for using dynamic boundary multicluster CDSs.

The header for log and journal records has been changed to include a one-byte field to store the host id that generated the record. In addition to this, sequence numbers are no longer written in the record header. The sequence numbers are not required because DFSORT has an EQUALS parameter that keeps sorted records in the chronological order that they are found. This requires that all journal records created using DFSMS/MVS DFSMShsm 1.4 be sorted by key and using the EQUALS parameter, instead of sorting by key and sequence number.

Functional Components Enhanced
DFSMShsm

Software Dependencies
Global resource serialization or an equivalent function is required.

Hardware Dependencies
To access the CDSs in RLS mode, processors must be in attached to enabled coupling facility hardware. For more details about the hardware dependencies, see “Hardware Dependencies” under the heading “VSAM Record-Level Sharing” on page 161.

Coexistence Issues
In multiprocessor installations where the processors are accessing the same CDSs, if one processor accesses the CDSs in RLS mode, all processors in the installation must access the CDSs in RLS mode.

DFSMShsm fails at startup if the selected CDS serialization technique of the starting DFSMShsm processor differs from the CDS serialization technique of another DFSMShsm processor already active in a multiprocessor installation.

DFSMShsm fails at startup on processors running a release prior to DFSMS/MVS 1.4 if an active DFSMS/MVS 1.4 processor is accessing the CDSs in RLS mode. A toleration PTF is available for this condition.

If the DFSMShsm CDSs are defined to use dynamic-key boundary multicluster support, they cannot be accessed by processors running DFSMShsm at releases prior to the DFSMS/MVS 1.4 release.

Performance Characteristics
Accessing CDSs in RLS mode is expected to reduce contention when running primary space management and automatic backup on two or more processors. DFSMShsm benefits from the serialization and data cache features of VSAM RLS, and does not have to perform CDS VERIFY or buffer invalidation.

To maximize performance for CDS accesses, a CI size of 4K is recommended for the CDSs. A higher value should be used if the maximum record length is greater than 4K.

System Interfaces Affected
The startup procedure keyword CDSSHR has been enhanced by adding RLS as a parameter. When RLS is specified, the DFSMShsm CDSs are accessed in RLS mode, and any values specified for CDSQ and CDSR are ignored.

ARCIMPRT has been enhanced to recover multicluster key-boundary CDSs. A new parameter, FORCE, has been added to ARCIMPRT to assist users in performing CDS recoveries.

When using DCOLLECT to analyze the DFSMShsm control data sets, DFSMShsm first attempts to open the CDSs in RLS mode. If the CDSs are not RLS eligible, an error message is issued. This message can be ignored for non-RLS eligible data sets because the OPEN will be retried in non-RLS mode.

Commands Modified

The QUERY CONTROLDATASETS command has been enhanced to display the CDS serialization technique in use, and the key ranges being used for dynamic key-boundary multicluster CDSs.

Processing Restrictions

If CDS backup is invoked while the CDSs are accessed in RLS mode, DFSMSdss must be the datamover. If the backup is directed to tape, the PARALLEL parameter of the SETSYS CDSVERSIONBACKUP(BACKUPDEVICECATEGORY) must be specified.

Other Migration Considerations

Before CDSs can be accessed in RLS mode, they must be defined or altered to be RLS eligible. You must do this for all three of the CDSs. The CDSs must also be SMS-managed, with a storage class definition that indicates which coupling facility to use.

All operating systems running with DFSMShsm must be coupling facility capable, and the processors must have access to the coupling facility.

The CDSs cannot be defined as key-range data sets. CDSs previously defined as multicluster must be redefined without key ranges.

CDSs accessed in RLS mode enqueue resources differently from CDSs accessed in non-RLS mode. In a multiprocessor environment, the resources must be propagated by using global resource serialization (GRS) or an equivalent product.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm CPU Time in WWFSR and ABR Records

This enhancement allows ABARS to maintain the CPU time for processing ABACKUP and ARECOVER requests in the WWFSR control block. Installations can then use the reported CPU time to assist in calculating the “charge back” costs for users' ABARS requests.

A WWFSR is created at the completion of each individual ABACKUP and ARECOVER request and can be written as an SMF record if requested.

The ABR record maintains an ABACKUP CPU time and an ARECOVER CPU time.

If an ARECOVER request fails and is reissued with a valid RESTART data set, the CPU time in the WWFSR only reflects the processing time of the remaining data

sets. The ABR record, however, accumulates the CPU times of each restart until the recovery of the aggregate is successful.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence issues.

ISMF Changes
The ISMF aggregate group definition panels allow specification of a 32-character accounting code. This accounting code is written to the WWFSR control block, ABR records, and the ABACKUP control file. It is written to the ABACKUP control file so the account code can be used at the recovery site without requiring an aggregate definition at the recovery site.

The affected ISMF aggregate group definition panels include:

- Aggregate Group Define/Alter
- Aggregate Group Display
- Aggregate Group List Display
- Aggregate Group List Sort
- Aggregate Group List View
- Aggregate Group List Print

System Interfaces Affected
If the installation specifies SETSYS SMF(*smfid*), then ABARS writes the WWFSR as *smfid*+1 in the SYS1.MANx or SYS1.MANy system data sets.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference
DFSMS/MVS DFSMShsm Diagnosis Reference

DFSMShsm Dump Analysis Elimination

DFSMShsm generates dumps to gather first-failure diagnostic information. Problems that occur multiple times, including on different hosts, often generate a storage dump for each occurrence. DFSMShsm supplies support to the MVS dump analysis elimination (DAE) function, allowing that function to build a symptom string to be used in the suppression of duplicate dumps.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues

Only systems running DFSMS/MVS 1.4 are able to utilize this support.

Programming Interfaces Affected

SETSYS SYS1DUMP must be used.

The SYS1.PARMLIB member ADYSET*xx* must be coded with the keyword SUPPRESSALL. For example:

```
DAE=START,RECORDS(400),
    SYSMDUMP(MATCH,UPDATE),
    SVCDUMP(MATCH,UPDATE,SUPPRESSALL)
```

To allow a single DAE data set to be shared by multiple systems in a parallel sysplex, and for dumps to be suppressed across multiple MVS systems, the coupling services of the cross system coupling facility (XCF), and global resource serialization (GRS) must be enabled.

Processing Restrictions

DAE support can be used for functions that run in the DFSMShsm primary address space, and the ABARS secondary address space.

DAE does not suppress SYSABEND, SYSUDUMP, SYSMDUMP, or SNAP dumps, nor does it suppress dumps that originate from SLIP or DUMP operator commands. It also does not suppress dumps produced by the DFSMShsm TRAP command.

DFSMShsm Duplex Tape Function

The duplex tape function provides an alternative to TAPECOPY processing for backup and migration cartridge-type tapes. Two tapes are created concurrently with one designated the original, the other the alternate. The intent is that the original tape be kept onsite, while the alternate can either be taken offsite or written to a remote tape library. The tapes labels indicate whether a tape is an original or an alternate.

The new tapes are compatible with tapes created by TAPECOPY. The TAPECOPY functions and methodology continue to be supported.

In an SMS environment, the ACS routines can direct the alternate tape to a different tape library. In a non-SMS environment, the output restrictor is used for both the original and the alternate. ACS filtering or esoteric restrictor unit names can be set up to cause the desired tape allocations to occur.

DFSMShsm selects tapes using the following selection process:

- A partial tape and its alternate are selected.
- If no partial tapes with alternates are found, an empty ADDVOLed volume with a scratch volume as an alternate are selected.
- If there are no empty ADDVOLed volumes, two scratch volumes are selected.

When problems occur that allow an original tape to be created, but prevent the alternate from being created by the duplex tape function, DFSMShsm automatically schedules internal TAPECOPY processing to ensure that valid alternate tapes exist. At either the EOV for the original volume or the end of the task, DFSMShsm runs TAPECOPY processing to create an alternate tape. If that also is unsuccessful, automatic scheduling of internal TAPECOPY processing occurs during secondary

space management for migration volumes, and during automatic backup processing on the primary host for backup volumes.

As part of the automatic internal scheduling of TAPECOPY processing, a tape copy needed (TCN) record is written in the OCDS. When the automatic internal TAPECOPY completes successfully, the TCN record for the target volume is deleted.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
When using the duplex tape function, one additional tape drive is required per writing task. Adequate numbers of drives have to be available for the tasking level used by the functions using duplexing during the time periods those functions run. This includes the recycling of duplexed tapes.

Coexistence Issues
PTFs are available to prevent releases prior to DFSMS/MVS 1.4 from processing tapes having duplex alternates. If you have a multiprocessor installation where some processors are running DFSMS/MVS 1.4 while others are on a prior DFSMS/MVS release, PTFs are required.

Performance Characteristics
It is recommended that you run comparison performance tests using the duplex tape function in one case, and using non-duplexing with TAPECOPY processing in the other case.

Programming Interfaces Affected
A new flag has been added to the ARCTVEXT installation exit parameter list. When an alternate tape is being replaced and returned to scratch, and if the ARCTVEXT installation exit is active, the flag indicates if the action is the result of a user-issued TAPECOPY command. This information is provided so users are aware that an alternate tape is being returned to scratch, and that the original tape is not.

Commands Modified
An optional DUPLEX parameter has been added to the SETSYS command, along with the optional subparameters of BACKUP and MIGRATION.

You have the option of selecting the DUPLEX tape function:

- Not at all
- For backup volumes only
- For migration volumes only
- For both backup and migration volumes

The QUERY SETSYS command has been enhanced to display the current duplex status of migration and backup processing.

The LIST and HLIST commands have been enhanced to show duplex tape status and alternate tape volume serial numbers.

Processing Restrictions
The alternate tape must have the same tape geometry as the original. For example, if the original tape is a 3490 ECCST tape, the alternate must be

also. DFSMShsm also maintains the existing TAPECOPY methodology for installations with tape-drive constraints.

The duplex tape function is not supported with SETSYS TAPEUTILIZATION NOLIMIT set for the output unit being duplexed. This is because the alternate tape must match the original tape exactly, and with SETSYS TAPEUTILIZATION set to NOLIMIT this is impossible to guarantee, since the original tape could be longer than the alternate.

The duplex tape function does not support creating duplex copies of ML2 tapes during ARECOVER processing. ARECOVER processing writes a TCN record in the OCDS when recovering an ML2 volume. That causes a TAPECOPY to be submitted during the next occurrence of secondary migration processing. The TAPECOPY is not submitted immediately because a TAPECOPY running during the recovery process might not be wanted.

Other Migration Considerations

Previous releases of DFHSM or DFSMS/MVS must not attempt to extend a DFSMS/MVS 1.4 DFSMShsm backup or migration tape created in duplex. A PTF is available to prevent attempts to extend these data sets.

If you are currently using the PARTIALTAPE(REUSE) option, and you are going to begin using the duplex tape function, you should consider marking the existing partial tapes as full. This can be done with the DELVOL MARKFULL command. Partial tapes without alternates are not selected for duplexing. If you mark the tapes as full, they become eligible to be recycled.

If you use duplex creation for both migration and backup and you have been auto scheduling tape copying, you probably want to remove the auto scheduling. If you are duplexing either migration or backup tapes but not both, you can auto schedule tape copies based on the type indicated in the ARC0421I message.

Be aware that the duplexing requirement for a second tape drive for each task also applies to the recycle of the migration and backup tapes.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm Enhanced Protection of Checkpointed Data Sets

Enhanced protection of checkpointed data sets is provided to improve the protection of MVS and IMS GSAM SMS physical sequential data sets that have been checkpointed. This is accomplished by giving the user greater control over the movement of these data sets.

When MVS and IMS GSAM SMS physical sequential data sets are checkpointed, some bits are set to indicate the condition. DFSMShsm and DFSMSdss use this information to identify checkpointed data sets.

For both command and automatic migration, DFSMShsm determines whether the data set being migrated is a checkpointed data set. If the data set is checkpointed, DFSMShsm determines whether the data set is eligible for migration. DFSMShsm migration, including extent reduction, is delayed for a fixed number of days for MVS

and IMS checkpointed data sets. The default delay is five days. A checkpointed data set is eligible for migration when the date-last-referenced, plus the number of days the data set is to be treated as unmovable, are less than or equal to the current date. If the data set is eligible, DFSMShsm builds the command syntax for a DFSMSdss logical data set dump, including the keyword and parameter FORCECP(0). For command migration, error messages are issued for ineligible data sets.

DFSMShsm does not allow users to include the FORCECP keyword as part of the DFSMShsm MIGRATION command syntax. Instead, a supported patch byte is used to force migration.

Any checkpointed data set that was migrated can be recalled without any additional delays. For recalls, DFSMShsm builds the command syntax for a DFSMSdss logical data set restore, including the keyword and parameter FORCECP(0).

The backup of a data set with BACKUP or ABACKUP is not delayed for a checkpointed data set. For BACKUP and ABACKUP, DFSMShsm builds the command syntax for a DFSMSdss logical data set dump, including the keyword and parameter FORCECP(0).

The RECOVER and ARECOVER commands allow recovery, including replacement, of any checkpointed data sets. There are no delays for the checkpointed data sets. For ARECOVER, DFSMShsm builds the command syntax for a DFSMSdss logical data set restore including the keyword and parameter FORCECP(0). For RECOVER, DFSMShsm builds the command syntax for a DFSMSdss logical data set restore, or a DFSMSdss physical data set restore including the keyword and parameter FORCECP(0).

DFSMSdss does not distinguish checkpointed data sets from other data sets during full volume dumps, so when DFSMShsm builds the command syntax for a DFSMSdss physical volume dump, no keyword is required. There are no delays for dump processing.

Functional Components Enhanced
DFSMShsm and DFSMSdss

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
This support only extends to DFSMS/MVS 1.4 and subsequent releases. Prior releases are not supported with this enhanced protection for checkpointed data sets.

Programming Interfaces Affected
A programming patch is available to enable users to modify the number of days elapsed since the date-last-referenced for a checkpointed data set to be eligible for migration.

Commands or Control Statements Modified
DFSMShsm now includes the FORCE(*days*) keyword and parameter when invoking DFSMSdss during migration, recall, backup, recover, ABACKUP, and ARECOVER of checkpointed data sets.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference
DFSMS/MVS DFSMSdss Storage Administration Reference

DFSMShsm Extended Invocation of the ARCBEEXT Exit

ABACKUP processing has been enhanced to extend the error conditions for which the ABARS backup error installation exit (ARCBEEXT) gets control. The exit allows the user to skip a data set when various error conditions are encountered during ABACKUP. With this enhancement, the ARCBEEXT exit will gain control when DFSMSdss errors occur while dumping level 0 DASD data sets in the INCLUDE list.

The ARCBEEXT installation exit is not invoked for DFSMSdss filter errors. When DFSMSdss errors occur during DFSMSdss filter processing, ABACKUP processing now fails immediately following the error detection.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Programming Interfaces Affected
The ARCBEEXT installation exit can be invoked by DFSMSdss during ABACKUP processing when DFSMSdss is dumping level 0 data sets. The interface to ARCBEEXT passes a new bit (in the flag bytes) that indicates that the exit is being invoked due to a failure during DFSMSdss processing.

Commands Affected
To use this support, the ARCBEEXT installation exit must be activated by specifying the DFSMShsm SETSYS EXITON(BE) command before the ABACKUP command is issued.

Where to Find More Information

DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference
DFSMS/MVS Installation Exits

DFSMShsm General Function and Usability Improvements

This section provides a summary of the general improvements to DFSMShsm function and usability introduced in this release.

Improvements include:

- **An alter without recall function:** This function is intended to replace the function previously supplied by the HALTER exec found in ARCTOOLS. The function is invoked by an IDCAMS ALTER *entryname* command, and can be used

to alter the storage class, or the management class, or both the storage and management classes for migrated data sets, without recalling the data sets.

The new alter without recall function causes both DFSMShsm's and catalog's records to be updated and kept in synchronization. The HALTER exec was rewritten to invoke the IDCAMS ALTER command, which updates both DFSMShsm's and catalog's records.

- **Improved granularity to the AUDIT DATASETCONTROLS restart (RESUME) function:** This improvement is accomplished by saving all 44 characters of the record key.

  When running on a DFSMS/MVS 1.4 system, an AUDIT DSCTL(MIG) RESUME or an AUDIT DSCTL(BACKUP) RESUME command can continue the immediately preceding AUDIT DSCTL(MIG) or AUDIT DSCTL(BACKUP) command on the same host. The granularity of the resuming position is expanded from 27 to 44 bytes. See *DFSMS/MVS DFSMShsm Storage Administration Reference* for more information.

Improvements to usability include:

- **DFSMSrmm and Optimizer no longer need be invoked through the DFSMShsm ARCTVEXT and ARCINEXT installation exits:** This allows you to use the ARCTVEXT installation exit for tape management systems other than DFSMSrmm, and releases the ARCINEXT for other uses.

  The DFSMSrmm code in the DFSMS/MVS 1.4 level of the ARCTVEXT installation exit, and the Optimizer code in the DFSMS/MVS 1.4 level of the ARCINEXT installation exit should be removed. If the exits were only being used for DFSMSrmm and Optimizer support, you can turn the exits off using the DFSMShsm SETSYS EXITOFF command.

  ADSM users who previously invoked the ARCTVEXT installation exit to utilize DFSMSrmm services should now invoke the new DFSMSrmm general-use programming interface EDGTVEXT.

- **Aliases for DFSMShsm keywords:** The intent in creating the following aliases is to simplify the usage of DFSMShsm commands. This change allows the term “BACKUP” to be used as an alias for BCDS, and the term “MIGRAT” to be used as an alias for MCDS on the DFSMShsm LIST and HLIST commands. This support includes the LIST LEVEL and LIST DSNAME commands. For example:

  ```
  LIST LEVEL(qualifier) BACKUP

  LIST LEVEL(datasetname) MIGRAT
  ```

  The BACKUPCONTROLDATASET and MIGRATIONCONTROLDATASET aliases that already existed will continue to be supported.

  In addition to BACKUP and MIGRAT, the abbreviation UUT can be used for USERUNITTABLE, and NOUUT for NOUSERUNITTABLE on the SETSYS command.

  For more information and examples regarding these aliases, see *DFSMS/MVS DFSMShsm Managing Your Own Data* and *DFSMS/MVS DFSMShsm Storage Administration Reference*.

- **Four new FSR types have been created:** This enhancement causes type 6 FSR records only to be written for user-requested, migrated data set deletions. The two records being converted from a type 6 to a type 17 are:

- Expire data set from ML1
- Expire data set from ML2

The four new FSR types are:

Type 17 Expire data set, which categorizes the data set being expired as being from a level 0, ML1, or ML2 volume. It also indicates if the data set is being deleted by its expiration date in the catalog or by the management class attributes, rather than by automatic deletion.

Type 18 Partial release data set on Level 0 DASD

Type 19 Expire incremental backup version, which indicates if the incremental backup version is being deleted by the EXPIREBV command, and if it's on a tape volume.

Type 20 Delete incremental backup version, which indicates if the incremental backup version being deleted is on a tape volume.

The following four new FSR records have been created:

- Expire data set from L0 DASD, ML1, and ML2 volumes
- Partial release data set on L0 DASD
- Expire incremental backup version
- Delete incremental backup version

- **DCOLLECT has been enhanced:** The enhancement enables DCOLLECT to provide the volume serial number from which the source is obtained for migration and backup processing. The volume serial number is for the first volume of the data set. The sizes of both record types M and B have been increased to accommodate the enhancement.

DFSMSrmm Enhancements

DFSMSrmm enhancements include:

- Improvements to DFSMSrmm inventory management vital records processing so vital record information is maintained for data sets even when they are no longer managed by a vital record specification
- Trial run vital record processing to see the effect of movement and retention policies on data sets and volumes before actually making changes to the DFSMSrmm control data set
- The new DFSMSrmm EDGTVEXT programming interface is provided as DFSMSrmm no longer provides its own version of the ARCTVEXT exit. EDGTVEXT as well as the existing EDGDFHSM programming interface can be used by products like DFSMShsm and ADSM for DFSMSrmm tape management support. DFSMShsm automatically calls EDGTVEXT so you no longer need to use ARCTVEXT to communicate with DFSMSrmm.
- The problem determination aid (PDA) trace facility is added to DFSMSrmm so that diagnostic information can be gathered at entry and exit points in the DFSMSrmm modules. The purpose of the trace is to gather sufficient information for problem analysis. The PDA trace is active by default.
- The DFSMSrmm EDGUX100 installation exit can be used to control recording of data sets on a volume. Use EDGUX100 to suppress recording for file 2 for certain volumes.

- DFSMSrmm TSO subcommands are enhanced to support search and change for ABEND and OPEN operands.
- ABEND and OPEN vital record specifications have been enhanced to support the use of JOBNAME.
- DFSMSrmm TSO CHANGEVOLUME PREVVOL subcommand can be used for all private volumes.
- DFSMSrmm recognizes DFSMShsm tape requests and allows input processing of volumes as long as the last 17 characters of the data set match the data set name that DFSMSrmm recorded.

Functional Components Enhanced
DFSMSrmm

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Performance Characteristics
There are no additional performance characteristics.

Programming Interfaces Affected
New general-use programming interface: EDGTVEXT

Where to Find More Information

DFSMS/MVS DFSMSrmm Command Reference Summary
DFSMS/MVS DFSMSrmm Diagnosis Guide
DFSMS/MVS DFSMSrmm Guide and Reference
DFSMS/MVS DFSMSrmm Implementation and Customization Guide

IBM 3590 Tape Subsystem

DFSMS/MVS 1.4 adds support for the IBM Magstar 3590 Tape Subsystem family of tape products. This includes support for MEDIA3, the IBM High Performance Cartridge Tape and MEDIA4, the IBM Extended High Performance Cartridge Tape.

Functional Components Enhanced
DFSMSdfp, DFSMShsm, DFSMSrmm

Software Dependencies
There are no other software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
Toleration PTFs are needed at the DFSMS/MVS 1.1 and DFSMS/MVS 1.2 levels if a 3494 or 3495 tape library dataserver containing 3590 devices and/or the TCDB containing 3590 media is shared with levels of software that do not support the 3590 devices (DFSMS/MVS 1.1) or support the 3590 but do not have all of the support PTFs installed (DFSMS/MVS 1.2). The toleration support is already built into the base DFSMS/MVS 1.3 release and is not a concern.

Other Migration Considerations

Tape drives in a tape library dataserver are defined using the HCD. A library ID defined to the ISMF Library application links the system-managed tape library definition to the tape library dataserver. The library ID is defined to HCD by specifying the LIBID and PORTID parameters for each library device. HCD help text for the LIBID and PORTID parameters explains how you can obtain the IDs.

Use the HCD to define 3590-1 devices to your system. An IPL or ACTIVATE of the resulting IODF is necessary. Changes to the LOADxx PARMLIB member are necessary if a new IODF is used.

You also need to update your ACS filter routines, SMS constructs, and installation exits to manage the usage of the new drives and media. Users of SVC 99 can change invocations to specify the new media type.

Where to Find More Information

DFSMS/MVS General Information
DFSMS/MVS Installation Exits
DFSMS/MVS DFSMSdfp Advanced Services
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Access Method Services for VSAM
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Utilities
DFSMS/MVS DFSMSdfp Diagnosis Reference
DFSMS/MVS Using Magnetic Tapes
DFSMS/MVS Using the Volume Mount Analyzer
DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference Summary
DFSMS/MVS DFSMSrmm Guide and Reference
DFSMS/MVS DFSMSrmm Implementation and Customization Guide
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries
Magstar and IBM 3590 High Performance Tape Subsystem Technical Guide

NaviQuest

NaviQuest is a component of DFSMSdfp that runs under ISMF. It is a tool that assists installations with their migration to system-managed storage (SMS). For those who have already migrated to SMS, it is useful for testing their configurations after changes. It can be used to extract or update a subset of the SMS configuration in a batch (non-ISMF) mode.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

NaviQuest runs under ISMF under the system administrator menu.

Hardware Dependencies

NaviQuest has no hardware dependencies.

Coexistence Issues

NaviQuest resides in the ISMF libraries.

Performance Characteristics

Performance is dependent on the TSO workload of the system.

SMS Changes

NaviQuest does not require any changes to SMS.

ISMF Changes

ISMF panels are modified to support NaviQuest functions. ISMF Primary Option Menu has an option (11) for invoking NaviQuest. Data Set Selection and Volume Selection Entry Panels contain a new field 'Query Name to Save or Retrieve' to support QSAVE and QRETRIEV. ACS Test Define/Alter panel has a new field, 'Expected Result', reducing the description field to one line.

Programming Interfaces Affected

NaviQuest provides batch processing but does not affect any interfaces.

Other Migration Considerations

Users that still have ISMF/E (5764-033) or NaviQuest Version 1 Release 1 (5655-ACS, FMID HACS110) need to delete it before applying this version.

Where to Find More Information

DFSMS/MVS NaviQuest User's Guide
DFSMS/MVS DFSMSdfp Diagnosis Reference
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS General Information
DFSMS/MVS Implementing System-Managed Storage

OAM Command Changes

Additions and changes to OAM commands include the following:

- F OAM,DUMP
- F OAM,RESTART
- F OAM,QUERY or MODIFY OAM,QUERY
- F OAM,UPDATE or MODIFY OAM,UPDATE
- LIBRARY DISABLE,CBRUXENT

OAM DUMP Command

OAM uses SVC dumps as a diagnostic tool for system hangs or performance delays. To capture this data, the operator usually issues a DUMP command specifying all required data areas needed for diagnostics. Most often this requires a recreate and technical assistance from IBM to ensure all necessary data is captured.

To streamline this process, OAM now provides the F OAM,DUMP command to allow OAM to collect all pertinent data instead of operators keying in the correct parameters after a recreate.

OAM RESTART Command

Previously, OAM would restart in the event of a change to an SCDS and in the activation of a new or changed SCDS into the current ACDS. This restart was done because OAM had no way of knowing if the activation of the SCDS included changes to the configuration information used by OAM. Often, these updates did not affect any work executing in the OAM address space.

OAM now provides users the ability to specify a RESTART option as a parameter on the OAM started procedure statement. This option indicates to OAM whether it should be restarted after a notification is received indicating that a new SCDS has been activated. Following is a sample OAM started procedure statement with the new RESTART option parameter.

```
//OAM PROC OSMC=YES,MAXS=2,UNLOAD=9999,EJECT=LRW,RESTART=YES
//IEFPROC EXEC PGM=CBROAM,REGION=0M,
// PARM=('OSMC=&OSMC,APLAN=CBROAM,MAXS=&MAXS,UNLOAD=&UNLOAD',
//    'EJECT=&EJECT,RESTART=&RESTART')
```

Users electing not to restart the OAM address space automatically have the opportunity to prompt OAM to restart in the case of an SCDS activation that does affect OAM processing. The OAM RESTART command causes the OAM address space to perform restart processing which rebuilds the in-storage control block structure to match the constructs and definitions found in the active SMS configuration. It helps to avoid having to perform a STOP and then a START of the OAM address space and it allows the OAM address space to retain its current ASID.

OAM QUERY Command

OAM provides a QUERY command that enables the system operator to query the status of active and waiting OAM requests. The F OAM,QUERY or MODIFY OAM,QUERY command can display information such as the following:

- Summary of active optical requests
- Summary of active object tape requests
- Summary of active tape library requests
- Summary of waiting optical requests
- Summary of waiting object tape requests
- Summary of waiting tape library requests
- Detailed information concerning active optical requests
- Detailed information concerning active object tape requests
- Detailed information concerning active tape library requests
- Detailed information concerning waiting optical requests
- Detailed information concerning waiting object tape requests
- Detailed information concerning waiting tape library requests

Note: OAM is the default name of the cataloged procedure in your SYS1.PROCLIB. If a name other than OAM is used for the cataloged procedure, use that name in the QUERY statement. For example, MODIFY *procname_name*,QUERY,ACTIVE,SUMMARY.

OAM MODIFY UPDATE Command

The MODIFY OAM command is changed to include the UPDATE parameter that provides users with the ability to update specific fields in the DB2 Volume Table or the Tape Volume Table. This command provides an alternative to using SPUFI to update the DB2 row for a specific volume in the table and using STOP and START on the OAM address space before the change is reflected.

OAM LIBRARY DISABLE Command

The LIBRARY DISABLE,*CBRUXENT* command provides the ability to disable cartridge entry processing on a particular system or to direct tape library cartidge entry processing to a particular system.

Functional Components Enhanced
DFSMSdfp (OAM)

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Program Interfaces Affected
No programming Interfaces are affected.

Other Migration Considerations
There are no migration considerations.

Where to find more information.

DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support and
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries.

OAM Move Volume Enhancements for Supporting Tape Media

The OAM Move Volume utility (MOVEVOL) is enhanced to include tape media. The utility is now capable of moving objects from a primary or backup source volume (a tape volume or one side of an optical disk) to one or more target volumes.

Additionally, if the source volume (either optical or tape) belongs to the OBJECT BACKUP storage group, the media type used for writing the objects is derived from the definition of the OBJECT BACKUP storage group. For example, if there is a valid SETOAM command for the OBJECT BACKUP storage group to which the source volume belongs, the target volume chosen is a tape volume. If there is no valid SETOAM command for the OBJECT BACKUP storage group belonging to the source volume, the target volume chosen is an optical disk volume.

The MOVEVOL utility can perform the following:

- Write objects using the optical drives defined in the OBJECT storage group when moving objects off a primary optical source volume

- Write objects using the tape drives allocated to the OBJECT storage group using the tape unit name specified in the SETOAM command for the OBJECT storage group when moving objects off a primary tape source volume
- Write objects using optical drives defined to or tape devices allocated to the OBJECT BACKUP storage group when moving objects off a backup optical or tape source volume.

Functional Components Enhanced
DFSMSdfp (OAM)

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Other Migration Considerations
For media migration purposes, the following scenarios are possible:

SCENARIO	MOVE FROM	MOVE TO
Movement from one optical media type to another optical media type for migration purposes to newer technology.	12-inch WORM 12-inch WORM 5.25-inch WORM	5.25-inch WORM 5.25-inch rewritable 5.25-inch rewritable
Movement from one optical media type to the same optical media type.	5.25-inch WORM 12-inch WORM	5.25-inch WORM 12-inch WORM
Movement from one optical media type to another optical media type of older technology.	5.25-inch WORM	12-inch WORM
Movement from one tape media type to another tape media type.	IBM Cartridge System Tape, IBM Enhanced Capacity Cartridge System Tape, IBM High Performance Cartridge Tape, or IBM Extended High Performance Cartridge Tape	IBM Cartridge System Tape, IBM Enhanced Capacity Cartridge System Tape, IBM High Performance Cartridge Tape, or IBM Extended High Performance Cartridge Tape
Movement from any optical media type to a tape volume belonging to the OBJECT BACKUP storage group.	12-inch WORM or 5.25-inch rewritable or 5.25-inch WORM	IBM Cartridge System Tape, IBM Enhanced Capacity Cartridge System Tape, IBM High Performance Cartridge Tape, or IBM Extended High Performance Cartridge Tape
Movement from any tape media type to optical media belonging to the OBJECT BACKUP storage group.	IBM Cartridge System Tape, IBM Enhanced Capacity Cartridge System Tape, IBM High Performance Cartridge Tape, or IBM Extended High Performance Cartridge Tape	12-inch WORM or 5.25-inch rewritable or 5.25-inch WORM
Note: WORM = write-once-read-many		

Where to find more information

DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support device connectivity requirements

OAM System Management Facility Recording

OAM System Management Facility (SMF) Recording allows users to collect data on OAM activities. This data can be used for:

- information system usage accounting and charge back to end user departments.
- OAM performance analysis and monitoring of systems to ensure optimum performance.
- capacity planning for procurement of additional hardware such as DASD devices and media.
- potential problem determination data.

OAM SMF Recording provides the flexibility for users to collect OAM statistical records at any specified time frame. Users can also choose to collect OAM statistics against all, a set of, one, or none of the OAM functions or activities. These functions and activities include:

- Invocations of Object Store and Retrieval (OSR) functions
- Invocations of the OAM Storage Management Component (OSMC) activities
- Optical library activities
- Object tape activities

Functional Components Enhanced
DFSMSdfp (OAM)

Software Dependencies
Maintenance is required for OS/390 1.3 and for MVS/ESA SP 5.2 and follow-on releases.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Program Interfaces Affected
The OSREQ macro (OAM Application Programming Interface) is updated with a new keyword—TTOKEN for user supplied information.

Other Migration Considerations
There are no migration considerations.

Where to find more information

DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support
DFSMS/MVS OAM Application Programmer's Reference

OAM Optical Library 3995-Cxx Series

OAM provides additional optical device support for the new 3995-Cxx optical libraries. These libraries consist of a controller (3995-C3A) and a multifunction optical disk drive (3995-SW3—the drive used in all the C-xx model libraries). The new multifunction optical disk drive provides faster processing speed, along with the capability of handling a larger cartridge capacity than previous 3995 optical disk drives. Additionally, all 3995 media types can be read from or written to (with the exception of single-density WORM or rewritable media—read only capability) using the 3995-SW3 multifunction optical disk drive. The 3995-SW3 can also be used as an operator-accessible drive.

The new 3995 optical library models are available in a variety of slot sizes. This provides users the versatility of smaller or larger storage capacities to suit their requirements. These libraries can vary in size from 52 to 258 slots.

OAM also supports a new media that has four times the capacity of a single-density optical disk cartridge and twice the capacity of the double-density cartridges used on previous 3995 optical devices. The 3995 quad-density (2600 MB), WORM or rewritable optical disk cartridge is used in support of the new 3995 devices.

Functional Components Enhanced
DFSMSdfp (OAM)

Software Dependencies
If you are running SMS in a shared sysplex environment with other DFSMS/MVS system levels and you have 3995-Cxx models defined, you must apply PTFs to all the DFSMS/MVS systems prior to installation of this release to allow the 3995-Cxx models to have zero drives defined to SMS.

Hardware Dependencies
3995-Cxx series Optical Library Dataserver. If you are at the MVS/ESA 5.2 level or higher, you need PTF maintenance for HCD to enable 3995-SDA HCD/UIM support.

Coexistence Issues
There are no coexistence issues.

Program Interfaces Affected
No programming Interfaces are affected.

Other Migration Considerations
Refer to *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support* for migration considerations.

Where to find more information

DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support

OAM Volume Stacking and the Virtual Tape Server Subsystem

DFSMS/MVS supports the virtual tape server (VTS) subsystem in an automated tape library (Magstar 3494 Tape Library) at the DFSMS/MVS 1.1 and subsequent release levels. This subsystem uses virtual tape drives and virtual tape volumes that emulate 3490-type tape devices and Cartridge System Tape or Enhanced Capacity Cartridge System Tape media, or both, to the host system for processing purposes. Data is read and written as if it is stored on Cartridge System Tape or Enhanced Capacity Cartridge System Tape; however, within the subsystem it is really stored on DASD.

The virtual volumes and the data associated with them are stored in the tape-volume cache when they are being used by the host system. The tape-volume cache consists of a high performance array of DASD and control software. The tape-volume cache on the VTS enables utilization of 3590 tape technology.

When a virtual volume is moved from tape-volume cache to a 3590 cartridge, it becomes a logical volume. Physical 3590 cartridges are used for these logical volumes just to exploit the 3590 tape technology and storage capacity. To the host system, these cartridges look like Cartridge System Tape or Enhanced Capacity Cartridge System Tape media. A 3590 cartridge that contains logical volumes is referred to as a stacked volume.

As virtual volumes are moved from the tape-volume cache (becoming logical volumes), they are stacked end to end on the cartridge taking up only the number of bytes written by the host, thereby, effectively utilizing all of the storage capacity of the 3590 cartridge. The VTS stacks multiple host created volumes onto a 3590

cartridge to create a stacked volume. The 3590 volumes used in a library as stacked volumes are identified through their volume serial numbers.

When a stacked volume is inserted into a library, it becomes part of the volumes managed by the VTS and is not reported to the host as a newly inserted volume. By buffering host-created volumes, then later stacking them on a 3590 cartridge, the cartridge capacity of the 3590 technology is fully utilized.

Functional Components Enhanced
DFSMSdfp (OAM)

Software Dependencies
There are no software dependencies.

Hardware Dependencies
The virtual tape server subsystem in an automated tape library (Magstar 3494 Tape Library).

Coexistence Issues
There are no coexistence issues.

Program Interfaces Affected
No program interfaces affected.

Other Migration Considerations
There are no migration considerations.

Where to find more information

For more information regarding volume stacking and the VTS, refer to *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

OPEN/CLOSE/End-of-Volume Serviceability Improvements

OPEN/CLOSE/End-of-Volume (O/C/EOV) serviceability improvements include:

- Started task for problem determination work area trace
- Problem determination trace that provides information for tape management system tape exits
- Work area extension above 16MB line
- Problem determination trace that formats additional fields in the O/C/EOV work area and new work area extension
- Abend message that displays the name of the module that detected an error
- Message IEC813I generated when abend 213 reason code 30 is issued
- LOGREC record written for ATL devices whenever a 613 or 637 abend occurs
- SMF type 42 subtype 9 record written for B37/D37/E37 abends

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues

There are no coexistence issues.

Where to Find More Information

DFSMS/MVS DFSMSdfp Diagnosis Reference
OS/390 MVS System Management Facilities (SMF)
OS/390 MVS System Messages, Vol 4 (IEC-IFD)

Program Management Extensions

Program management has been extended in this release to include support for dynamic linking and loading and support for DLLs.

The program management binder's role is to bind program sections into modules, and save them in program libraries or prepare them for immediate execution. The binder API continues to provide support for calling programs to access module data and to construct, bind, or copy modules. The binder is further enhanced in this release to include functions provided by the LE/370 Prelinker Utility.

The binder now excepts, as input, the object text files from the C and C++ compilers. This eliminates the need for the intermediate pre-link job step between the compilation and binding of a C or C++ module when the target SYSLMOD library is a PDSE. The binder also creates a new level or format of the program object for this release (P03). P03 can consist of multiple text classes and associated information that supports DLL processing.

The program management loader has changed to include support for the deferred binding of program objects until execution. This release also provides three new program management utilities and changes to DE services to support new extensions of the DESERV interface.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

OS/390 Release 4

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

When the program management binder stores a program object (PO) in a PDSE, it will store it using the the highest program management (PM) format by default. For example, the binder running at level PM3 will store a program object using format PO3. Earlier levels of the binder and program loader do not support new PM formats.

Program objects that will be loaded on systems with earlier PM levels should either be bound on those systems or, if the new level of the binder is used, they must be bound using the COMPAT option. The COMPAT option is specified as COMPAT=xxx, where xxx is the PM level corresponding to the lowest level system on which the program object can be loaded. The PM levels are:

PM1 For DFSMS/MVS 1.1 and DFSMS/MVS 1.2

PM2 For DFSMS/MVS 1.3

PM3 For DFSMS/MVS 1.4

You can specify COMPAT in these ways: By including the COMPAT parameter on the EXEC statement in JCL; by including it in the installation options for the binder; by modifying CSECT IEWBODEF in load module IEWBLINK.

During API processing for "intent access" a module will be saved in the same format it had on input if followed by a copy operation. During API processing for "intent bind" (and both libraries are PDSEs), the module will be saved as a PO3 program object unless overridden with the COMPAT option.

PO3 program objects do not support overlay format. The binder will automatically produce a PO1 version of the program object if overlay is requested and the SYSLMOD data set is a PDSE.

Programming Interfaces Affected

DE Services supports new extensions to the DESERV interface:

- DESERV FUNC=GET supports the SUBPOOL and AREAPTR parameters.
- DESERV FUNC=GET_ALL supports the CONCAT(ALL) parameter.

In support of these changes, the interface to the DESERV exit has changed. For information on possible updates you should make to the DESERV exit defined for your system, refer to *DFSMS/MVS DFSMSdfp Advanced Services*.

Utilities Added

The following utilities are added to support the new functions:

IEWTPORT Transport Utility
IEWBFDAT(Fast Data Utility)
IEWDLIST (DLL List Utility)

Where to Find More Information

DFSMS/MVS Program Management
DFSMS/MVS DFSMSdfp Advanced Services
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS DFSMSdfp Diagnosis Reference
DFSMS/MVS Using Data Sets
DFSMS/MVS Macro Instructions for Data Sets
OS/390 MVS System Messages, Vol 4 (IEC-IFD)
OS/390 MVS Diagnosis: Tools and Service Aids

Sequential Access Method Tailored Compression

Tailored compression introduces a new form of compression for sequential extended format data sets. With tailored compression, the system tries to derive a compression dictionary that is tailored specifically to the initial data written to a data set. Once a tailored dictionary is derived, it is imbedded in the compressed data set. This technique is expected to provide improved compression ratios, thereby, reducing DASD usage.

Since the dictionary is tailored to the user data, significantly more data is sampled than is done by generic compression. The process of sampling the data and

building the dictionary during creation of a new data set is costly in terms of CPU cycles and is, therefore, most noticeable when compressing small data sets. For larger data sets, the cost of sampling is amortized significantly. Reuse of a tailored compressed data set saves the cost of sampling and dictionary creation. Input operations are expected to perform better than with generic compression.

Functional Components

DFSMSdfp

Software Dependencies

There are no software dependencies.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

Data sets created with tailored compression are accessible only on systems with DFSMS/MVS 1.4 or later releases. Toleration PTFs are required for prior releases of DFSMS/MVS, to fail an attempt to access such a data set via an OPEN SVC, DFSMSdss, or DFSMShsm functions.

Performance Characteristics

Tailored compression will improve DASD usage.

SMS Changes

A new parameter, COMPRESS(TAILORED|GENERIC) is available for the IGDSMSxx member of SYS1.PARMLIB which allows the user to specify the form of compression the system is to use for newly created sequential extended format data sets. COMPRESS(GENERIC) refers to generic DBB based compression which was introduced in DFSMS/MVS 1.2. This is the default. COMPRESS(TAILORED) refers to tailored compression where the system attempts to derive a dictionary tailored specifically to the initial data written to the data set.

For a complete description of this parameter, refer to *DFSMS/MVS DFSMSdfp Storage Administration Reference*.

System Interfaces Affected

To request tailored compression, specify the parameter COMPRESS(TAILORED) in the IGDSMSxx member of SYS1.PARMLIB. Once this member is activated, (via SET SMS=xx command or an IPL), the system will use tailored compression for newly created sequential compressed data sets.

Other Migration Considerations

During data set creation when tailored compression is in effect, compression services require approximately 3MB of storage to perform dictionary creation. The majority of this storage (>2.5MB) is released once the dictionary is created.

Where to Find More Information

DFSMS/MVS Access Method Services for ICF
DFSMS/MVS DFSMSdfp Diagnosis Reference
DFSMS/MVS Macro Instructions for Data Sets
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Using Data Sets

Storage Management Subsystem Out-of-Space (X37 Abend) Reduction

Users occasionally encounter data set allocation or extension failures (X37 abends) because there is not enough space available on a volume to satisfy the request. SMS alleviates this situation to some extent by performing volume selection, checking all candidate volumes before failing an allocation.

Out-of-space conditions are now further reduced for new volume processing of SMS-managed data sets. VSAM and non-VSAM data sets can now acquire up to 123 extents instead of just 5 extents on a volume. Multivolume VSAM data sets can now have a maximum of 255 extents across volumes for each component, but no more than 123 extents per volume.

Two new parameters, *SPACE CONSTRAINT RELIEF* and *REDUCE SPACE UP TO (%)*, are added to the SMS data class definition for this support.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
Compatibility PTFs are required to accommodate the increase in the maximum number of extents for a VSAM component from 123 to 255. With these PTFs applied, lower level systems can access all 255 extents of a VSAM component created on a DFSMS/MVS 1.4 system.

ISMF Changes
ISMF Data Class panels contain two new fields:

- Space Constraint Relief
- Reduce Space Up To (%)

IDCAMS Changes
For allocations initiated by IDCAMS DEFINE or ALLOCATE, IDCAMS recognizes the new data class attributes, *SPACE CONSTRAINT RELIEF* and *REDUCE SPACE UP TO (%)*.

Other Migration Considerations
Users who specify *SPACE CONSTRAINT RELIEF* in one or more data classes should note the following:

- Very large allocations that would have failed previously could succeed if a sufficiently large volume count is specified in the data class or in the JCL.
- Existing data sets could end up with less space than requested on extends due to the changes in a data class or in the active configuration.
- The space allocated for new data sets could be less than requested depending on the usage of new attributes in the data class.
- The number of extents used during initial allocation could result in fewer extensions to new volumes or even current volumes.
- Overall, there should be fewer X37 abends.

Where to Find More Information

DFSMS/MVS DFSMSdfp Diagnosis Reference
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
OS/390 MVS System Messages, Vol 2 (ASB-EWX)
OS/390 MVS System Messages, Vol 3 (GDE-IEB)
OS/390 MVS System Messages, Vol 5 (IGD-IZP)

VSAM Catalog Search Interface

The catalog search interface provides a high speed application programming interface (API) from assembler and high-level language programs to catalog information. It is an alternative to IDCAMS LISTCAT which allows data extraction using filtering criteria of the output. The catalog search interface also provides some information that LISTCAT does not provide.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no software dependencies.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

The catalog search interface can be used on earlier releases by including module IGGCSI00 in the bind for the user application.

Programming Interfaces Affected

The catalog search interface is a general-use programming interface that can be invoked by a user application written in assembler or a high level language.

In an assembler program, the catalog search interface module IGGCSI00 can be invoked by a LINK macro, by the LOAD and CALL macros, or by a CALL statement that results in a V-type address constant.

Using a high level language, the catalog search interface can be invoked by using instructions that translate to one of the assembler language invocations described above.

Upon invocation, the user application supplies a parameter list that includes the selection criteria for locating information in the catalog. A generic filter key is used to specify either a fully-qualified name to obtain information for one entry, or a partially-qualified name to obtain multiple entries from the catalog.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Managing Catalogs
OS/390 MVS System Messages, Vol 2 (ASB-EWX)
OS/390 MVS JCL Reference

VSAM DCOLLECT Enhancements

IDCAMS DCOLLECT now provides the volume serial number for the volume where a data set resided before it was migrated or backed up. Only the volume serial number for the first volume of the data set is included in the DCOLLECT report.

A new parameter has also been added to the DCOLLECT command to give the user a method for limiting the amount of output generated for the DCOLLECT report. The new parameter, EXCLUDEVOLUMES, allows the user to eliminate volumes identified by the STORAGEGROUP and VOLUMES parameters; these parameters allow the user to specify a set of volumes for which data is to be collected.

In addition, DCOLLECT reports new VSAM association information and also now reports actual device type information, if available, rather than the generic 3380/3390 type device information.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no software dependencies.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

There are no coexistence issues.

IDCAMS Changes

IDCAMS DCOLLECT has a new optional parameter, EXCLUDEVOLUMES to specify the volumes to be excluded from the collection.

IDCAMS DCOLLECT output provides the following new information:

- The new field, UMFRVOL, contains the first DASD volume that was the original source of a migrated data set. UBFRVOL is used for a data set that is backed up.
- VSAM Association Information (record type A) contains two new fields DCACISZ and DCACICA for control interval and control area size information.
- Volume Information (record type V) now reports the actual physical device type rather than the generic 3380/3390, which is the logical device type.
- Data Class Information (record type DC) contains a new bit DDCRABS for record access bias, and a new field, DDCRBIAS for the VSAM record access bias constant.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Managing Catalogs
OS/390 MVS System Messages, Vol 2 (ASB-EWX)
OS/390 MVS JCL Reference

VSAM Last-Referenced Date at Close Processing

The last-referenced date (LRD) for a VSAM data set is now updated in the VTOC format 1 DSCB of the first volume at the time the data set is closed. Previously, the LRD for the base data component of a VSAM sphere was updated only at open time. Updating at close time helps to prevent the premature migration of data sets that can occur due to the elapsed time between open and close processing.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Programming Interfaces Affected
There are no changes to user interactions for this support. The IDATMSTP routine is still maintained for compatibility.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Managing Catalogs
OS/390 MVS System Messages, Vol 2 (ASB-EWX)
OS/390 MVS JCL Reference

VSAM Load Enhancements

An improvement in load performance can be realized for extended format key-sequenced data sets (KSDS) when the data set is defined with the SPEED option and system-managed buffering is used during the load process.

The number of data buffers is optimized for load mode processing, which is sequential by nature. System-managed buffering monitors the current state of the data set and adjusts to the load mode requirements by regulating the number of buffers it uses, thereby minimizing the number of I/O operations needed.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues for this support.

Other Migration Considerations
To take advantage of the improvements in load performance, users need to define their extended format key-sequenced data sets using SPEED on the

IDCAMS DEFINE CLUSTER and specify system-managed buffering in the associated data class or in the JCL AMP parameter. See “VSAM System-Managed Buffering” on page 113 for more information on ways to specify system-managed buffering for the data set.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Managing Catalogs
OS/390 MVS System Messages, Vol 2 (ASB-EWX)
OS/390 MVS JCL Reference

VSAM LSR Enhancements

VSAM local shared resources (LSR) has increased the maximum number of buffer pools from 16 (0-15) to 256 (0-255).

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues for this support.

Other Migration Considerations
The ACB, BLDVRP, DLVRP, GENCB ACB and MODCB ACB macros are changed to allow a specification of LSR pools from 0 through 255.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Managing Catalogs
OS/390 MVS System Messages, Vol 2 (ASB-EWX)
OS/390 MVS JCL Reference

VSAM System-Managed Buffering

System-managed buffering (SMB) allows VSAM to determine the optimum number of buffers as well as the type of buffer management to use for a data set. System-managed buffering is available for extended format data sets only.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no software dependencies.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

There are no coexistence issues.

ISMF Changes

ISMF data class panels are changed to include a new data class parameter, RECORD ACCESS BIAS.

JCL Changes

Several new JCL AMP parameters and subparameters are added for this support. JCL AMP specifications override any ACB or data class specifications. These parameters include:

ACCBIAS - record access bias
- USER - no change to current processing
- SYSTEM - force system-managed buffering
- SO - system-managed buffering with sequential optimization
- SW - system-managed buffering weighted for sequential processing
- DO - system-managed buffering with direct optimization
- DW - system-managed buffering weighted for direct processing

RMODE31 - virtual storage location of buffers and control blocks
- BUFF - buffers above the 16MB line
- CB - control blocks above the 16MB line
- ALL - control blocks and buffers above the 16MB line
- NONE - control blocks and buffers below the 16MB line

SMBDFR - record management PUT defer processing

SMBHWT - LSR hiperspace weighting factor

SMBVSP - Maximum data buffer space

Other Migration Considerations

VSAM users who wish to take advantage of system-managed buffering for their extended format data sets will need to do the following:

- Set the Record Access Bias to *SYSTEM* in the associated data class or set up the JCL AMP ACCBIAS with one of the following subparameters: SYSTEM, SO, SW, DO, or DW.
- Ensure that the ACB MACRF on OPEN does not contain LSR, GSR, RLS, ICI, AIX, or UBF. NSR is required for the buffer management specification in the ACB (NSR is the default).
- If the Record Access Bias (or ACCBIAS) is set to SYSTEM, also set up the ACB MACRF with SEQ, DIR, or SKP, to specify how the data will be accessed. Optionally, the storage class values of BIAS and MSR can also be specified.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Managing Catalogs
OS/390 MVS System Messages, Vol 2 (ASB-EWX)
OS/390 MVS JCL Reference

Migration Planning for DFSMS/MVS Version 1 Release 3

This chapter presents a technical overview of all new or enhanced functions provided in DFSMS/MVS 1.3:

- Access method services usability enhancements
- BSAM, QSAM and VSAM access to files in a hierarchical file system
- Catalog usability enhancements
- Character Data Representation Architecture services
- Concatenated Parmlib
- Data set account field support
- DFSMSdss multivolume data set allocation enhancement
- DFSMSdss support for IMS Backup-While-Open
- DFSMSdss utility message control
- DFSMShsm CPOOL performance improvement
- DFSMShsm greater than 13 backup versions
- DFSMShsm multitasking recovery
- DFSMShsm general performance, function, and usability improvements
- DFSMSrmm enhancements
- Enhanced LNKLST processing
- Expiration date override user notification
- Extended remote copy support
- Hierarchical file system standards compliance
- Installation control of PDSE hiperspace size
- ISMF usability enhancements
- Licensing compliance changes
- Program management extensions
- SMS 32 name support
- SMS usability enhancements
- Tape mount management enhancements
- Unit control block virtual storage constraint relief
- VSAM extended addressability
- VSAM record-level sharing

For each new type of support, this chapter outlines environmental dependencies and migration considerations and, where appropriate, refers to other books in the DFSMS/MVS and MVS/ESA libraries where you can find more implementation assistance.

Access Method Services Usability Enhancements

Usability enhancements are introduced for several access method services (IDCAMS) commands. The following commands are changed for this support:

- **DCOLLECT:** Output from DCOLLECT provides the following new information:
  - SMS base configuration coupling facility (CF) cache structure names and cache set names associated with VSAM record-level sharing.
  - Number of stripes for striped data sets
  - VSAM candidate amount for a data class
  - Cache set information and weight values for a storage class
  - Volume level reset count for volumes

© Copyright IBM Corp. 1979, 1999

– ACS routine headers that show the routine description, last change date and user ID responsible.

New fields were also added to the mapping macro, IDCDOUT, to enable DCOLLECT to report on the expanded system and system group information for SMS 32-name support. User applications processing DCOLLECT output should be changed to access these new fields.

- **DEFINE USERCATALOG:** The DEFINE USERCATALOG function now defaults to ICFCATALOG instead of VSAMCATALOG.
- **DELETE GDG FORCE:** All SMS-managed generation data sets (GDS) in a generation data group (GDG) can now be deleted, along with the GDG, in a single step using DELETE GDG FORCE. The DELETE GDG FORCE command deletes the SMS-managed entries pointed to by the GDG base in addition to removing the entry from the catalog.
- **REPRO:** The REPRO data set copy function offers more flexibility in error handling to help prevent job failures and unnecessary reruns when numerous key mismatches are encountered. REPRO now allows an end user to specify how many I/O errors should be tolerated before the operation is terminated. The previous limit was four errors.
- **REPRO MERGECAT:** The REPRO MERGECAT function is enhanced to preserve candidate volumes for VSAM data sets when copying an entry from one catalog to another. This eliminates the need for users to issue the IDCAMS ALTER ADDVOLUMES command to reconstruct the candidate volume information after REPRO MERGECAT completes. This capability was already available for non-VSAM data sets.

Many other DFSMS/MVS 1.3 enhancements have an impact on IDCAMS functions and these effects are listed throughout this chapter. Of special note are the following:

- **Data set account field support:** A new data set attribute, the account field, allows you to gather accounting data or general-purpose user data using the IDCAMS DEFINE, ALTER, and LISTCAT functions. For more information see “Data Set Account Field Support” on page 122.
- **LISTCAT output changes for RBA values:** Several fields in LISTCAT output have changed to accommodate larger RBA values as a result of VSAM extended addressability requirements. The following field names are affected:

  FREESPC-BYTES is now FREESPC
  HI-ALLOC-RBA is now HI-A-RBA
  HI-USED-RBA is now HI-U-RBA

  Applications that interrogate LISTCAT output might need to be updated to recognize these new field names.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
These changes do not have any unique software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues

There are no coexistence implications related to these changes.

Control Statements Affected

- **DCOLLECT:** There are no changes in the invocation of the DCOLLECT function for this usability enhancement. However, if you have programs that process output from DCOLLECT, make sure they can handle the longer output records introduced with this support.

  Other new functions that introduce changes in the DCOLLECT output format are described under "Data Set Account Field Support" on page 122, "Storage Management Subsystem 32 Name Support" on page 141, and "VSAM Extended Addressability" on page 157.

- **DEFINE USERCATALOG:** There are no changes in the invocation of the DEFINE USERCATALOG function. The difference is in the default value of ICFCATALOG rather than VSAMCATALOG. The explicit specification of either keyword is still honored. Jobs that previously took the default for a VSAM catalog will need to explicitly specify the VSAMCATALOG keyword to define this catalog type.

- **DELETE GDG FORCE:** There are no changes in the invocation of the DELETE GDG function; however, a user must have authority to the RACF facility profile named STGADMIN.IGG.DELGDG.FORCE to issue the DELETE GDG FORCE command.

  If a user does not wish to scratch SMS-managed non-VSAM data sets pointed to by the GDG base, then DELETE GDG RECOVERY should be used instead. The user must have READ authority to the RACF facility profile named STGADMIN.IGG.DELGDG.RECOVERY to issue the DELETE GDG RECOVERY command.

- **REPRO:** A new parameter, ERRORLIMIT is used to specify a maximum number of errors that REPRO should tolerate before terminating the data set copy function. The ERRORLIMIT default is four, but it can handle up to 2,147,483,647 errors.

- **REPRO MERGECAT:** There are no changes in the invocation of the REPRO MERGECAT function.

Where to Find More Information

DFSMS/MVS Access Method Services for ICF

BSAM, QSAM, and VSAM Access to Files in a Hierarchical File System

Using BSAM, QSAM, or VSAM access methods, MVS applications can access enterprise data in an open system environment. You can use many of the standard BSAM, QSAM, and VSAM interfaces to access data in OS/390 UNIX hierarchical file system (HFS) files. Most applications that use these access methods can access HFS files without reassembly or recompilation.

The following types of HFS files are supported with this function:

regular files
character special files (null files only)
FIFO special files
symbolic links

The system attempts to support an HFS file such that the application program sees the file as if it were a single-volume physical sequential data set residing on DASD (BSAM, QSAM) or as a VSAM entry-sequenced data set (ESDS).

Functional Components Enhanced

DFSMSdfp

Software Dependencies

This function is available with PTFs installed for DFSMS/MVS 1.3. For full-function support, you also need MVS/ESA SP 5.2.2 with maintenance or OS/390.

Hardware Dependencies

There are no hardware dependencies for this support.

Coexistence Issues

An attempt to access an HFS file with BSAM, QSAM, or VSAM on a system prior to DFSMS/MVS 1.3 results in an abend.

System Interfaces Affected

Close processing does not write SMF type 14, 15, or 60 through 69 records for HFS files. DFSMS/MVS relies on OS/390 UNIX System Services to write appropriate SMF records as requested by the system programmer.

There is no catalog support for HFS files; therefore, any requests that result in catalog processing are ignored or rejected.

Programming Interfaces Affected

You can access an HFS file using BSAM, QSAM (DCB DSORG=PS), or VSAM (ESDS only) by specifying PATH=pathname on the JCL DD statement, by issuing an SVC 99, or with a TSO ALLOCATE command.

Both EBCDIC text HFS files and binary HFS files are supported. When allocating an HFS file, use the new FILEDATA parameter to specify whether the file consists of text or binary data. Use the PATHOPTS parameter instead of the DISP parameter to specify file status.

Since HFS files are not actually stored as physical sequential or ESDS, the system cannot simulate all the characteristics of these types of data sets. Because of this, certain macros and services have incompatibilities or restrictions when HFS files are processed. For example, data set labels and UCBs do not exist for HFS files; therefore, any service that relies on a DSCB or UCB for information might not work with these files. Also, with traditional MVS data sets, the system usually maintains record boundaries. This is not true with byte-stream files such as the OS/390 UNIX files.

Most BSAM and QSAM macros that support a physical sequential DASD data set and VSAM macros supporting an ESDS also support an HFS file. Although there are no syntax changes for these macros, some macros either do not support HFS files or process them differently.

For QSAM and BSAM:

- Some DCB and DCBE fields are treated differently when opened for an HFS file.
- CHECK, ISITMGD, READ, SETPRT, SYNADAF, SYNCDEV, and WRITE have slight processing differences for HFS files.
- BSP, CLOSE, FEOV, OPEN, NOTE, POINT, have some incompatibilities when issued against an HFS file.

- PDAB, RELSE, and TRUNC are ignored for HFS files.
- CHKPT is not supported for HFS files.

For VSAM:

- BLDVRP, DLVRP, ERASE, IDALKADD, MRKBFR, SCHBFR, SHOWCAT, VERIFY, and WRTBFR are not supported.
- ACB, MODCB, SHOWCB, TESTCB, RPL, CHECK, CLOSE, EXLST, POINT, have some processing differences.

Processing Restrictions

Processing restrictions associated with BSAM and QSAM are:

- OS/390 UNIX directories and external links cannot be accessed.
- An HFS file cannot be opened for update processing.
- EXCP processing is not supported.
- A spanned record format, DCB RECFM=V(B)S, is not supported.
- DCB MACRF=P for NOTE or POINT is not supported for FIFO, character special files, or if PATHOPTS=OAPPEND is specified.
- NOTE and POINT do not support a file that contains more than 16,777,214 (X'FFFFFE') records.
- In a binary file with RECFM=V(B) or RECFM=U, a POINT to other than the first block in the file results in an abend.
- BSP can only be issued following a successful CHECK for READ or WRITE, NOTE, or CLOSE (TYPE=T) LEAVE request.

Processing restrictions associated with VSAM are:

- Only ESDS data sets are simulated.
- Alternate indexes are not supported.
- No file sharing or buffer sharing, except for multiple readers or for a reader and a writer for FIFO processing.
- Chained request parameter lists (RPLs) are not supported.
- Shared resources and cross-memory requests are not supported.
- Update and backward processing are not supported.
- Direct processing or POINT for FIFO and character special files.
- JRNAD, UPAD, and EXCEPTION exits are not supported.
- Variable length binary records do not retain record boundaries during conversion to a byte stream.
- To specify the maximum record size, the LRECL keyword must be included on the JCL DD statement, the SVC 99, or the TSO ALLOCATE command. Otherwise, the default is 32,767.

Services and utilities that require a valid DSCB or UCB do not support HFS files. These include OBTAIN, SCRATCH, RENAME, PARTREL, PURGE by DSID, and sector convert routines. ISPF Browse and Edit functions also cannot be used on HFS files.

Because HFS files are not cataloged, these access method services for ICF (IDCAMS) commands are not supported: ALTER, DEFINE, DELETE,

DIAGNOSE, EXAMINE, EXPORT, IMPORT, LISTCAT, and VERIFY. The IDCAMS commands REPRO, PRINT, and DEVTYPE are supported, but have some processing differences.

Also be aware that the OS/390 UNIX fork service should not be used while a DCB or ACB is open for an HFS file.

Security Considerations

DFSMS/MVS relies on OS/390 UNIX System Services to handle the security functions. For HFS files, authorization checking must be performed by the file system called by OS/390 UNIX System Services.

SMS Considerations

If a DD statement contains a PATH parameter, DATACLAS, STORCLAS, and MGMTCLAS cannot be specified. ACS routines are not called.

Other Migration Considerations

Before attempting to take advantage of this new function, you should be familiar with the information found in the *OS/390 UNIX System Services User's Guide*regarding HFS files.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS Macro Instructions for Data Sets
DFSMS/MVS DFSMSdfp Advanced Services
OS/390 MVS JCL Reference
OS/390 UNIX System Services User's Guide

Catalog Usability Enhancements

The following changes improve DFSMS/MVS catalog management:

Catalog alias authorization checking: A new RACF facility class is available to authorize users to define or delete aliases related to user catalogs. The new facility class enables a user to define an alias without having update authority to the master catalog or to delete an alias without having alter authority to either the user or master catalog.

Catalog MODIFY command: The MODIFY operator command is enhanced to:

- Allow a volume serial number to be specified as a value from one to six characters for functions that use volume information. Previously, six characters were always required.
- Provide the more meaningful alias, ALLOCATED, for the OPEN function. ALLOCATED requests a listing of all allocated ICF catalogs, not just the catalogs that are open. The OPEN keyword is still supported.
- Issue more descriptive messages when an error is detected in the format of the MODIFY command entered, or in the processing of the command.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no additional software dependencies for this support.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

There are no coexistence implications related to this support.

Security Changes

A new RACF facility class, STGADMIN.IGG.DEFDEL.UALIAS, is implemented for catalog alias authorization checking.

Operator Commands Modified

The catalog MODIFY operator command is changed to accept one to six characters for a volume serial number. ALLOCATED was added as an alias for the OPEN function. Additional error messages are now issued when an error is detected in the use of this command.

Where to Find More Information

DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Managing Catalogs
OS/390 MVS System Commands

Character Data Representation Architecture Services

Character Data Representation Architecture (CDRA) is a methodology for describing character data and certain control characters on any computer system, whether it is in ASCII, EBCDIC, single-byte, double-byte, or multiple-byte representation. This architecture fully describes the character sets and code pages registered and used throughout the world, and provides a means of simplifying the description and identification of these conventions.

A set of callable application programming interfaces (API) for CDRA is now available on the MVS platform and is supported by DFSMS/MVS. These interfaces ease the implementation of national language support (NLS) and can facilitate application development, portability, and use across multiple IBM operating systems.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

The CDRA character and numeric conversion routines require IBM Language Environment support for a successful installation. With MVS/ESA SP 5.2 or higher, this requirement is met if you use either the C/C++ Language support feature of MVS/ESA SP or the Language Environment for MVS and VM program product.

If you do not have Language Environment support, or elect not to use the character and numeric conversion routines provided in this release of DFSMS/MVS, you must change the SMP/E DDDEF statement from its default setting of SCEELKED (the Language Environment linkedit library) to a dummy setting of ACDRSKIP.

If you decide after the DFSMS/MVS installation that you want to use the new conversion routines, install the Language Environment support and then tailor and run CDRARELK in SYS1.SAMPLIB to relink the CDRA modules.

Note: In this case, SMP/E cannot keep the conversion routines at current maintenance levels.

Hardware Dependencies
There are no additional hardware dependencies.

Coexistence Issues
There are no coexistence considerations for this support.

Programming Interfaces Affected
The CDRA API routines are in the callable system services library, SYS1.CSSLIB. CDRA provides a character data encoding identifier (CCSID), which allows you to identify the character sets, encoding scheme, and other characteristics of the data represented. You can associate a given CCSID with the data through IDCAMS ALTER. You can display the CCSID for a data set through the ISMF Data Set Application.

Other Migration Considerations
The *Character Data Representation Architecture Reference and Registry* describes the steps for setting up an MVS application to exploit this support.

Where to Find More Information

Character Data Representation Architecture Overview (GC09-2207)
Character Data Representation Architecture Reference and Registry (SC09-2190)

Concatenated Parmlib

This support provides the capability to concatenate additional data sets to SYS1.PARMLIB at IPL.

Functional Components Enhanced
DFSMSdfp, DFSMShsm, DFSMSrmm

Software Dependencies
This feature requires OS/390 Release 2.

Migration Considerations
See “DFSMShsm General Performance, Function and Usability Improvements” on page 130 for planning considerations.

Data Set Account Field Support

A new data set attribute, the account field, helps you gather accounting data or user data specific to your installation. You can define, update, and read the account field information for an SMS-managed data set with the IDCAMS DEFINE, ALTER, and LISTCAT functions. This field can be used for both VSAM and non-VSAM data sets.

The account field information is stored in the catalog's attribute extension cell. It is intended as a general purpose field for any type of user data and is not checked or used by any MVS/ESA system component.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no unique software dependencies for this support.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

Toleration PTFs are required for any MVS/DFP or MVS/XA DFP system that shares catalogs with a DFSMS/MVS system. See “Shared Catalogs” on page 30 for more information.

Programming Interfaces Affected

A new parameter, ACCOUNT(account_info), is added to the IDCAMS DEFINE and ALTER commands to allow up to 32 bytes of accounting data or other user-defined information.

IDCAMS LISTCAT output is modified to include the account information in the history section of the listing. IDCAMS DCOLLECT also includes the account information in its output. The invocation of LISTCAT and DCOLLECT is not changed for this support.

Where to Find More Information

DFSMS/MVS Access Method Services for ICF

DFSMSdss Multivolume Data Set Allocation Enhancement

DFSMSdss provides new flexibility for allocating multivolume data sets with the addition of an optional VOLCOUNT parameter. VOLCOUNT can be used during data set RESTORE or COPY operations on SMS-managed data sets.

In addition to supporting the current method of allocation for SMS data sets, VOLCOUNT allows the user to:

- Allocate the target data set on the same number of volumes as the source data set
- Allocate the target data set on as many volumes as required, up to a maximum of 59 volumes
- Specify the number of volumes on which to allocate the target data set

DFSMShsm optionally passes the VOLCOUNT keyword and parameter to DFSMSdss, easing allocation during RECALL, RECOVER, or ARECOVER of DFSMShsm-managed data sets.

Functional Components Enhanced

DFSMSdss, DFSMShsm, and DFSMSdfp

Software Dependencies

There are no additional software dependencies.

Hardware Dependencies

There are no additional hardware requirements.

Coexistence Issues

There are no coexistence issues.

Commands Modified

The DFSMSdss COPY and RESTORE commands have been modified with the addition of the optional VOLCOUNT parameter.

Where to Find More Information

DFSMS/MVS DFSMSdss Storage Administration Reference

DFSMSdss Support for IMS Backup-While-Open

DFSMSdss supports Backup-While-Open processing of IMS data sets by providing for concurrent copy dumps. Backup-While-Open serialization is applicable to HISAM, SHISAM, and index (primary and secondary) databases.

IMS requests backup through a DFSMSdss application program interface, as described in the *DFSMS/MVS DFSMSdss Storage Administration Reference*. Backup on an open IMS data set is triggered through a user interaction module (UIM) request, rather than through a Backup-While-Open status or BWO(TYPEIMS) definition that IMS uses.

Functional Components Enhanced
DFSMSdss, DFSMSdfp

Software Dependencies
IMS/ESA Version 6 Release 1, Data Base Manager Feature

Hardware Dependencies
There are no additional hardware requirements.

Coexistence Issues
Data sets created through IMS Backup-While-Open should not be processed with downlevel DFSMSdss software. Data integrity problems could result.

Programming Interfaces Affected
The DFSMSdss application programming interface, which uses the ADREID0 macro, is changed for this support.

Control Statements Modified
IDCAMS DEFINE BWO(TYPEIMS) and ALTER BWO(TYPEIMS) are modified for this enhancement.

Processing Restrictions
DFSMSdss operations COPY with DELETE, DUMP with DELETE and RESTORE with REPLACE could cause irreparable damage to IMS data sets and, therefore, should not be used.

Where to Find More Information

DFSMS/MVS DFSMSdss Storage Administration Reference

DFSMSdss Utility Message Control

DFSMSdss invokes certain utility programs including IDCAMS, IEBISAM, IEHMOVE, and ICKDSF, to perform requested functions. These utilities expect a SYSPRINT data set in which to place their informational and error messages. DFSMSdss provides the keyword parameter UTILMSG= in the PARM field of its EXEC statement to control the disposition of the utility's SYSPRINT data sets.

Functional Components Enhanced
DFSMSdss

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware requirements.

Coexistence Issues
There are no coexistence considerations for this support.

Control Statements Affected
The UTILMSG=YES|NO|ERROR parameter controls the output of messages from auxiliary programs invoked by DFSMSdss (including ICKDSF, IDCAMS, IEBCOPY, IEBISAM, and IEHMOVE) to the DFSMSdss SYSPRINT output. When `YES` is specified, informational and error messages from these auxiliary programs are copied to the DFSMSdss SYSPRINT output. When `NO` is specified, messages are not copied to the output. When `ERROR` is specified, messages are copied only if the auxiliary program returns an error code to DFSMSdss. The default is `ERROR`.

Where to Find More Information

DFSMS/MVS DFSMSdss Storage Administration Reference

DFSMShsm CPOOL Performance Improvement

CPOOL is intended to improve DFSMShsm performance by replacing many GETMAIN and FREEMAIN calls with the invocation of the MVS/ESA CPOOL macro and the use of cell pools. These changes are made for automatic data storage of high-usage modules and frequently created and freed control blocks. This results in a significant reduction in the total number of instructions processed for DFSMShsm.

There are five nonextendable cell pools allocated for the automatic data storage of modules. The DFSMShsm defaults for the number of cells in each are:

Cell pool 1, 200 cells
Cell pool 2 and 3, 100 cells
Cell pool 4 and 5, 50 cells

The CPOOL enhancement also reduces storage fragmentation and storage usage below the 16MB line.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
There are no coexistence issues.

Commands Modified
There are no commands modified.

Other Migration Considerations
There is a new startup procedure keyword in the DFSMShsm SYS1.PROCLIB that allows you to specify the cell pool storage size. The parameter is:

`CELLS=(nnnn,nnnn,nnnn,nnnn,nnnn)`

The variable *nnnn* specifies the number of cells (1 - 9999) that DFSMShsm uses when creating the cell pool for the modules' dynamic storage. The number of cells specified are used to define a table of the cell pool sizes for the installation. A default table is provided, and omission of a parameter will cause DFSMShsm defaults to be used.

Message ARC0019I and the daily statistics record (DSR) provide information to aid the user in determining the proper number of cells in each cell pool.

Performance Characteristics

CPOOL is designed to reduce CPU utilization in the primary address space by reducing the GETMAIN/FREEMAIN activity associated with frequently accessed modules and frequently created and freed control blocks. The overall performance improvement is determined by the number of times modules are invoked and the number of times key control blocks are allocated and freed.

Where to Find More Information

DFSMS/MVS DFSMShsm Implementation and Customization Guide

DFSMShsm Greater than 13 Backup Versions

Prior to DFSMS/MVS 1.3, the maximum number of backup versions of a data set supported by DFSMShsm was 13. With this enhancement, DFSMShsm can support up to 100 backup versions of a data set.

Installations now have the ability to keep a backup version for every day in a month or a quarter. The backup versions can also be used instead of generation data groups.

All data sets currently supported for DFSMShsm incremental backups are supported by this enhancement, including:

- SMS-managed data sets
- Non-SMS-managed data sets
- Uncataloged data sets
- Data sets cataloged in CVOLs or VSAM catalogs

Functional Components Enhanced

DFSMShsm

Software Dependencies

There are no additional software dependencies.

Hardware Dependencies

There are no hardware dependencies.

Coexistence Issues

If DFSMS/MVS 1.3 coexists with prior releases of DFSMS/MVS, or with DFHSM 2.6.0, and more than 13 backup versions are created, there are coexistence considerations. The following terms are used in describing these coexistence considerations:

Local Generations 1-12, which represent the most recent backup versions of a data set.

Distant Generations 13-28. These versions can be referred to in the existing 2040 byte MCB record of the BCDS.

Remote Generations 29-100. These versions require that the MCB record be expanded up to a maximum size of 6544 bytes.

Compatibility maintenance for systems running DFHSM 2.6.0, DFSMS/MVS 1.1, or DFSMS/MVS 1.2, include:

- A PTF to enable recovery of backup versions 14 through 100.
- A PTF to prevent distant and remote backup versions from being deleted during EXPIREBV processing.

  If you use DFSMS/MVS 1.3 to create remote backup versions, EXPIREBV needs to run on the DFSMS/MVS 1.3 system. If you run EXPIREBV with DFHSM 2.6.0, or with DFSMS/MVS 1.1 or 1.2, EXPIREBV processing will bypass any remote backup version data sets encountered.
- A PTF for RECYCLE to support 100 backup versions.
- A PTF to enable the LIST, DCOLLECT, ARCIMPRT, and TAPEREPL commands to support 100 backup versions.

You should also be aware of the following:

- The AUDIT command supports only local and distant backup versions. The command fails if it encounters remote backup versions.
- The BDELETE function supports only local backup versions. BDELETE requests for distant or remote backup versions will fail.
- Backup processing will fail when more than 13 backup versions are specified as a limit for a data set under these functions:
  - BACKDS
  - HBACKDS
  - ARCINBAK
  - ARCHBACK
  - BACKVOL
  - Automatic backup
  - Backup processing on unchanged data processed by the guaranteed backup frequency function

System Interfaces Affected The maximum record length defined for the BCDS determines the maximum number of backup versions that it supports. If the record length is defined from 2040 to 6543, DFSMShsm allows a maximum of 29 backup versions. If the record length is defined as 6544 bytes, DFSMShsm allows a maximum of 100 backup versions. See *DFSMS/MVS DFSMShsm Implementation and Customization Guide* for instructions on how to change the maximum record length of the BCDS.

SMS Changes The management class attributes NUMBER OF BACKUP VERSIONS (DATA SET EXISTS) and NUMBER OF BACKUP VERSIONS (DATA SET DELETED) need to be increased, up to 100, to allow SMS-managed data sets to have up to 100 backup versions.

ISMF Changes Management class DEFINE/ALTER, DISPLAY, and LIST panels have been modified to support 100 backup versions.

The line operator panels for HBDELETE, HALTERDS, and HRECOVER have been modified to support 100 backup versions.

Commands Modified The SETSYS VERSIONS, ALTERDS, HALTERDS, HBDELETE HALTERDS commands have been enhanced to support 100 backup versions.

The FIXCDS command has been modified to allow for a GEN(*nn*) value.

Programming Interfaces Affected The ARCXTRCT and ARCHRCOV macros have been enhanced to support 100 backup versions.

Performance Characteristics Performance of backup or recovery of individual data sets is not affected by this enhancement.

Performance of the EXPIREBV, AUDIT, and DCOLLECT functions are directly proportional to the size of the BCDS, and could be affected if there is an increase in the number of data set backup versions kept. Performance per version record is not affected.

Other Migration Considerations The ARCIMPRT program when used with a sort product such as DFSORT can forward recover DFSMShsm control data sets.

The UPDATEC function has been enhanced to support 6544-byte entries in the DFSMShsm journal.

Where to Find More Information

DFSMS/MVS DFSMShsm Managing Your Own Data
DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Implementation and Customization Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm Multitasking Recovery

DFSMShsm multitasking recovery enhances the performance of both data set recovery and volume recovery from incremental backups. Anywhere from one to 64 individual data set recovery tasks can be performed concurrently, as defined by an installation. Data set recovery can be initiated by data set command processing or by volume command processing.

The multitasking capabilities of individual data set recoveries can speed volume recoveries when incremental data set backup versions are used in the recovery process. The incremental backup data sets are scheduled to be processed by the multitasking data set recovery process rather than the single-task volume recovery process. If more than one volume is being recovered, the incremental data set recovery requests for one volume can be scheduled, and the recovery process can begin immediately on the next volume, decreasing the time needed for the total recovery process.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies

There are no additional hardware dependencies.

Coexistence Issues

There are no coexistence issues.

Commands Modified

The SETSYS MAXDSRECOVERTASKS command allows an installation to define up to 64 concurrent data set recovery tasks.

Performance Characteristics

There can be a significant improvement in data set and volume recovery performance from this support. Multiple data set recovery time and volume recovery time can be reduced.

Up to 64 individual data set recovery tasks can be performed concurrently, reducing the total recovery time for multiple data sets, or for volumes when incremental data sets are part of the recovery process.

When recovering multiple volumes and using incremental backups as part of the recovery process, the recovery function schedules all of the individual data set recoveries for one volume, then moves on to begin recovery processing on the next volume. Total recovery time for multiple volume recoveries can be reduced.

Processing Restrictions

The number of concurrent data set recovery tasks defined is limited by the number of tape drive resources available.

The number of data set recovery tasks that actually run concurrently can be less than specified if the number of backup tape cartridges is fewer than the number of concurrent recovery tasks specified.

Other Migration Considerations

The database administrator needs to perform forward recovery or some other synchronization activity after recovery completes.

The function statistics record (FSR) created by DFSMShsm and written to SMF and to the DFSMShsm log contain indicators to distinguish between requests initiated by data set command processing and requests initiated by volume command processing.

The DFSMShsm REPORT command displays the total number of data sets recovered and the number of bytes processed, but doesn't distinguish between requests initiated by data set command processing and requests initiated by volume command processing.

Where to Find More Information

DFSMS/MVS DFSMShsm Managing Your Own Data
DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Implementation and Customization Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm General Performance, Function and Usability Improvements

There are many general improvements in DFSMShsm performance, function, and usability in this release of DFSMS/MVS. A summary of these changes is provided in this section.

Improvements to performance include the following:

- **Multiple buffer usage when writing to or reading from ML1 DASD:** This improvement enables DFSMShsm to use multiple buffers, in extended private storage, when accessing DFSMShsm owned DASD. This support is used only when the optimum DASD blocking (ODB) option is in effect and the DASD buffer patch (at offset X'390' in the MCVT) is not in use. Optimum DASD blocking is now the DFSMShsm default.

  Access to backup and dump VTOC copy data sets is similarly improved.

- **Improved space checking for Tape Mount Management (TMM):** With this improvement, the hourly space checks are done on the primary host, even when primary space management is being performed on the primary host. This enables TMM to be processed more expeditiously on the other hosts.

  Note: When hourly space checks are done on the primary host, the space value is available to other hosts by being stored in the shared CDS volume record.

Improvements in DFSMShsm functions include the following:

- **Return priority installation exit (ARCRPEXT):** This new installation exit enables installations to assign relative priorities to wait-type recalls. This allows installations to give recalls for specific applications priority over recalls for specific users.

  The ARCRPEXT exit is also taken for data set and volume recovers, and can affect their respective relative priorities.

- **Shutdown installation exit (ARCSDEXT):** This new installation exit is invoked during DFSMShsm shutdown. The exit is particularly useful if you have used the DFSMShsm installation initialization exit (ARCINEXT) and have collected information that you want to print or save on DASD.

- **Volume dumps using concurrent copy:** By using the new SETSYS keyword VOLUMEDUMP(CC), you can request that DFSMShsm cause DFSMSdss to use concurrent copy when dumping volumes. This keyword has an effect only when the hardware and microcode levels support the concurrent copy feature.

- **Overflow ML1s reserved for (H)BACKDS output:** A new subcategory of ML1 volumes is available to relieve the problem of not enough ML1 space to contain larger data-set backups. If you choose, you can specify one or more ML1 volumes as an OVERFLOW ML1.

Improvements to usability include the following:

- **Enhanced generic HBACKDS data set backup:** When requesting cataloged data sets to be processed, an expanded generic filter specification is being used for the five user commands: HBACKDS, HRECOVER, HMIGRATE, HRECALL and HDELETE. The special characters %, **, and multiple instances of * are now accepted. Also, an alias can be used in place of a fully-qualified

name. A new keyword of CHANGEDONLY is added to enable users to limit HBACKDS processing to only those data sets that have been changed.

Only integrated catalog facility (ICF) catalogs are supported.

- **SWAPLOG PDA command:** The storage administrator will now be able to switch the PDA trace recording log by issuing a 'SWAPLOG PDA' command instead of having to remember the offset for using the PATCH command. The command can be abbreviated to 'SWAP PDA'. To switch both the DFSMShsm LOG recording and the PDA trace recording, issue 'SWAP PDA LOG'.

  Another improvement is that if PDA tracing is idle, a switch occurs without unnecessary recording just to cause the switch to happen.

- **Observing interval migration:** To assist users in determining whether interval migration processing might be looping, interval migration activity recorded in the migration activity log will be surrounded by new start and end messages, ARC0580I and ARC0581I. This is particularly useful for tape mount management users, where one invocation of interval migration might last for more than an hour, and the next invocation begins immediately after the first finishes.

- **Specifying DFSMShsm environment as single host at start up:** There are situations where DFSMShsm performs unnecessary I/O overhead in protecting its control data sets (CDSs). These situations can occur in both exception and normal processing environments.

  A typical exception processing environment situation occurs during the merging of the sets of DFSMShsm CDSs from multiple computing centers, with the result being a multihost computing center. A portion of the processing involves DFSMShsm doing CDS updates, and typically from only one host. If you are in this situation, you can avoid unnecessary CDS I/O overhead by specifying that DFSMShsm single-host serialization is adequate for the DFSMShsm start up where these updates are performed.

  A typical normal processing environment situation exists when the migration control data set (MCDS) resides on a shared DASD volume, but only one DFSMShsm host ever accesses the CDSs. This improvement enables you to specify the DFSMShsm environment as either single host or multihost in the DFSMShsm start-up procedure.

- **Migrate prefix "HSM" and "SYS1":** DFSMShsm now allows the user to use the SETMIG LEVEL command to deactivate the default retain list elements (RLE) that prevent space management from being performed on data sets with a high level qualifier of "HSM" or "SYS1." If you deactivate the default RLE, you must be very certain that you don't migrate data that is needed during the time before DFSMShsm is started. It is expected that SMS management classes would exclude nearly all of the data sets that these two RLEs currently exclude.

- **Concatenated Parmlib Support:** DFSMShsm supports usage of the MVS concatenated parmlib function. The concatenated parmlib function allows up to 10 data sets to be used to create a logical parmlib. This provides greater flexibility in managing parmlib members, and handling changes in member formats.

  DFSMShsm also continues to support the use of HSMPARM DD statements. If an HSMPARM DD statement is supplied through JCL, the concatenated parmlib function is not used. If an HSMPARM DD statement is not supplied, the MVS concatenated parmlib function is used, if available.

- **Year 2000 Support:** DFSMShsm year 2000 support allows users to specify dates beyond 1999. The support adds restrictions to the way dates are specified in commands for the years both before and after 1999.

  Commands affected by the year 2000 support include:

  – The DEFINE command CYCLESTARTDATE parameter

  – The DEFINE DUMPCLASS command TAPEEXPIRATIONDATE parameter

  – The RECOVER and HRECOVER commands DATE parameter

  – The REPORT command FROMDATE and TODATE parameters

  – The SETSYS CDSVERSIONBACKUP command BACKUPDEVICECATEGORY(TAPE) parameters

  – The TAPECOPY command EXPDT, RETPD, and INDATASET parameters

  Also affected by year 2000 support are:

  – The ARCPRPDO program START and END parameters

  – The ARCHRCOV macro DATE parameter

Where to find more information:

DFSMS/MVS DFSMShsm Managing Your Own Data
DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Implementation and Customization Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMSrmm Enhancements

DFSMSrmm extends its capabilities to include:

- Back up of the DFSMSrmm control data set using DFSMSdss concurrent copy to minimize impact to the tape environment
- Threshold support for the DFSMSrmm journal data set which can be used to automate submission of DFSMSrmm control data set back up
- Enhanced reporting:
  – Sample reports using DFSORT's ICETOOL
  – New standard reports generated from the report extract file
- Use of a new graphical user interface (GUI) that gives DFSMSrmm the look and feel of ISPF Version 4 Release 2 panels
- Concatenated parmlib support
- Enhancements for initializing and erasing volumes
- Enhancements for running DFSMSrmm with JES3
- New support for creating sticky labels
- New support for storing JOB or STEP accounting information
- Installation control over the use of DFSMSrmm subcommands
- New support for managing data sets and volumes

Functional Components Enhanced
DFSMSrmm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues
The data set record in the DFSMSrmm control data set is increased in size and a coexistence PTF is provided. The PTF must be applied to all of the systems sharing a control data set.

Performance Characteristics
Using DFSMSdss concurrent copy allows updates to the DFSMSrmm control data set during back up of the control data set.

Programming Interfaces Affected
New and Changed macros since the last release: EDGRDEXT, EDGRREXT, and EDGRVEXT

Where to Find More Information

DFSMS/MVS DFSMSrmm Command Reference Summary
DFSMS/MVS DFSMSrmm Diagnosis Guide
DFSMS/MVS DFSMSrmm Guide and Reference
DFSMS/MVS DFSMSrmm Implementation and Customization Guide

Enhanced LNKLIST Processing

This enhancement allows up to 255 extents for linklist data sets. You can specify that libraries other than SYS1.LINKLIB, SYS1.MIGLIB, and SYS1.CSSLIB be placed first in the LNKLST concatentation during IPL.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
This feature requires OS/390 Release 2.

Other Dependencies
There are no unique dependencies for this enhancement.

Expiration Date Override User Notification

When you specify OVRD_EXPDT(YES) in the IGDSMSxx member of SYS1.PARMLIB, the system ignores expiration dates for SMS-managed DASD data sets, allowing the data sets to be deleted using JCL, IEHPROGM, ISPF D or DEL line operators, the SCRATCH macro, or through dynamic unallocation. Although this function was available with DFSMS/MVS 1.2, it did not provide for user notification when data sets were deleted as a result of administrator override.

SMS now issues message IGD17042I to notify a user, in the job log, when an expiration date for a data set is overridden during delete processing.

Refer to “Overriding Expiration Dates When Deleting Data Sets” on page 190 for migration considerations related to this support.

Extended Remote Copy Support

Extended remote copy (XRC) is a disaster recovery and workload migration solution that relies on a combination of cached storage subsystems and DFSMS/MVS.

XRC is an optimal performance choice for shadowing your critical application volumes to a remote storage subsystem. XRC automatically sends copies of updated data to a remote recovery system with almost no impact to the operations of the application system. XRC is also an efficient tool for migrating data from one set of DASD volumes to another set, with minimal impact to applications.

To implement remote copy, you need to establish two systems: an application system at one location and a recovery system at another location. Each system has specific storage devices dedicated to handling the data you have identified as remote copy-managed. Once established, remote copy makes changes to your data on the remote storage subsystem to mirror the changes you make at your application location. If your application system fails, recovery involves a takeover by the recovery system.

Functional Components Enhanced
DFSMSdss, DFSMSdfp

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
When migrating data with XRC, the primary storage device must be attached to a remote copy-capable storage subsystem.

Coexistence Issues
There are no coexistence issues.

Where to Find More Information

DFSMS/MVS Remote Copy Administrator's Guide and Reference

Hierarchical File System Standards Compliance

The OS/390 UNIX hierarchical file system (HFS) is now compliant with the Universal UNIX** (UU) standards and the X/OPEN Portability Guide Issue 4 (XPG4) standards. This enables an installation to have an open system and Universal UNIX system with the capacity, reliability, and storage management capabilities of an MVS/ESA system.

DFSMS/MVS also introduces changes that make it possible for VSAM, BSAM, and QSAM programs to sequentially access UNIX data stored in an HFS environment on MVS/ESA systems.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
The support is dependent on the latest release of OS/390 UNIX.

Hardware Dependencies
There are no hardware dependencies.

Coexistence Issues

File systems created by DFSMS/MVS 1.3 can coexist with file systems created by previous releases of DFSMS/MVS.

This HFS support is dependent on the latest release of OS/390 UNIX; however, the support can coexist with previous releases of OS/390 UNIX and can operate in a compatible manner.

Programming Interfaces Affected

There are no programming interfaces changed for this support.

Other Migration Considerations

The following HFS statistics are now available:

- **File statistics:** HFS files created prior to DFSMS/MVS 1.3 do not have a physical block count stored in their attributes. For those files, the physical block size is always the file size divided by 4096, which is accurate for all but sparse files. For new files created in DFSMS/MVS 1.3, the physical block size is stored correctly, even for sparse files.
- **File system statistics:** On HFS file systems created prior to DFSMS 1.3, the number of allocated file nodes was not stored. Therefore, f_files, and the fields calculated from f_files (f_freee and f_favail), are not available. If older files are copied to a new file system created on DFSMS/MVS 1.3, these statistics will become available.

Where to Find More Information

OS/390 UNIX System Services Planning

Installation Control of PDSE Hiperspace Size

DFSMS/MVS installations can now limit the amount of expanded storage used for PDSEs, thereby increasing the amount of expanded storage available to CICS or other applications that use hiperspace buffers. This can potentially decrease paging activity on the system.

Since the introduction of PDSEs, the size of the hiperspace created by the system to cache PDSE member pages has been fixed at 256MB. An installation can now specify a hiperspace size of up to 512MB, or it can specify that the hiperspace not be created at all. The hiperspace is created at IPL time and, therefore, a re-ipl is necessary to change its size.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no unique software dependencies for this support.

Hardware Dependencies

Expanded storage is required to create the hiperspace.

Coexistence Issues

Systems with releases prior to DFSMS/MVS 1.3 do not recognize the IGDSMSxx parameter in SYS1.PARMLIB used for this support.

System Interfaces Affected

A new optional parameter, HSP_SIZE, is added to the IGDSMSxx member of SYS1.PARMLIB to specify PDSE hiperspace size of up to 512MB or to

indicate that the hiperspace should not be created. If HSP_SIZE is not specified, the hiperspace defaults to half of expanded storage or 256MB (whichever is less), as long as expanded storage is available.

Performance Characteristics

Performance remains the same as previous releases if the HSP_SIZE parameter is not specified or if an HSP_SIZE of 256MB is specified (the default). If the value specified for HSP_SIZE is less than what is optimal given normal PDSE hiperspace usage, PDSE performance could be degraded. If the value specified for HSP_SIZE is more than what is needed for normal PDSE activity, and there is contention for expanded storage on the system, performance of other non-PDSE components or applications could be degraded.

The effectiveness of the PDSE hiperspace can be evaluated through the analysis of SMF type 42, subtype 1 records.

Where to Find More Information

DFSMS/MVS DFSMSdfp Storage Administration Reference

ISMF Usability Enhancements

ISMF has a new graphical user interface (GUI) that gives ISMF panels the look and feel of ISPF Version 4 Release 2 panels. The panels can be displayed as an OS/2 window. ISMF application panels are also more standardized for the different DFSMS/MVS functions, making it easier for ISMF users to move between data, storage, and device management tasks.

ISMF also introduces a new menu-driven function for printing lists generated by the various ISMF applications.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

The GUI functions available with this support require ISPF Version 4 Release 2.

If the GUI function is not used, this support requires a minimum of ISPF Version 3 Release 5 with a toleration PTF.

Hardware Dependencies

There are no dependencies on hardware products.

Coexistence Issues

There are no coexistence implications related to these changes.

ISMF Changes

Enhancements to the ISMF panels include action bars and pull-down menus (for GUI mode); field-level help for input fields; and cursor or mouse sensitivity. Application selection panels have radio buttons and check boxes, making it easier for users to quickly choose their options. ISMF panels have also been converted to use the Dialog Tag Language.

The new LISTPRT function enables users to print lists generated under various ISMF applications. The LISTPRT command is supported in the list panels of all ISMF applications and in the List Application. It acts on the list

that is being viewed or on one that has been saved using the SAVE list command. The types of lists that can be printed include the following:

- Data set lists
- DASD volume, optical volume, and tape volume lists
- Data class, management class, storage class, and storage group lists
- Optical drive, optical library, or tape library lists
- Aggregate group lists

Where to Find More Information

DFSMS/MVS Using ISMF

Licensing Compliance Changes

The DFSMS/MVS licensing compliance mechanism is enhanced to:

- Eliminate the need for you to IPL the system each time you modify SYS1.PARMLIB member IGDDFPKG; for example, to activate a new DFSMS/MVS functional component.
- Allow you to enable optional features found on your DFSMS/MVS product tape.

These changes are described in “Licensing and Enablement” on page 10.

Program Management Extensions

Program management offers many new features including enhanced function for newer language products, such as C++, FORTRAN/90, and High Level Assembler. These improvements encompass changes to the program object, and to the binder, loader, and utility programs that support program objects:

- **Program objects:** Program objects now support long names, unlimited data classes and distributed loading. External names, labels and references can be up to 1024 bytes in length. Data classes are expanded beyond the eight binder-defined classes to include compiler-defined and application-defined classes. Distributed loading makes it possible for the loader to place different parts of module text into noncontiguous storage locations, similar to the scatter loading approach used for the MVS nucleus. To improve disk utilization and reduce I/O, uninitialized pages are now removed from program objects.
- **Binder:** The binder extends the support of program object changes and provides additional function as well. Long external names, up to 1024 bytes, are allowed on all parameters, function calls, control statements and listings. The new generalized object file format (GOFF), accepted as an alternative to the traditional object module, supports long names, multiple text classes, and associated data (ADATA) received from language products. ADATA allows compilers and other applications to save processing, historical, and administrative information with a program object.

  There are several new binder options available along with a new application program interface (API) to access data from program objects. A replaceable binder module allows a system programmer to reset installation defaults for most binder options.
- **Loader:** Two new features are added to the loader. It now supports program objects with uninitialized pages removed. It reinstates those pages during

loading and initializes them with the FILL byte, if that option was specified during the binding process. The loader also supports distributed loading of module text, where each unique text class can be loaded at a different storage location.

- **Directory entry services:** Directory entry services (DESERV) for PDSE data sets are now available through a user interface, enabling a user to list, update, and delete PDSE directory entries. Alias names up to 1024 bytes long are supported.
- **Utility programs:** IEBCOPY, IEWTPORT, IEHLIST, IGWSPZAP and AMBLIST are enhanced to support the new program object format and, except for IEBCOPY, long names. A new utility, IEWBFDAT, provides quick access to program object data without the need for a binder dialog.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no additional software dependencies.

Hardware Dependencies

There are no new hardware features required for these changes.

Coexistence Issues

Program management functions available in DFSMS/MVS 1.1 and 1.2 are upward compatible with the current release. New and old program object formats can reside in the same PDSE program library.

Program objects created on a DFSMS/MVS 1.3 system cannot be processed by prior releases of DFSMS/MVS. DFSMS/MVS 1.1 and 1.2 toleration PTFs enable these earlier versions of the binder and loader to issue appropriate messages and terminate if they encounter a higher level program object. A compatibility option (COMPAT) is available in the binder to force a generated program object into the pre-DFSMS/MVS 1.3 format for processing on down-level systems. Program objects cannot be processed by the linkage editor or batch loader.

The binder does not run on pre-DFSMS/MVS releases. Do not install the binder on DASD volumes shared with pre-DFSMS/MVS systems or copy the binder to those systems in preparation for a software upgrade. DFSMS/MVS does not require the binder for the installation process.

If you need to move program objects to a system without binder support, you must first move or copy all associated modules to a PDS load library, using the binder during this operation.

System Interfaces Affected

A new replaceable binder module, IEWBODEF, enables you to specify system defaults for selected binder options and module attributes. IEWBODEF is a data-only module containing a parameter string identical to that used on the PARM= JCL parameter, without enclosing apostrophes or parentheses. The string can be up to 1024 bytes in length, including comma delimiters. More information on establishing installation defaults for binder options is given in *DFSMS/MVS DFSMSdfp Advanced Services*.

Programming Interfaces and Control Statements Affected

A summary of new and modified interfaces and control statements is included here to help you plan for implementation:

- A new SYSLIN control statement, SETOPT, allows users to set or reset most binder options and attributes during execution. This option permits module attributes to be specified separately for each output module in a batch operation.
- New binder options are:
  - FILL, which lets you define the character byte that fills all uninitialized areas of a module. Setting the fill to non-zeros could be useful for diagnostic purposes.
  - RMODE(SPLIT), which allows you to split program modules into two parts for loading both above and below the 16 MB line, according to the RMODE specified for each section.
  - HOBSET, which instructs the binder to set the high-order bit in each V-type address constant according to the AMODE of the target entry point.
  - COMPAT, which allows for a user-specified compatibility level for the binder. This will restrict the use of certain processing options, alter the behavior of the binder when handling certain error conditions, and influence the format of the generated program object.
- The batch interface to the binder, used for JCL EXEC PGM= and TSO commands, is changed to support the new binder options. No JCL changes are required.
- The general API, or call interface, IEWBIND, is changed to accept longer symbol names and to recognize new class names. A new function call, PUTData, is used to store data into a new or existing workmod element.
- IEWBFDAT is a new API that provides efficient read-only access to class-oriented data in a program object, similar to the GETData function of the binder, without the overhead of a binder dialog. IEWBFDAT does not provide update capability, data selection or other binder functions, but does access module data with a minimum of instructions and virtual storage.
- The loader's IEWLSERV parameters, LOADPT and XTLST, are affected by multipart program objects. You can have multiple load points in a program, but only a single load point parameter in the interface.
- Directory entry services for PDSE data sets provides a new user interface. DESERV is an executable macro that can be used by both authorized and unauthorized programs.
- The AMBLIST utility has a new OUTPUT keyword that is accepted on LISTLOAD control statements. OUTPUT=ADATA causes the contents of ADATA classes to be displayed in a traditional dump format.
- The IGWSPZAP (“superzap”) service aid requires changes to the NAME control statement in support of long names and multipart program objects.

Performance Characteristics

The binder was enhanced to improve processing performance and to reduce the demands on system storage. Binding modules under DFSMS/MVS 1.3 might take less CPU and elapsed time, and use less

DASD storage, than in prior releases. Certain new features, however, might have an adverse affect on performance due to the additional processing they require.

Your system performance could be degraded by up to 1% when multipart, noncontiguous program objects are loaded. DASD storage requirements for a given module could decrease, however, depending on the amount of uninitialized storage contained in the program text.

Processing Restrictions

There are some restrictions on the use of the new fast data access utility, IEWBFDAT. You can use it to obtain data from program objects residing in PDSE program libraries, but not from load modules, object modules or program objects stored in a hierarchical file system. IEWBFDAT provides access to class-oriented data, but without the finer selectivity of the GETEsd function. Update capability is not provided.

Binder support of C language reentrant programs is limited. Although the extended object module produced by C can now be processed by the binder, C programs compiled with the RENT option must be processed first by the C pre-linkage utility.

Other Migration Considerations

No migration is necessary for installations that already use the binder and program management functions available in DFSMS/MVS 1.1 and 1.2. Installations migrating from a pre-DFSMS/MVS release to the full program management support offered by DFSMS/MVS 1.3 need to plan for the following:

- **Copying existing PDS load module libraries into PDSE program object libraries:** You can use IEBCOPY or DFSMSdss COPY to copy programs from a PDS library into a PDSE library. The binder is invoked automatically to perform the conversion between load modules and program objects. You can also use the binder directly to rebind modules for the purpose of migrating them. Where possible, you should take advantage of the data class definitions and cataloged procedure changes to implicitly migrate program modules.

  Note: There is no requirement for copying programs into PDSE libraries. The binder accepts and creates both program objects and load modules.

- **Modifying existing linkage editor and batch loader job streams to invoke the binder:** The binder accommodates MVS/DFP linkage editor job streams with few JCL changes required. The MVS REGION parameter should be increased to a minimum of 2MB, or larger to allow the binder to use storage above the 16 MB line. The PARM field might also need to change for additional options. Two cataloged procedures, LKED and LKEDG, shipped with this support, provide the changes for invoking the binder in batch instead of the linkage editor. The TSO commands LINK and LOADGO also invoke the binder.

  JCL, programs, and cataloged procedures that specify linkage editor and batch loader parameters might require additional changes, since new options have been added to the binder.

Note the following differences and incompatibilities between the binder and the linkage editor:

- Object modules in virtual storage are not accepted as input to the binder.
- Linkage editor option HIAR is not supported by the binder.
- The binder processes errors differently from the linkage editor. It is much less tolerant of data errors. Printed output, including messages, is also different.
- The binder uses more working storage than the linkage editor and can potentially run out of storage due to insufficient data space available or a region size that is too small.
- The binder processes certain control statements and options more logically than the linkage editor and uses different rules of precedence for ordering the statements. Reusability processing has also changed. AMODE and RMODE processing are also slightly different.
- The binder's auto-call sequence is different from the linkage editor.
- When writing a load module, the binder does not initialize the same amount of space as the linkage editor.
- Nonexecutable modules do not replace executable ones by default in the binder.
- The binder does not allow aliases in PDSE libraries to steal member names from other modules.

For users who might be impacted by these operational differences, the linkage editor and batch loader are still shipped, unchanged, with DFSMS/MVS. They can be invoked by specifying entry names HEWLKED or HEWLDIA, respectively, in the PGM= parameter of the JCL EXEC statement.

Where to Find More Information

DFSMS/MVS Program Management
DFSMS/MVS DFSMSdfp Advanced Services
DFSMS/MVS DFSMSdfp Storage Administration Reference

Storage Management Subsystem 32 Name Support

With Storage Management Subsystem 32 name support, you can specify up to 32 system names, system group names, or both system and system group names, in an SMS configuration. This enables you to:

- Define more than eight systems in either a JES2 or JES3 SMS complex. Although JES3 can now manage scheduling for up to 32 systems in an SMS complex, it does not recognize system group names. System groups can be specified only in a JES2 environment.
- Set up system-level unique operations, such as VARY SMS commands, for more than eight systems.
- Have more than eight unique variations with regard to connectivity, status, and workload. For example, you can maintain more object and object backup storage groups, as well as optical libraries that are not shared between systems. You can also specify distinct storage group and volume connectivity for systems handling online transaction processing, different from those handling batch processing.

When a system in an SMS complex begins running DFSMS/MVS 1.3, the constructs in the SMS configuration are automatically expanded to prepare for the additional system information. This change occurs for all DFSMS/MVS 1.3 SMS configurations regardless of whether SMS 32-name support is used.

Initially, the DFSMS/MVS 1.3 system runs in *compatibility mode*, or *8-name mode*, which allows the system to coexist with previous releases of DFSMS/MVS and MVS/DFP. While a system runs in compatibility mode, only eight system and/or system group names can be defined in the SMS configuration. The system can continue to share SMS control data sets with non-DFSMS/MVS 1.3 systems or with other DFSMS/MVS 1.3 systems running in compatibility mode. The system cannot access any control data set that supports more than eight systems and/or system groups.

To convert the system to *32-name mode*, you must update SYS1.PARMLIB member IGDSMSxx and restart SMS. This allows the SMS control data sets to be converted to a new format that supports up to 32 names. This conversion is permanent.

Functional Components Enhanced

DFSMSdfp, DFSMShsm

Software Dependencies

There are no additional software dependencies for JES2 installations.

JES3 users must have JES3 5.2.1 installed, along with compatibility PTFs for JES3 to manage scheduling for the additional systems supported in the SMS complex. The JES3 compatibility PTFs have prerequisites of DFSMS/MVS compatibility PTFs, described in “Coexistence Issues” for this support.

Hardware Dependencies

There are no special hardware requirements for this support. However, to define more than 16 systems in an SMS complex, you must allocate the ACDS and COMMDS on volumes capable of attaching to more than 16 systems, such as devices attached to an IBM 3990 Model 6 Storage Control.

Coexistence Issues

DFSMS/MVS 1.3 can coexist with previous SMS-supported releases provided no more than eight system or system group names are defined in the SMS configuration. Once the configuration is converted to 32-name mode, it cannot be accessed by a DFSMS/MVS 1.3 system running in compatibility mode or by a system running a prior release of DFSMS/MVS or DFP.

When sharing configuration (ACDS) and communications (COMMDS) data sets in a multiprocessor complex, a system must run in compatibility mode until all systems in that complex have DFSMS/MVS 1.3 installed and are ready for simultaneous conversion. All down-level systems and any other DFSMS/MVS 1.3 systems remaining in compatibility mode must form a separate SMS complex.

You must install compatibility PTFs on all pre-DFSMS/MVS 1.3 systems that can share system-managed data in order to:

- Prevent them from accessing a configuration that supports more than eight systems, and potentially damaging the SMS control data sets.

- Allow SMS to retrieve information from a compatibility-mode configuration without receiving incorrect length errors caused by the expanded constructs.
- Allow DFSMShsm or DFHSM to tolerate the increased length in SMS storage group and volume definitions.
- Enable the systems to interpret new error return and reason codes created for the compatibility mode support.

If you have a JES3 installation, you also need to apply JES3 compatibility PTFs to support changes in the parameter list format used on interfaces between SMS and JES3. DFSMS/MVS or MVS/DFP PTFs must be applied prior to the JES3 PTFs.

If, in a critical situation, it becomes necessary to revert to a product level prior to DFSMS/MVS 1.3, be aware that a configuration supporting more than eight systems cannot be used by the system after regression. Also, if you have JES3, you must apply the SMS/JES3 compatibility PTFs on the system you are regressing before you IPL the system to the restored level.

System Interfaces Affected

A new keyword, SYSTEMS, is added to the IGDSMSxx member of SYS1.PARMLIB to specify whether a system should run in compatibility mode (8-name mode) or 32-name mode:

- SYSTEMS(8), the default, indicates that a maximum of eight system and/or system group names can be defined in the SMS configuration. The system runs in compatibility mode, which allows it to share an SCDS/ACDS and COMMDS with systems that are running earlier releases of DFSMS/MVS or MVS/DFP, or other DFSMS/MVS 1.3 systems running in compatibility mode. Essentially, the system continues to operate as it has in the past.
- SYSTEMS(32) indicates that a maximum of 32 names, representing systems, system groups, or both, can be defined in the SMS configuration. The system no longer runs in compatibility mode. The SMS control data sets cannot be shared with down-level systems or any DFSMS/MVS 1.3 system running in compatibility mode. The only exception is an SCDS defined for only eight systems that is activated on the DFSMS/MVS 1.3 system but not converted.

A system continues to run in compatibility mode while SYSTEMS(8) is specified (or the SYSTEMS keyword is omitted) in the IGDSMSxx member of SYS1.PARMLIB. To exploit this new support, you must explicitly specify SYSTEMS(32) in IGDSMSxx and restart SMS, either by IPLing the system or by issuing the SET SMS=xx operator command. This enables SMS to convert the configuration to 32-name mode. This must be done on every system in an SMS complex at the same time. See “Preparing a conversion plan” on page 146

SMS control data sets, when accessed for update, are converted to a new format which supports the additional system and/or system group names. You confirm the conversion through the operator console or an ISMF panel.

Down-level systems or DFSMS/MVS 1.3 systems running in compatibility mode attempting to access the converted SMS configuration or communications data sets will receive an error message.

Table 6 on page 144 summarizes the results when accessing SMS configurations under the two system modes.

Table 6. SMS control data set access results based on system mode.

System Mode	ACDS/COMMDS[1] Format	Result
SYSTEMS(8)	8 systems	No conversion required
SYSTEMS(8)	32 systems	Not supported; message IGD044I issued
SYSTEMS(32)	8 systems	Conversion required before update (define/alter/validate); message IGD064D issued, prompting for conversion. No conversion required for display or list.
SYSTEMS(32)	32 systems	No conversion required

Note 1: This also applies to an SCDS accessed for update through ISMF (for define, alter, delete, or validate).

SMS Changes

The size of the SMS base configuration and certain constructs increases to support the additional system and/or system group names. Previously allocated SMS control data sets might not have sufficient space to accommodate these changes, and an error could result when SMS attempts to write the data sets to DASD.

Use the formulas in Figure 1 to calculate storage requirements for your SMS control data sets. If your control data sets are not large enough, be sure to increase the sizes before installing DFSMS/MVS 1.3, to accommodate the automatic expansion of SMS constructs.

Note: The potential impact of VSAM record-level sharing is also considered in these calculations.

```
SCDS/ACDS size (bytes) = 150000 +
                         (14000 * SG) +
                         (2200  * (MC + SC + DC + AG + CS)) +
                         (12000 * (DRV + LIB + VOL))


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

Variables represent estimates for the following:

  DC  = number of data classes
  MC  = number of management classes
  SC  = number of storage classes
  SG  = number of storage groups
  AG  = number of aggregate groups
  CS  = number of cache sets in the base configuration (for VSAM RLS only)
  VOL = number of SMS-managed DASD volumes
  DRV = number of SMS-managed optical drives
  LIB = number of SMS-managed optical and tape libraries
```

Figure 1. Storage requirements for expanded SMS control data sets

ISMF Changes

The following changes to ISMF provide for the support of up to 32 system and/or system group names in the SMS configuration:

- The Storage Group Application uses a second panel for pool, VIO, tape, object, and object backup storage groups to allow you to define, alter, or display the status of up to 32 names.
- A new STATUS line operator for DASD volumes allows you to display more than 8 sets of SMS and MVS volume statuses.
- The Library Management Application lets you define, alter, or display the status for up to 32 systems or system groups.
- The Control Data Set Application allows you to define, alter, or display up to 32 system names, system group names, or both.
- A new Conversion Confirmation panel is provided for compatibility mode support.

Operator Commands Modified

Output of the operator commands DISPLAY SMS and SET SMS is changed for this support.

DISPLAY SMS is expanded to show information for up to 32 systems or system groups when displaying storage group, volume, library, or drive information. DISPLAY SMS, ACTIVE provides information for up to 32 systems or system groups, and also displays the value of the new IGDSMSxx SYSTEMS keyword.

SET SMS, used to restart the Storage Management Subsystem with SYS1.PARMLIB member IGDSMSxx, can also be used to change the value of the IGDSMSxx SYSTEMS keyword.

Note: The SETSMS command does not support the new SYSTEMS keyword.

Programming Interfaces Affected

When SMS expands constructs to allow space for additional system information, the changes are transparent to the user and require no special action unless the user runs a program that accesses any of the new or changed fields in the SMS configuration. If you have a program that retrieves information from your SMS configuration, check with the vendor who supplies the product to verify that you are using a version that supports these changes.

New fields were added to the mapping macro, IDCDOUT, to allow IDCAMS DCOLLECT to report on the expanded SMS information. User applications processing DCOLLECT output should be changed to access these new fields.

Performance Characteristics

No changes in job throughput should occur when this support is installed. However, because more systems can now share the SMS control data sets, you should use global resource serialization (GRS) to convert hardware reserves on the ACDS and COMMDS to SYSTEMS scope enqueues to avoid contention problems. Reserves are issued with a major name of IGDCDSXS and a minor name of the data set name.

You should place resource name IGDCDSXS in the RESERVE conversion resource name list (RNL) as a generic entry. This will minimize contention

delays and prevent deadlocks which are sometimes associated with the VARY SMS command.

Other Migration Considerations

This section includes examples for migration and fallback plans and considerations for setting up a remote configuration.

- **Preparing a conversion plan:** You can begin converting your SMS complex from compatibility mode to 32-name mode once every system in the complex has DFSMS/MVS 1.3. installed and other prerequisites are met.

  All systems must run in the same mode to share SMS control data sets. Because it is physically impossible to convert all systems at exactly the same moment, there is a brief window of time in which some systems are running in compatibility mode while others are running in 32-name mode. To diminish the impact of this window, plan to perform the conversion during a period in which allocation activity is very low. Otherwise, some systems could start issuing messages indicating the SMS configuration or communications data set is inaccessible due to a conflict between the system and control data set modes.

  When you convert an SMS control data set to support more than eight systems, the change is permanent. Be sure to copy your current SMS control data sets to data sets with different names before you begin the conversion.

  Include the following steps in your conversion plan:

  1. Set a date and time for the conversion when activity on the systems is likely to be low.
  2. Create an SCDS, ACDS, and COMMDS by copying existing SMS control data sets. Be sure to use the new formulas for data set space requirements. To provide for emergency fallback, do not convert your existing primary SCDS, ACDS, and COMMDS at the outset. Once converted, an SMS control data set cannot be changed back to compatibility mode.
  3. On each system in the SMS complex, create an IGDSMSxx member in SYS1.PARMLIB that points to the new ACDS and COMMDS and specifies SYSTEMS(32).
  4. Immediately before conversion, issue SETSMS INTERVAL(999) on all systems involved in the conversion. This prevents a system from rereading the COMMDS and attempting to activate the new configuration before you have an opportunity to change the mode of that system.
  5. On each system, issue SET SMS=xx to restart SMS or re-IPL the system using the new IGDSMSxx member of SYS1.PARMLIB. This restarts SMS in 32-name mode using the new ACDS and COMMDS.

     Note: Restarting SMS resets the INTERVAL parameter for the system. If INTERVAL is specified in the IGDSMSxx member of SYS1.PARMLIB, that value is used, otherwise, the value defaults to 15.

- **Preparing a fallback plan:** Reverting to compatibility mode is possible only if the original SMS control data sets are retained unchanged. While an SCDS or ACDS can be reformatted from compatibility mode to 32-name mode, the reverse cannot be done. Timing is an important issue for the fallback procedure as it is for conversion, therefore, try to perform this emergency procedure during a period in which SMS activity is low.

  Include the following steps in your fallback plan:

  1. Save your previous SCDS, ACDS, and COMMDS to ensure that fallback is possible.
  2. Keep an IGDSMSxx member in SYS1.PARMLIB, for each system, that points to the old ACDS and COMMDS and specifies SYSTEMS(8).
  3. Immediately before converting, issue SETSMS INTERVAL(999) on all systems involved in the conversion.
  4. Issue SET SMS=xx to restart SMS, or re-IPL each system, using the new SYS1.PARMLIB IGDSMSxx member that specifies SYSTEMS(8).

- **Setting up a remote configuration:** If you have a case where a storage administrator on your local system needs to create an SMS configuration for a remote system, note the following:

  – If both systems are pre-DFSMS/MVS 1.3, either system can create a configuration for the other.

  – If the local system is pre-DFSMS/MVS 1.3, it can create a configuration for the remote system running DFSMS/MVS 1.3, provided the remote system is operating in compatibility mode. If the remote system operates in 32-name mode, the configuration must be converted prior to update or activation on the remote system.

  – If the local system has DFSMS/MVS 1.3. installed and runs in compatibility mode, it can create configurations for remote systems running previous releases. SMS control data sets should be allocated using the new formulas for DFSMS/MVS 1.3 to compensate for the changes in SMS constructs.

  – If the local system has DFSMS/MVS 1.3 installed and runs in 32-name mode, it can create configurations for only DFSMS/MVS 1.3 systems running in the same mode.

Where to Find More Information

DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
DFSMS/MVS Using ISMF
DFSMS/MVS Access Method Services for ICF
OS/390 MVS Initialization and Tuning Reference

Storage Management Subsystem Usability Enhancements

Enhancements to the storage management subsystem include:

- **Construct access checking:** A new method for SMS construct access checking allows users to take advantage of storage and management classes even when they do not use a naming convention, user ID, or group ID as the high-level qualifier for their data set names. The user ID of the user executing the job can now be extracted, instead of the resource owner (RESOWNER), when SMS construct access checking is performed.
- **Management class ACS routine redriven for data set rename:** When the name of an SMS-managed data set changes, a new management class policy can be derived automatically. The SMS management class ACS routine is redriven when an SMS-managed data set is renamed, allowing the management class to be reassigned based on the new name and other data set attributes.
- **Data class SPEED and REUSE support:** Data class definitions now allow specifications for SPEED and REUSE, enabling users to define VSAM data sets with these attributes through JCL or dynamic allocation. IDCAMS DEFINE CLUSTER applies these attributes derived from the data class before applying other defaults.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no additional software dependencies for this support.

Hardware Dependencies

There are no hardware dependencies

Coexistence Issues

For rename support, if there are mixed product levels in your SMS complex, whether or not the management class routine is redriven depends on which system the rename operation is performed.

SMS Changes

The ACS routines support a new environment, rename, where the &ACSENVIR read-only variable is set to RENAME. The management class ACS routine must be modified for the rename environment to be become effective.

ISMF Changes

The Data Class Application is enhanced with two new fields, REUSE and INITIAL LOAD, which are added to the Data Class Define, Alter, Display, List, Sort, and View Panels. The REUSE parameter is for specifying the VSAM REUSE or NOREUSE options. The INITIAL LOAD parameter indicates whether VSAM SPEED or RECOVERY options apply.

The Automatic Class Selection Application's ACS Test Case Define/Alter functions are modified to accept the new value, RENAME, for the ACSENVIR variable.

System Interfaces Affected

To support construct access checking, SYS1.PARMLIB member IGDSMSxx has a new optional parameter, USE_OWNER, that determines whether the execution user ID will be extracted instead of RESOWNER. RESOWNER is extracted for construct access checking if the default is taken.

Other Migration Considerations

If you plan to take advantage of the rename support, keep the following in mind:

- Installations that use CONVERTV operations to redrive the ACS routines or exits might want to reevaluate this practice.
- These changes should be communicated to users. It is possible that if the management class is not changed automatically on a rename request or if it is changed and the user isn't aware of the new policy assigned, the data set might not get backed up or migrated as the user expects.

  Users who wish to retain the original management class when renaming a data set can be allowed to invoke IDCAMS ALTER to override the management class assigned by the ACS routines.

Where to Find More Information

DFSMS/MVS DFSMSdfp Storage Administration Reference

Tape Mount Management Enhancements

Tape mount management (TMM) is a methodology for improving tape utilization and for reducing costs associated with tape mounts. This methodology involves analyzing tape usage, building ACS routines to intercept selected tape data set allocations, and redirecting these allocations to a DASD buffer. Once on DASD, these data sets can be migrated to tape by DFSMShsm, thus allowing multiple data sets to be written to a single tape or small set of tapes. All of this can be accomplished without requiring users to change their existing JCL.

Tape mount management enhancements, available in DFSMS/MVS 1.3, resolve restrictions for certain types of tape processing that might otherwise inhibit a successful implementation of tape mount management:

- **Data set stacking:** When two or more data sets are placed on the same tape volume or set of tape volumes, the data sets are said to be *stacked*. A *data set collection* is the collection of data sets that are intended to be allocated on the same tape volume or set of tape volumes as a result of stacking.

  DFSMS/MVS now recognizes most cases where data set stacking is used within a job step, and for all data sets allocated within a data set collection, attempts to ensure consistency of the device category assigned. *Device categories* are SMS-managed DASD, SMS-managed tape, non-SMS-managed DASD, and non-SMS-managed tape.

  A tape or DASD request is considered part of a data set collection when the following conditions are true:

  – A data set sequence number greater than one is specified on the LABEL parameter.

  – VOL=SER is specified, or can be obtained from the catalog. At least one of the volume serial numbers associated with this request must match a volume serial number for a previous request in the same step.

  – VOL=REF is not specified. Implementation for VOL=REF processing already ensures that consistency is maintained.

– The data set is not an existing SMS-managed data set. Existing SMS-managed data sets are not included in any data set collection.

New values are set in SMS ACS read-only variables to help determine whether a stacked data set should go to system-managed DASD or to tape.

- **Unit affinity:** When two or more volumes are assigned the same device, the volumes are said to have *unit affinity* within the same job step allocation. Unit affinity implies deferred mounting for all except one of the volumes.

  The system can now ignore unit affinity when inconsistent device categories are detected in cases where data set stacking is not used. This allows you to redirect some allocations in a unit affinity chain, while not necessarily redirecting all of them.

  As explained before, the system does attempt to ensure consistency among device categories when data set stacking is used.

- **Volume referencing:** For non-SMS-managed data sets, the volume reference (VOL=REF) JCL keyword indicates that data sets are to reside on the same volume. For SMS-managed data sets, VOL=REF implies that the data sets are to have the same storage class.

  When VOL=REF is used, new values are stored in SMS Automatic Class Selection (ACS) read-only variables to help you determine what type of media was used for the referenced data set allocation, so that the referencing data set can be directed to the same type of media.

  When the ACS routines get control for data sets allocated with VOL=REF to a non-SMS-managed data set, the ACS routines can either allow the non-SMS allocation or fail it. It cannot make the referencing data set SMS-managed because this could result in subsequent locate or data integrity problems.

  When VOL=REF is used with data sets going to SMS-managed tape, the referenced and referencing data sets must be allocated in the same storage group. When VOL=REF is used with data sets redirected to SMS-managed DASD or VIO, the referencing data set can be allocated in the same storage group as the referenced data set, or in a different storage group—called a spill or overflow storage group—provided it is of a compatible type, that is, POOL or VIO.

- **Volume reference chains:** Allocation failures have been eliminated in cases where volume reference (VOL=REF) chains have a NEW to OLD reference preceding a NEW to NEW reference. An example of this case follows:

```
//DD1 DD DSN=A,DISP=(OLD,KEEP)
//DD2 DD DSN=B,DISP=(NEW,CATLG),VOL=REF=*.DD1
//DD3 DD DSN=C,DISP=(NEW,CATLG),VOL=REF=*.DD2
```

Refer to the *OS/390 MVS JCL User's Guide*for examples of data set stacking and related discussions.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

MVS/ESA SP 5.2.2 is a prerequisite of this support.

JES3 systems require OS/390 Release 3 for the data set stacking and unit affinity capabilities.

Hardware Dependencies

There are no unique hardware requirements for this support.

Coexistence Issues

An SMS configuration can be shared between DFSMS/MVS 1.3 systems and down-level releases of DFSMS/MVS and MVS/DFP. If you have systems in an SMS complex that are at mixed software levels, you should be aware that JCL execution on the individual systems will vary. A system in this environment might be unable to resolve data set stacking or volume reference problems. In these cases, the job will still fail but it will probably fail earlier than it did with previous support.

If you intend to take advantage of these tape mount management enhancements, we recommend you bring all systems in your SMS complex up to the software levels indicated before you implement this support.

JES3 installations running earlier releases of DFSMS/MVS or MVS/DFP, or prior levels of MVS/ESA SP, must apply compatibility PTFs to successfully compile the new JES3 source code related to these enhancements.

System Interfaces Affected

A new SMS data set stacking subsystem interface (SSI) is introduced for this support. The SMS data set stacking SSI is invoked by allocation and JES3 processing when the system detects a case of data set stacking involving mixed device categories. Control is passed to the ACS routines to resolve the device categories in a data set collection. This could result in the ACS routines being invoked two, or even three, times.

The following SYS1.PARMLIB members are also changed for this support:

- IGDSMSxx has a new subparameter, DSTACK, added to the SELECT and DESELECT parameters. This value is used to trace the execution of the new SMS data set stacking SSI.
- ALLOCxx has a new subparameter, UNITAFF, added to the UNIT parameter. This supplies the system with a default unit name on which to place data sets when it cannot determine a unit name through other means. This value is used only for specific cases where unit affinity is ignored and data set stacking is not involved.

  Note: If a unit name is specified on the UNITAFF subparameter on one system in a JES3 complex, the same unit name must be specified in ALLOCxx for all systems in the JES3 complex. If a unit name is not specified for UNITAFF, then the system-derived name default must be the same on all systems in the JES3 complex.

SMS Changes

You need to update your ACS routines to take advantage of this support. New and modified ACS read-only variables make it easier for your ACS routines to determine what to do with data set stacking and volume referencing requests.

When UNIT=AFF is used, the &UNIT read-only variable is set to AFF= upon initial invocation of the ACS routines. If the system determines that data set stacking is involved and data sets are going to different device categories, the ACS routines are given control again to resolve the data set stacking situation.

With data set stacking, the &UNIT read-only variable is set to one of the following values:

STK=SMSD — Data set collection is on SMS-managed DASD

STK=NSMS — Data set collection is on non-SMS-managed DASD or tape

When VOL=REF is used, the &ALLVOL and &ANYVOL ACS read-only variables contain one of the following values:

REF=NS — Reference is to a data set on non-SMS-managed DASD

REF=ST — Reference is to a data set on SMS-managed tape

REF=SD — Reference is to a data set on SMS-managed DASD

If the reference is to a data set on an SMS-managed volume, the storage group of the referenced data set, if available, will be provided in the &STORGRP read-write variable. If the reference is to a new data set not yet allocated, &STORGRP will contain the first candidate storage group of the referenced data set.

For DASD or VIO data set references, the storage group ACS routine can allow allocation in the same storage group as the referenced data set or can select another storage group, provided it is compatible with the storage group of the referenced data set. If the referenced data set is in a tape storage group, the allocation must go to that same storage group.

ISMF Changes

The Automatic Class Selection Application's test case define/alter functions are modified to accept a storage group value for the &STORGRP read-write variable, a value that SMS passes when volume referencing is used.

Operator Commands Modified

The following operator commands are enhanced for this support:

- The DISPLAY SMS,OPTIONS and DISPLAY SMS,TRACE commands are used to display the value of the new tracing subparameter DSTACK.
- The SET SMS command can pick up the value of the new tracing subparameter DSTACK from the IGDSMSxx member of SYS1.PARMLIB.
- The SETSMS command provides for the specification of the new tracing subparameter DSTACK on the SELECT and DESELECT parameters.

Performance Characteristics

Processing differs if you have a DISP=NEW, or DISP=MOD treated as NEW, request involving a data set collection where the request is directed to a different device category than the other requests in the data set collection. Allocation invokes the new SMS data set stacking SSI once for each pair of DD statements in the data set collection, which in turn redrives the ACS routines to resolve the inconsistent device categories.

Allocation invokes the SMS data set stacking SSI during execution processing. In a JES3 environment, JES3 also invokes the SSI during pre-execution processing. In cases where the SMS data set stacking SSI is called, the ACS routines will be driven a total of two times in a JES2 environment and three times in a JES3 environment. This can cause a negative impact in performance; however, note that before this support was avail-

able, the same situation caused by inconsistent device categories resulted in a job failure.

This support can also affect overall performance in a positive way. By making more tape data sets eligible for tape mount management, you effectively improve job throughput—because the jobs are not always queued up on tape drives.

Processing Restrictions

You should be aware of the following restrictions when implementing TMM with this support:

- Specific volume allocations for data sets in an SMS-managed tape library cannot be redirected to any other media. This also applies to allocations that use volume referencing to data sets managed in a tape library. These allocations are always directed to SMS-managed tape. This restriction is unchanged from previous releases.
- Specific volume allocations which specify volumes that are not SMS-managed tape cannot be redirected to SMS-managed tape. This restriction is unchanged from previous releases.
- The system cannot detect data set stacking across job or step boundaries when VOL=SER is used and, therefore, cannot ensure the consistency of device categories in these cases.

Other Migration Considerations

You should be aware of the following operational changes and other processing differences associated with this support:

- **Error detection with data set stacking:** When specifying data set stacking with the VOL=SER option, be aware that the system now detects errors during allocation that were previously not discovered until OPEN time. A failure results when the system determines that not all of the data sets in the data set collection are directed to the same device category and the system cannot resolve the conflict.
- **Unit Affinity for TMM and non-TMM Users:** Both TMM and non-TMM users might notice a difference in the unit selected for a referencing request in a unit affinity chain when unit affinity is ignored. This occurs when the primary request is a cataloged SMS-managed tape data set and the referencing request is a non-SMS-managed DASD or tape data set.

  The system previously derived the unit for the referencing request from the unit information found in the catalog for the primary data set. For a cataloged SMS-managed tape data set request in which a unit value is not specified in the JCL, the system now uses the default unit-affinity-ignored unit name found in UNITAFF of ALLOCxx. The request is handled this way because the catalog for the primary data set might contain unit information that resulted from TMM redirection, not what was originally specified in the JCL. In the case where a unit value is specified in the JCL, the system takes the value from the unit parameter of the referencing request.
- **JES3 considerations:** The system on which JES3 invokes the new SMS data set stacking SSI might differ from the system on which allocation invokes the SSI. As long as the JES3 systems are part of the same SMS complex, JES3 and allocation should be able to reach a

consistent decision about the number of devices needed to satisfy a request, independent of which system the call is made on.

There are unique JES3 considerations when implementing spill storage groups for VOL=REF processing, especially in the case where the storage group selected for the referencing data set differs from the one selected for the referenced data set. JES3 might not take into account the connectivity of the storage group selected for the referencing data set when selecting a processor on which to set up the job. If the storage group selected for the referencing data set is in an unusable state, such as disabled, this could result in a job failure. Spill storage groups should remain in a usable status to provide for overflow processing.

Where to Find More Information

DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Implementing System-Managed Storage
DFSMS/MVS Using the Volume Mount Analyzer
OS/390 Planning for Installation
OS/390 MVS Conversion Notebook
OS/390 MVS Initialization and Tuning Reference
OS/390 MVS JCL User's Guide

Unit Control Block Virtual Storage Constraint Relief

Unit control block (UCB) virtual storage constraint relief enables you to define a DASD, tape or optical device to MVS so that the UCB representing that device resides in 31-bit storage above 16MB. This provides a way for you to attach more devices to your MVS system, as many as 65,536 device numbers.

UCB virtual storage constraint relief can improve systems management in a coupled systems environment by facilitating the definition of a single I/O configuration across the parallel sysplex. This eliminates the need to tailor I/O configurations to the specific data access requirements of each system in the parallel sysplex.

You can migrate existing UCBs above the 16MB line on a device-by-device basis. This support maintains compatibility with 24-bit UCB addresses through the use of captured UCBs. A *captured UCB* is created at allocation and represents a virtual window into the *actual UCB* used for I/O operations. Because the captured UCB can reside in private storage below 16MB, it enables you to realize the benefits of UCB virtual storage constraint relief without changing most of your existing applications.

Functional Components Enhanced
DFSMSdfp, DFSMSdss, DFSMShsm, DFSMSrmm

Software Dependencies
To move a UCB above the 16MB line, you must have MVS/ESA SP 5.2, along with Hardware Configuration Definition (HCD) support at the same 5.2 level.

Hardware Dependencies
UCB virtual storage constraint relief requires a processor supported by MVS/ESA SP 5.2.

Coexistence Issues

This support is available on MVS/ESA SP 5.2 systems. A device defined in the HCD can have a UCB above 16MB only if the system has, at a minimum, DFSMS/MVS 1.3 and MVS/ESA SP 5.2 installed.

Performance Characteristics

A small increase in path length can occur when a UCB is captured.

Programming Interfaces Affected

Some applications could require source code changes if their programs handle actual UCB addresses, or if they call a DFSMS/MVS service that passes an input or output UCB address. Because software owners might not be aware of this situation, DFSMS/MVS interfaces provide diagnostic information, where possible, to help make this determination.

The following program interfaces are modified for this support:

- **Checkpoint/Restart:** Programs requesting checkpoints after allocating to a device that has an uncaptured UCB receive a return code 8—checkpoint not taken—even after all such allocations are released.
- **DADSM/CVAF:** Macro syntax and expansions have not changed. However, the system code invoked by the macros was changed to require the caller in 31-bit addressing mode to pass only 31-bit UCB addresses. An address passed can be an actual 31-bit address or a captured UCB. In the latter case, the high-order byte must be zero. In previous releases the system always cleared the high-order byte.
- **DEVTYPE Macro:** The DEVTYPE macro obtains information about the characteristics of an I/O device and places this information into an area specified by the caller. DEVTYPE previously cleared the high-order byte of a 4-byte UCB address passed by the caller, to ensure a pure 24-bit address. DEVTYPE now determines the addressing mode of the invoking program, before it accepts the UCB address.

  If the invoking program is in 31-bit addressing mode and ANY is coded as the third value of the UCBLIST keyword, the program is passing 31-bit UCB addresses. The address can point to an actual UCB above or below the 16MB line or to a captured UCB. If ANY is not coded, or the invoking program is in 24-bit mode, the program can pass only a 24-bit real or captured UCB address.
- **IDCAMS:** If a DD statement is allocated with the NOCAPTURE option and the UCB resides above 16MB, IDCAMS will fail. Call interface users are not affected as long as they ensure that the UCB is captured. JCL users are not affected; normally the associated JCL DD statements apply to captured UCBs.
- **IEBCOMPR, IEBCOPY, IEBDG, IEBGENER, and IEBPTPCH:** Call interface users are not affected unless they pass DD statements that have been allocated with the NOCAPTURE option, in which case, the utility fails when attempting to open the DD statement. JCL users are not affected.
- **IEHINITT:** A program can use the nocapture option of SVC 99 to allocate a tape drive and pass the device to IEHINITT. JCL users are not affected.
- **IEHLIST, IEHMOVE, IEHPROGM:** Call interface users are not affected unless the UCBs are allocated above 16MB with the NOCAPTURE or

XTIOT options. In this case, the program fails during execution. JCL users are not affected.

- **IGWSPZAP:** IGWSPZAP continues to work if invoked through JCL where, by default, the UCBs are captured. If IGWSPZAP is invoked through the callable program interface, the caller must ensure that the UCB is captured; if the UCB is uncaptured and resides above 16MB, IGWSPZAP will fail.
- **Open/Close/End-of-Volume Processing:** Open processing could result in a return code of 8, indicating that the caller passed an extended TIOT (XTIOT) entry, possibly tied to the specification of NOCAPTURE for a dynamic allocation. Open processing supports the NOCAPTURE option if VSAM is the access method used.
- **Program management:** The AMBLIST, linkage editor, and binder programs continue to work if invoked through JCL, where the UCBs are captured. If a program is invoked through the callable program interface, the caller must ensure that the UCB is captured. If the UCB is uncaptured and resides above 16MB, the function will fail.
- **DFSMSrmm:** The interface to installation exit EDGUX100 always passes an actual UCB address. The exit will fail if it passes the address to a service that does not support a UCB above the 16MB line.
- **Tape Processing:** The tape device dependent segment and the tape device dependent extension are now mapped as contiguous storage by IECDUCBT. The fields are addressable through the UCB common segment. For UCBs that reside above the 16MB line, the tape device class extension also resides above the 16MB line. The 4-byte pointer from the common extension to the device class extension has not changed. The 3-byte pointer from the tape device dependent segment to the tape device dependent extension is zero for UCBs that reside above the line. For compatibility, this pointer is still valid for UCBs residing below the line.

  MSGDISP accepts 24- or 31-bit actual or captured UCBs. However, if TEST=YES is specified, the AMODE of the caller must be compatible with the UCB address.

  SVC 91 accepts actual or captured UCB addresses. The high-order byte of the UCB address is cleared if the caller is in 24-bit addressing mode. If the caller is in 31-bit addressing mode and passes a 24-bit actual or captured UCB address, it must ensure that the high-order byte of the 4-byte field is zero.
- **TRKCALC Macro:** TRKCALC, which runs in the AMODE of the caller, receives a 4-byte UCB address at the time of the branch entry. Previously, TRKCALC cleared the high-order byte of the UCB address to ensure a pure 24-bit address. If the invoking program is in 31-bit addressing mode and LOC=ANY is coded, the program is passing a 31-bit UCB address. The address can point to an actual UCB above or below the 16MB line or to a captured UCB. If LOC=ANY is not coded, or the invoking program is executing in 24-bit mode, the program can pass only a 24-bit real or captured UCB address.

- **VSAM:** VSAM runs in 31-bit addressing mode and uses 31-bit UCB addresses. Whenever possible, it passes the actual UCBs to macros and calling programs.

Processing Restrictions

VSAM catalogs and CVOLs are not eligible for UCB virtual storage constraint relief; they must reside on volumes that have actual UCBs below the 16MB line. Password data sets, on system residence volumes, and password-protected data sets also must reside on volumes that have UCBs below the 16MB line.

JCL JOBCAT and STEPCAT allocations are not allowed for this support. A JOBCAT or STEPCAT DD statement must refer to a catalog on a volume whose actual UCB is defined below 16MB.

Other Migration Considerations

You have the option to define devices either above or below 16MB by specifying the LOCANY parameter on the HCD panel.

UCB virtual storage constraint relief is limited, by the configuration definition process, to newer devices only. If you are running MVS/ESA SP 5.2 and DFSMS/MVS at 1.3, HCD will not generate UCBs for the following devices types: 3330, 3333, 3340, 3344, 3350, 3375, 3350P, 3351P, 2305-2. If you have data on these older devices, it must be migrated to an IBM 3380, 3390, or 9345 device geometry before you migrate the system to MVS/ESA SP 5.2 and DFSMS/MVS 1.3.

Where to Find More Information

DFSMS/MVS Installation Exits
DFSMS/MVS Checkpoint/Restart
DFSMS/MVS Using Data Sets
DFSMS/MVS DFSMSdfp Advanced Services
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Utilities
DFSMS/MVS Program Management
DFSMS/MVS DFSMSrmm Implementation and Customization Guide
DFSMS/MVS DFSMShsm Storage Administration Guide
OS/390 MVS Conversion Notebook

VSAM Extended Addressability

VSAM extended format key-sequenced data sets (KSDS) can now grow larger than 4GB in size. This extends the addressability, or relative byte address (RBA), of the data, index and alternate index components of a KSDS beyond 32 bits. Only SMS-managed data sets are eligible for this support.

The new size limit for a VSAM extended format KSDS is determined by the control interval (CI) size times 4GB, or the volume size times 59. A 4K CI size yields a maximum data set size of 16TB, while a 32KB CI size yields a maximum data set size of 128TB. A 4K CI size is preferred by many applications for performance reasons.

Applications that access a VSAM extended format KSDS through a user-specified key can take advantage of this support without making JCL or code changes.

Applications that process an extended format KSDS by RBA must provide an 8-byte RBA when the data set is defined for extended addressability and the type of access uses the RBA specified. Special provisions are allowed for certain types of requests (for example GET SEQ,ADR or GET ADR,DIR,LRD) to use either a 4-byte or 8-byte RBA because, although ADR or CNV is specified in the RPL, the RBA argument passed is not actually referenced.

Note: With DFSMS/MVS 1.5, support for extended addressability is expanded to all other VSAM record organizations.

Functional Components Enhanced
DFSMSdfp, DFSMSdss, DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
An extended format KSDS must be defined on DASD attached to a cached storage control; for example, an IBM 3990 Model 3 with the Extended Platform, an IBM 3990 Model 6, or an IBM 9340 DASD array subsystem.

Coexistence Issues
A VSAM extended format KSDS defined for extended addressability must not be shared with any system running a release prior to DFSMS/MVS 1.3. Toleration maintenance is available for DFSMS/MVS 1.2 systems which support VSAM extended format data sets without extended addressability. A DFSMSdfp toleration PTF allows systems at this level to issue an error message if any attempt is made to open a VSAM KSDS with extended addressability. DFSMSdss provides a toleration PTF so that an error message is issued when a logical DUMP, RESTORE, or COPY operation is attempted on a KSDS with extended addressability.

A VSAM extended format KSDS not defined for extended addressability, is compatible with all releases (that is, DFSMS/MVS 1.2 and above) that recognize the extended KSDS format.

A VSAM extended format KSDS cannot be accessed on any release prior to DFSMS/MVS 1.2. If an extended format data set is opened in load (create) mode on a system that does not recognize the format type, the data will be written in a manner that makes it unusable on a system that does support extended format data. Toleration PTFs are available for releases prior to DFSMS/MVS 1.2 to prevent a system from opening an extended format KSDS.

SMS Changes
Only SMS-managed extended format key-sequenced data sets are eligible for extended addressability. To provide for applications that want to use this support, you need to set up at least one SMS data class with the necessary data set name type (DSNTYPE) and EXTENDED ADDRESSABILITY parameters. You can specify EXTENDED ADDRESSABILITY when DSNTYPE = EXT (extended format), provided that RECORG = KS (KSDS). There is no default for EXTENDED ADDRESSABILITY; a value of Y or N must be indicated. A value of N for EXTENDED ADDRESSABILITY restricts the data set from having an RBA greater than 4GB.

You also need to update your data class ACS routine to reflect this new extended format type data class, and to assign this class to eligible VSAM data sets, when appropriate.

ISMF Changes

The ISMF Data Class Application contains the new field, EXTENDED ADDRESSABILITY, on the Data Class Define, Alter, Display List, Sort, and View panels.

Programming Interfaces Affected

VSAM extended addressability affects VSAM record management and catalog processing. The following programming interfaces are modified as a result of this support:

- The VSAM RPL has two new option codes, XRBA and RBA, that allow you to specify whether or not extended addressability should be used for a request. When XRBA is indicated, the RBA argument field (pointed to by RPLARG) is 8 bytes, or two full words, in length. When two RBAs are requested, the argument field is 16 bytes, or four full words, in length.

  Shared resource processing normally provides you with RBA information in a parameter list, or in the BUFC data area, when the JRNAD exit is used. If the XRBA option code is specified in the RPL, a CI number is provided instead of the RBA.

  The following macros are also affected by these changes: POINT, GET, MRKBFR, SCHBFR, WRTBFR, GENCB, MODCB, SHOWCB, and TESTCB.

- The JRNAD exit provides a 4-byte RBA in the BUFC data area for the input or output location of a CI. The RBA is replaced with the CI number when the RPL indicates processing with extended addressability. The JRNAD exit for data movement is not taken for data sets that can grow beyond the 4GB boundary.

- IDCAMS LISTCAT displays a new EXT_ADDR attribute and modified statistics for data sets with extended addressability.

  The following LISTCAT output field names are changed to accommodate the larger RBA values displayed in LISTCAT output:

  - FREESPC-BYTES is now FREESPC
  - HI-ALLOC-RBA is now HI-A-RBA
  - HI-USED-RBA is now HI-U-RBA

  Applications that interrogate LISTCAT output might need to be updated to recognize these new field names.

- IDCAMS REPRO and PRINT can accommodate the larger RBA values.

- IDCAMS EXPORT/IMPORT offers data set portability for systems with extended addressability support.

- IDCAMS DCOLLECT output normally contains high-used RBA and high-allocated RBA values. These fields contain the high-used CI and the high-allocated CI for an extended format KSDS allocated on a DFSMS/MVS 1.3 system.

Performance Characteristics

No increase in processing time is expected for extended format key-sequenced data sets that grow beyond 4GB.

Processing Restrictions

Extended format key-sequenced data sets cannot be defined with key ranges (KEYRANGE parameter) or with an imbedded sequence set (IMBED parameter). Temporary data sets cannot be allocated in extended format. Other data sets that are not eligible for extended addressability include catalogs, SMF data sets, SYS1.STGINDEX (used for the pageable link pack area), SYS1.DDIR dump directory data sets, or any other VSAM data sets that require special processing at a system level.

Hiperbatch processing is not available for extended format data sets. If an extended format KSDS is larger than 4GB, it cannot be accessed for VSAM record-level sharing (RLS).

Other Migration Considerations

When you elect to use VSAM extended addressability, you need to plan for both data and application migration:

- **Data migration:** There are four possible data migration scenarios to consider with this level of support:
  - Migrating an extended format KSDS to an extended format KSDS with extended addressability
  - Migrating a standard format KSDS to an extended format KSDS with extended addressability
  - Migrating an extended format KSDS that has extended addressability to an extended format KSDS without extended addressability.
  - Migrating an extended format KSDS that has extended addressability to a standard format KSDS

  In each of these situations, you should use IDCAMS REPRO to perform the migration. When copying data from an extended format KSDS that has extended addressability to either an extended format KSDS, or a standard format KSDS, you must first determine if the data occupies space beyond 4GB and, if it does, copy a subset of the data so that it fits into a smaller data set. Keep in mind that compression is also available for extended format data sets.

  To provide a means for users to create extended addressable key-sequenced data sets, you need to define one or more data classes, as discussed under “SMS Changes” in this topic. You can give users the option to code the DATACLAS value on their DD statements or let the ACS routines assign the appropriate class for their eligible data. The only other method in which JCL can be used to create a KSDS with extended addressability, is through the DD statement keyword LIKE.

- **Application migration:** Programs that will use a KSDS with extended addressability must be evaluated for their addressing methods. A program that accesses a KSDS by RBA needs to be changed to use an 8-byte RBA value, or to access the data through an index search key instead. A program that uses the indexing method does not require any changes.

  IMS supports extended format key-sequenced data sets only when extended addressability is not permitted.

CICS, which provides an interface that allows users to access data by RBA, is not restricted from using VSAM extended addressability. Contact your IBM representative for more information on the product level that exploits this support.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS Macro Instructions for Data Sets
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS DFSMSdss Storage Administration Reference

VSAM Record-Level Sharing

VSAM record-level sharing (RLS) extends the DFSMS/MVS storage hierarchy to support data sharing across multiple systems in a System/390 parallel sysplex. This support offers CICS OnLine Transaction Processing (OLTP) the performance and availability benefits of data sharing in a coupled-systems environment.

As a new data access mode, VSAM RLS allows multisystem access to a VSAM data set while ensuring cross-system locking and buffer invalidation. VSAM RLS uses MVS coupling facility (CF) services to perform data set level locking, record locking, and data caching. VSAM RLS maintains data coherency at the control interval level. It uses CF caches as store-through caches; when a control interval of data is written, it is written to both the CF cache and to DASD. This ensures that a failure in the CF cache does not result in the loss of VSAM data.

The SMSVSAM server is a new system address space used for VSAM RLS. The data space associated with the server contains most of the VSAM control blocks and the system-wide buffer pool used for data sets opened for record-level sharing. SMSVSAM assumes responsibility for synchronizing this control block structure across the parallel sysplex.

With VSAM RLS, multiple CICS systems can directly access a shared VSAM data set, eliminating the need for function shipping between application owning regions (AORs) and file owning regions (FORs). CICS can provide logging, commit, and rollback functions for VSAM recoverable files; VSAM provides for serialization of data sets and cross-system caching.

A new attribute, LOG, defines a data set as recoverable or non-recoverable. Because CICS maintains the log of changed records for a data set (thus allowing transactional changes to be undone), only VSAM data sets under CICS are recoverable. Whether a data set is recoverable or not determines the level of sharing allowed between applications:

- Both CICS and non-CICS jobs can have concurrent read/write access to non-recoverable data sets;
- Non-CICS jobs can have read-only access to recoverable data sets, concurrent with read/write access by CICS transactions. Full read integrity is ensured.

To enable the coupling facility for VSAM RLS, you must define one or more CF cache structures to MVS and add these to your SMS base configuration. Cache set names are used to group CF cache structures in the SMS base configuration. In

setting up for VSAM RLS processing, you also need to define the CF lock structure to MVS.

A data set is assigned to a CF cache structure based on the SMS policies you set up. If the storage class for a VSAM data set contains a non-blank cache set name, the data set is eligible for record-level sharing. When the data set is opened for RLS-processing, the cache set name is used to derive an eligible CF cache structure to use for data set access.

To fully implement VSAM record-level sharing, you need to consider all of the elements outlined in the planning checklist in Figure 2.

- Determine hardware, software, and availability requirements for the coupling facilities.
- Evaluate applications to determine which can take advantage of VSAM record-level sharing for their data.
- Map out the coupling facility (CF) cache structures; include number, sizes, and connectivity,
- Determine the size of the coupling facility lock structure.
- Plan the changes to MVS to define the coupling facility cache and lock structures.
- Plan the changes to the SMS configuration; include base configuration definitions for CF cache structures, new storage classes, storage group connectivity, and modifications to the ACS routines.
- Determine the size, number, and placement of sharing control data sets (SHCDS).
- Analyze the potential use of reserves on volumes where sharing control data sets will reside. Consider a conversion to enqueues.
- Plan new procedures to ensure data integrity with the coupling facility cache structure added to the storage hierarchy.
- Set up new operational procedures for coupling facility management.
- Add new parameters to SYS1.PARMLIB member IGDSMSxx.
- Establish new authorization levels required for VSAM record-level sharing access and support.
- Determine which VSAM data sets are eligible for record-level sharing. Use IDCAMS DEFINE or ALTER to specify the LOG attribute for each eligible data set, declaring whether the data set is recoverable or non-recoverable.
- Establish application procedures for VSAM record-level sharing. Use IDCAMS DEFINE or ALTER to specify LOG and LOGSTREAMID attributes for VSAM cluster definitions. Define procedures for batch jobs that update recoverable spheres.

Figure 2. Planning checklist for VSAM record-level sharing.

Functional Components Enhanced

DFSMSdfp, DFSMSdss, DFSMShsm

Software Dependencies

VSAM RLS support has a prerequisite of MVS/ESA SP 5.2. It requires global resource serialization (GRS) or an equivalent function for multi-system serialization.

There are additional dependencies on the following program products, or equivalent products, for full-function support:

CICS Transaction Server for OS/390
CICSVR Version 2, Release 3
RACF Version 2, Release 1
Appropriate levels of COBOL, PL/I, FORTRAN, and Language Environment run time libraries, for batch applications that will use VSAM RLS data access.

Hardware Dependencies

To exploit this support, you must have at least one coupling facility connected to all systems capable of VSAM record-level sharing. For multiple coupling facilities, you need to select one facility with global connectivity to contain the master lock structure. You can attach additional coupling facilities to a subset of the systems, but it is the responsibility of system administrators and users to ensure their jobs run on a processor with the required connectivity.

For maximum availability, we recommend you set up at least two coupling facilities with global connectivity. If a coupling facility is not operational, this allows the storage management locking services to repopulate its in-storage duplexed copy of locks to the secondary facility.

Your coupling facility must be at CF control level 2. It must be large enough to contain either a lock structure or a cache structure (or both), and have enough surplus space to allow the structures to be modified.

You define a single, nonvolatile CF lock structure, IGWLOCK00, to enforce VSAM RLS protocols and to perform record-level locking. The CF lock structure must have global connectivity to all systems capable of VSAM record-level sharing. A nonvolatile coupling facility for the lock structure is not required, but it is recommended for high availability environments. If you maintain a volatile CF lock structure, when a power outage causes a parallel sysplex failure and loss of information in the CF lock structure, all outstanding recovery (CICS restart and backout) must be completed before new sharing work is allowed.

To enable the coupling facility for VSAM record-level sharing, you must define the CF cache structures to SMS. You use cache set names to group CF cache structures in the SMS base configuration. CF cache structures associated with a given storage class must have, at a minimum, the same device connectivity as the storage groups mapped to that storage class.

In summary, we recommend you set up multiple coupling facilities for maximum availability and workload balancing, and ensure that these coupling facilities have global connectivity to all systems that will use VSAM RLS. The *DFSMS/MVS DFSMSdfp Storage Administration Reference* provides more information on defining the lock and cache structures and includes recommendations on sizing these structures.

Coexistence Issues

Within a parallel sysplex, an SMS configuration can be shared between systems running DFSMS/MVS 1.3, DFSMS/MVS 1.2, DFSMS/MVS 1.1 or

MVS/DFP 3.3. Toleration PTFs must be applied to all pre-DFSMS/MVS 1.3 systems so that they do not conflict with RLS access. If the SMS configuration is converted to SMS 32-name mode under DFSMS/MVS 1.3, however, no toleration support is provided.

Cache sets and other RLS-related SMS functions can only be defined through ISMF on a DFSMS/MVS 1.3 system. An SMS source control data set (SCDS) altered by ISMF on a DFSMS/MVS 1.3 system can be activated from any of the systems that share the configuration.

A catalog can still be shared between DFSMS/MVS 1.3 and prior releases. LISTCAT output from DFSMS/MVS 1.3 systems, or down-level systems with toleration maintenance, will indicate whether RLS information is present in the catalog for SMS-managed data sets. The existence of RLS information in the catalog does not imply that VSAM RLS is actually in use; however, if the catalog does contain a VSAM sphere that is RLS-capable, the following conditions must be met:

- Data access to the VSAM sphere must be performed from a DFSMS/MVS 1.3 system. Toleration maintenance enables the down level system to fail an OPEN request on a data set that has the LOG attribute.
- IDCAMS operations on the VSAM sphere must be performed from a DFSMS/MVS 1.3 system. Most data set-level operations will fail on down-level systems. A DELETE operation on a down-level system also fails if a VSAM sphere has RLS recovery pending. In the case of EXPORT/IMPORT operations, RLS information in the catalog of the exported copy is lost when the data set is imported on the down-level system.
- DFSMSdss and DFSMShsm functions must be performed by the DFSMS/MVS 1.3 level of these functional components. This applies to both data set and volume-level operations.

JES3 installations must be careful to define cache set names only in the SMS storage classes that are used by data sets opened for RLS processing. A non-blank cache set name causes a job to be scheduled on an RLS-capable system. If all storage classes have non-blank cache set names, then all jobs accessing SMS-managed data sets will be scheduled to DFSMS/MVS 1.3 systems, causing an imbalance in workload between the DFSMS/MVS 1.3 systems and down-level systems.

Product Fallback Considerations

In terms of fallback from DFSMS/MVS 1.3, an SMS SCDS defined on a DFSMS/MVS 1.3 system can also be used by a lower-level system provided the configuration is in compatibility mode (8-system mode) and all toleration PTFs are applied to the down-level system.

When a DFSMS/MVS 1.3 system is in 32-name mode, fallback can only be achieved by reverting to an SMS SCDS that was saved before the system was converted to 32-name support (as specified in SYS1.PARMLIB member IGDSMSxx). Additional fallback planning for this scenario is discussed under “Storage Management Subsystem 32 Name Support” on page 141.

New parameters added to SYS1.PARMLIB member IGDSMSxx for RLS processing also must be removed as part of the fallback procedure.

When falling back from DFSMS/MVS 1.3 to a prior release, RLS information must be deleted from the catalog. The SHCDS CRFESET command can be used on a DFSMS/MVS 1.3 system to delete RLS information from the catalog.

When returning to a previous level of DFSMS/MVS, CICS installations must revert to using CICS File Control Table (FCT) definitions even if the level of CICS is not changing. For additional CICS and CICSVR fallback considerations, refer to the CICS *CICS Recovery and Restart Guide*.

Related fallback information is discussed under "Fallback from VSAM RLS Processing" on page on page 175, later in this topic, and further detailed in the *DFSMS/MVS DFSMSdfp Storage Administration Reference*.

Performance Characteristics

DFSMS/MVS exploits the services of the MVS coupling facility to provide shared VSAM data access across a System/390 parallel sysplex; thus allowing an application to execute in parallel on multiple systems in a parallel sysplex. This can provide a performance advantage over other system structures.

SMS Changes

To implement VSAM RLS, you need to modify your SMS configuration in the following ways:

- Add the CF cache structures to your SMS base configuration.

  You can have up to 256 CF cache set definitions in your base configuration. Each cache set can have one to eight cache structures defined to it, allowing data sets to be assigned to different cache structures in an effort to balance the workload. CF cache structures associated with a given storage class must have, at a minimum, the same connectivity as the storage groups mapped to that storage class.

  Having multiple CF cache structures defined in a cache set could provide improved availability. If a CF cache structure fails and a rebuild of the structure is not successful, SMS dynamically switches all data sets using the failed CF structure to other CF structures within the same cache set.

- Update storage class definitions to associate storage classes with CF cache set names. Direct and sequential CF weight values are ignored and the defaults are used.

- Change the ACS routines to recognize data sets that are eligible for VSAM record-level sharing so that they can be assigned to storage classes that map to CF cache structures.

Storage requirements for SMS control data sets have also changed to accommodate new VSAM RLS information and expanded system information available with SMS 32-name support. See Figure 1 on page 144 for the new formulas for allocating SMS control data sets.

ISMF Changes

The following changes to ISMF provide for VSAM RLS support:

- The Control Data Set Application provides a means for you to define the CF cache set names and associated cache structures in the base configuration.

- The Storage Class Application allows you to specify a cache set name on a storage class definition. CF Cache Set Name is a new storage class attribute.
- The Volume Application uses a new volume status indicator for volumes containing SMS-managed VSAM data sets to indicate whether the data on the volume is CF enabled or quiesced. CF Volume Status is a new DASD volume attribute.
- The Data Set Application includes new status indicators for data sets that use VSAM record-level sharing. CF Status Indicator, CF Monitor Status, CF Cache Structure Name, and CF Cache Set Name are new data set attributes.

Operator Commands Modified

New operational procedures need to be established to manage the coupling facility. These include:

- Use of the MVS DISPLAY XCF command to display the coupling facility resource management (CFRM) policy, the names and sizes of CF facilities, structures within a CF facility, and connectivity status.
- Use of the MVS SETXCF START,REBUILD command to rebuild the CF cache structures or CF lock structure.
- Use of the VARY SMS and DISPLAY SMS commands to manage the CF cache structures, CF lock structure, and sharing control data sets, and for the statistical monitoring of data sets.

System Interfaces Affected

The following types of system interactions used by the system programmer, storage administrator, and database administrator are changed for this support:

- ISMF is used to define CF cache structures to the Storage Management Subsystem. Access the Control Data Set Application to list cache set names in the base configuration. Use the Storage Class Application to specify cache set names for storage class constructs.
- New sharing control data sets are allocated through JCL, IDCAMS, or TSO. Use the VARY SMS,SHCDS and DISPLAY SMS,SHCDS commands to manage these sharing control data sets.
- New parameters are added to the IGDSMSxx member of SYS1.PARMLIB:
  - RLSINIT is used to request that the SMSVSAM server be started during system initialization. The default is to not start it.
  - DEADLOCK_DETECTION is used to set the deadlock detection interval.
  - RLS_MAX_POOL_SIZE is used to set the maximum size, in megabytes, of the SMSVSAM local buffer pool.
  - If you record SMF type 42 records, subtypes 1, 2, 15, 16, 17, 18, or 19 to gather coupling facility information, you can specify SMF_TIME to synchronize SMF type 42 data with SMF and RMF data intervals.
  - The CF_TIME parameter is used to set the recording interval for subtypes 15, 16, 17, 18, and 19.

- VARY SMS,CFCACHE and VARY SMS,CFVOL commands, respectively, are used to control processing for coupling facility cache structures and volumes. VARY SMS,SMSVSAM controls processing for VSAM record-level sharing.
- VARY SMS,FALLBACK and SHCDS CFRESET commands are used to fall back from RLS processing.

The following diagnosis, monitoring, tuning, and installation interfaces are also affected by this enhancement:

- Cache and lock structure statistics are recorded by selecting SMF type 42 records, subtypes 15, 16, 17, 18 and 19.
- DISPLAY SMS is the command used to obtain information about VSAM RLS resources:

  DISPLAY SMS,CFLS - CF lock structure status
  DISPLAY SMS,CFCACHE - CF cache status
  DISPLAY SMS,CFVOL - volume status
  DISPLAY SMS,SHCDS - sharing control data sets
  DISPLAY SMS,MONDS - data sets monitored for statistics

- Statistical monitoring is controlled with the VARY SMS,MONDS command.
- SMSVSAM components are traced using the command TRACE CT,ON,COMP=SYSSMS.

Programming Interfaces Affected

A summary of new and modified program interfaces is included here to help you plan for application changes.

- LOG is a new parameter added to DEFINE CLUSTER and ALTER to specify that a VSAM sphere (base cluster and alternate indexes) can be opened for RLS access. LOG also indicates the recoverable or non-recoverable attributes of the sphere. LOGSTREAMID is a new parameter added to DEFINE CLUSTER and ALTER for recoverable VSAM spheres.
- ACB MACRF=RLS and JCL keyword RLS are methods for accessing VSAM data for record-level sharing.
- IDALKADD (lock-for-add) is a new VSAM request type for record-level sharing. New read integrity options are also added for GET NUP, POINT, and the new JCL keyword RLS(NRI, CR).
- Only user control blocks ACB, RPL, and EXLST are addressable for record-level sharing. These control blocks contain new and modified fields.
- RLSWAIT is a new exit, replacing the UPAD exit for record-level sharing.
- IGWARLS is a new callable service used to query the catalog for information pertaining to record-level sharing.
- SHCDS is a new IDCAMS command for recovery operations.
- ENF code 45 (in decimal) signals the availability of the SMSVSAM Server. Parameter list IDAENF45 is passed to the listener's ENF exit.

Processing Restrictions

Use of VSAM record-level sharing introduces the following processing restrictions:

- VSAM RLS access cannot be used for VSAM linear data sets, key-range data sets, temporary data sets, VSAM volume data sets (VVDS), or catalogs. It also cannot be specified when using the ISAM compatibility interface.

  Note: With DFSMS/MVS 1.4 installed, RLS processing is available for use with extended format key-sequenced data sets larger than 4GB. With prior releases, this is not supported.

- A VSAM KSDS defined for record-level sharing cannot have the index set imbedded (with IMBED option). REPLICATE is allowed.
- VSAM RLS does not allow addressed access (RPL OPTCD=ADR) to a VSAM KSDS, or control interval access (RPL OPTCD=CNV) to any data set organization.
- Individual components of a VSAM cluster cannot be opened for record-level sharing. A direct open of an alternate index is not supported.
- Data sets opened specifically for system use cannot take advantage of VSAM RLS.
- A checkpoint on a VSAM data set cannot be taken while a data set is open for RLS access in the address space.
- Opening a data set for record-level sharing does not establish implicit positioning. An explicit POINT or GET NSP is required.
- The GETIX and PUTIX macros are not supported.
- VSAM RLS does not support Hiperbatch.
- Some parameters specified on the ACB are either ignored or not supported. These include buffer-related parameters BSTRNO, BUFND, BUFNI, BUFSP, and SHRPOOL, which are ignored. Data identifiers DDN and DSN are also ignored. DFR is ignored and NDF is assumed for direct requests that do not specify NSP. CFX is ignored, while NFX is assumed. Control blocks in common (CBIC) and user buffering (UBF) are not supported. RMODE31 is assumed.

VSAM record-level sharing does not apply to IMS databases that are stored in VSAM data sets. IMS provides data sharing based on local shared resources (LSR) with the CF cross-invalidate capability.

Other Migration Considerations

There are a number of other planning issues that must be addressed before you can implement VSAM record-level sharing:

- **Program Evaluation:** Applications using VSAM record-level sharing benefit from increased data availability inherent in a shared environment that has read integrity and record-level locking, as opposed to control interval-level locking.

  VSAM RLS provides record-locking serialization, handles buffer invalidation, provides read integrity, and assumes the responsibility of synchronizing the data set high watermark across systems. Most programs should be compatible with the sharing environment created with this support.

You need to determine which programs at your installation will use VSAM RLS. Every program should fall into one of the following categories:

VSAM RLS tolerant program

The program operates correctly in a multi-update environment when RLS is specified in the JCL or when MACRF=RLS is specified on the ACB. The RLS JCL parameter allows batch read programs to use RLS without requiring a recompile of the program. For batch update programs running against non-recoverable VSAM spheres, it might be possible to modify the allocation from DISP=OLD to DISP=SHR.

The key requirement is that the batch update program must be designed to permit concurrent update sharing of VSAM spheres. The spheres being updated are non-recoverable. This means that VSAM RLS does not provide transactional recovery to isolate changes made by a program until the program declares COMMIT, as CICS would do in an update of a VSAM RLS recoverable sphere. The batch program must tolerate the fact that, as an individual record is changed, the change is visible to all sharing programs.

VSAM RLS exploiting program

The program recognizes when a VSAM data set can be shared at the record level and uses the functions of VSAM RLS to access the data set. CICS is an example of an exploiting program.

VSAM RLS intolerant program

The program uses facilities not supported by VSAM RLS, accesses VSAM internal data structures, or is incompatible with the functions of VSAM record-level sharing; for example, the program design might assume that no other program is concurrently updating the data set.

Before changing an application that currently uses non-shared resource (NSR) access, to use RLS access, consider the following:

- The program might not be coded to function in a multi-update environment. For shared access, the application must wait for an exclusive record lock if another user has the record, and is subject to deadlock or timeout return codes. The application might require modification to change allocations from DISP=OLD to DISP=SHR.
- Using no read integrity (NRI) for read access will give the application similar locking rules with RLS as it has with NSR. Using consistent read (CR) can result in the application waiting for the availability of a record and could result in a deadlock, timeout, or a retained lock condition with other applications.
- An explicit point is required to establish positioning at the beginning of a data set. The step is not required for single string applications using NSR.

  - A UPAD exit is ignored by VSAM RLS. The program can be modified to use the new RLSWAIT exit.
  - Control blocks other than the user ACB, RPL, and EXLST are not available to the application.
  - VSAM open processing and record management can give a return code 16, which cannot happen with NSR.
  - RLS also introduces new error codes and processing restrictions.

- **Sharing Control Data Sets:** Sharing control is a key element in maintaining data integrity in a shared environment. Give careful consideration to the allocation and maintenance of your sharing control data sets (SHCDS):
  - Allocate sharing control data sets so that the number of active and spare data sets ensures the data is always duplexed. At a minimum, define and activate two SHCDSs and one spare SHCDS for recovery purposes. Include procedures for adding to the number of data sets when I/O errors occur.
  - Place the data sets on volumes that maximize availability.
  - Ensure that the space allocation for active and spare sharing control data sets is the same. The *DFSMS/MVS DFSMSdfp Storage Administration Reference* provides formulas for calculating the size of a sharing control data set.
  - Convert reserves to enqueues on volumes where each SHCDS resides.
  - Use the commands VARY SMS,SHCDS and DISPLAY SMS,SHCDS for data set maintenance tasks.
  - Add SHCDS to the AUTHCMDS list in SYS1.PARMLIB member IKJTSO00 to allow the use of the SHCDS command in TSO foreground.
- **Recovering Catalogs:** Before the recovered catalog is made available, SHCDS CFREPAIR is run to reconstruct critical RLS information in the catalog.
- **Facility Authorization:** There are new system interactions that must have restricted access. For example, you need to determine who should have access to the new RACF facility class, STGADMIN.IGWSHCDS.REPAIR that allows a user to issue IDCAMS SHCDS commands.

  You can use the new RACF subsystem name class, SUBSYSNM, to limit access to applications (such as CICS) that are allowed to use VSAM RLS interfaces to release locks protecting uncommitted updates. Only CICS subsystems should be given this authority.
- **Data Integrity Provisions:** To help ensure data integrity in this expanded sharing environment, you need to address each of the following issues:
  - Use GRS to manage cross-system serialization of VSAM resources and other DFSMS/MVS control structures altered by VSAM RLS. Non-RLS open requests are not allowed when RLS access to a

data set is active. Non-RLS open-for-output requests are not allowed if RLS transaction recovery for the data set is pending.

– Establish procedures for non-IBM products and user programs that can modify data on a volume without using a VSAM application program interface, DFSMSdss, or DFSMShsm. If these programs run at the time VSAM RLS data exists in CF cache storage, they can compromise the integrity of the cached data. Note that operations such as DISP=OLD serialization, VSAM enqueues, or VTOC serialization do not affect data in the CF cache storage.

  Prior to running these programs, vary the volumes to be modified offline to each system in the parallel sysplex, or have them quiesced to the coupling facility by entering the VARY SMS,CFVOL(volid),QUIESCE command.

– Be aware that ICKDSF REFORMAT is not supported for SMS-managed volumes, except to rebuild a volume label using the VTOCPTR option, without changing the VOLSER. Other uses of REFORMAT will cause unpredictable results.

– Ensure that all SMS-managed volumes have unique volume serial numbers (volsers). This requirement existed before the introduction of RLS support.

  There are DFSMSdss and ICKDSF operations that work in standalone or offline mode that can change the volser of a volume. When using these operations, you must take care not to duplicate a volser for a volume that contains cached data for VSAM RLS data sets. If such a volume is varied online, it can compromise the integrity of the data on that volume, as well as the data on the volume whose volser was duplicated.

  Before varying a suspected volume online, we recommend you use the D SMS,CFVOL(volid) command to determine if there is any data in CF caches for the volser to be brought online. If there is, you must determine the disposition of that data. The suspected volume should not be varied online until the volume with the duplicate volser is varied offline to the parallel sysplex, thus ensuring that there is no data in CF cache for that volume.

- **Volume Reserves:** The use of volume reserves is inconsistent with the availability objectives of parallel sysplex management. Consider converting your reserves to enqueues, especially on volumes where the sharing control data sets will reside. You can use GRS to set this up without changing source code for the applications that use the volumes.

- **JES3 Considerations:** In a JES3 environment, it is possible that a job is scheduled to a processor that finds the CF cache structure unavailable. To help avoid this situation, existing SMS commands allow you to disable scheduling on a storage group basis. To accomplish this, you must understand both the storage class to CF cache structure mapping and the storage class to storage group mapping.

  For JES3 scheduling to work properly, the scheduling system must be at the highest level of DFSMS/MVS. If you have a single JES complex containing mixed levels of MVS/DFP and DFSMS/MVS, be careful not to specify cache set names on all storage classes. Otherwise, jobs

using any storage class could be scheduled only to RLS-capable systems. Down-level systems would not share the workload.

- **VSAM Considerations:** A VSAM sphere can be accessed for record-level sharing when the data set is opened with ACB MACRF=RLS, or when JCL parameter RLS is specified. VSAM inhibits simultaneous access to a VSAM sphere using RLS and non-RLS access protocols. A VSAM sphere is considered to be in use by RLS if it is currently accessed or is pending RLS transaction recovery (recoverable VSAM spheres only). A sphere can be opened with standard VSAM protocols (VSAM NSR/LSR/GSR) if RLS access is quiesced first. Subsystems such as CICS provide commands to accomplish this.

  VSAM RLS supports the following data set organizations: KSDS, RRDS, VRRDS, and ESDS. Path access for KSDS and ESDS data sets is also supported. Linear and key range data sets are not supported.

- **CICS Considerations:** Migration considerations for CICS are covered in the*CICS Transaction Server for OS/390 Release Guide* and the *CICS Transaction Server for OS/390 Migration Guide*. These books include information on the following:

  – Changes in file sharing between CICS regions: Prior to VSAM RLS, shared data set access across CICS Application Owning Regions (AORs) was provided by CICS function-shipping file access requests to a CICS File Owning Region (FOR). VSAM RLS can now provide direct shared access to a VSAM data set from multiple CICS regions. The highest performance is achieved by a configuration where the CICS AORs access VSAM RLS directly. However, a configuration that places a CICS FOR between CICS AORs and VSAM RLS is supported as well. FORs can continue to exist to provide distributed access to the files from outside the sysplex.

  – LOG and LOGSTREAMID parameters for DEFINE CLUSTER: These parameters replace the corresponding definitions in the CICS File Control Table (FCT). LOG specifies whether the VSAM sphere is recoverable (where CICS ensures backout) or non-recoverable. LOGSTREAMID specifies the name of a forward recovery log stream to use for the VSAM sphere.

  – BWO parameter for DEFINE CLUSTER: If VSAM RLS is used, this is the mechanism for specifying backup-while-open in a CICS environment.

  – Sharing control: Sharing control is a key element of this support. Careful consideration must be given to managing the sharing control data sets which contain information related to transaction recovery.

  – Recovery: New error conditions affect subsystem recovery. These include loss of the lock structures, loss of a CF Cache Structure, SMSVSAM server failure, or errors during backout processing.

  – Batch job updates of a recoverable VSAM sphere: VSAM RLS does not permit a batch job to update a recoverable sphere in RLS access mode. The sphere must be RLS-quiesced first, using a CICS command, and then the batch job can open the VSAM sphere for output using non-RLS protocols. If it is necessary to run

critical batch window work while transaction recovery is outstanding, there are protocols that allow non-RLS update access to the VSAM sphere. Backouts done later must be given special handling. If you intend to use this capability, you need to plan for the use of IDCAMS SHCDS PERMITNONRLSUPDATE and DENYNONRLSUPDATE commands and for the special handling of these transaction backouts.

- **Retained Lock Considerations for Data Set Recovery:** If a VSAM sphere is defined as recoverable, the locks protecting uncommitted updates or pending indoubts for the data set are preserved across all failures until backout or indoubt resolution occurs. Therefore, even if a dataset is closed, it could still have retained locks tied to it.

  As part of the process of recovering a data set, it is likely that a new data set will be created. The retained locks must be attached to the new data set.

  If CICSVR is used to recover a VSAM sphere, then CICSVR automatically manages the attachment of the locks. If a forward recovery utility other than CICSVR is used, or if IDCAMS IMPORT or REPRO is used to increase the available space in a data set after a VSAM out of space condition, the user is responsible for attaching the locks to the new data set using the SHCDS FRUNBIND and FRBIND commands. Exclusive use of the sphere must be obtained by using the SHCDS FRSETRR and FRRESETRR commands for these operations to work correctly.

  If CONVERTV SMS to non-SMS is done, then locks associated with the converted data sets will be purged. Run CONVERTV in TEST mode to determine which data sets require backout and run the backouts prior to the actual CONVERTV operation.

  COPY with DELETE for either a sphere or an AIX will not be allowed if the sphere or AIX has retained locks.

  If a volume is lost, special procedures are required to correctly manage the locks. Refer to the *CICS Recovery and Restart Guide* for more information.

- **Standalone Restore:** If the parallel sysplex fails, or if the volume is varied offline to the sysplex, or if operator command V SMS,CFVOL(volser),QUIESCE is issued prior to the operation, the data for the volume in CF caches is logically deleted.

  When the restored volume is brought online, care must be taken to ensure that the volser does not duplicate a volser that contains cached data for VSAM RLS data sets.

- **CONVERTV-SMS to non-SMS Conversion:** With a CONVERTV SMS to non-SMS conversion, DFSMSdss will issue informational messages if the data set to be converted has retained locks, or if the recovery required indicator is on, but will still allow the conversion. As part of the conversion, all associated locks will be purged.

  CONVERTV has a test mode where the actual conversion is not done, but informational messages are produced. This mode of operation can be used to determine those data sets that have outstanding recovery

so that the backouts can be done prior to running CONVERTV to actually do the SMS to non-SMS conversion.

- **IBM Language Considerations:** COBOL, PL/I, C/370 and FORTRAN run time libraries can be updated to support VSAM RLS for batch applications. This does not automatically imply that the application is a VSAM RLS tolerant program. The proper functioning of the application in a multiple update environment must be determined before RLS access is specified.

  With DD-based allocations, the RLS JCL parameter can be used to specify RLS access and read integrity options. A recompile of the application is not necessary to invoke RLS access.

  With FORTRAN dynamic allocation, the language supports the specification of RLS access through the FILEINF statement. In this case, a recompile of the program is necessary.

  Both COBOL and FORTRAN have the ability to create self-contained load modules. At a minimum, applications using these languages must relink in order to use RLS access.

- **Enabling VSAM RLS Processing:** The SMSVSAM address space is automatically started when this release of DFSMS/MVS is installed; The capability for VSAM RLS processing, however, is also dependent on the following:

  – All systems are running as a parallel sysplex. Local mode is not allowed.

  – At least two sharing control data sets and one spare sharing control data set are activated.

  – At least one CF cache structure is defined to MVS and the SMS configuration.

  – The CF lock structure IGWLOCK00 is defined to MVS and is available.

  Until VSAM RLS processing is enabled, all attempts to open a data set, where VSAM RLS is specified through the ACB or JCL, will fail. Once enabled, VSAM RLS processing continues unless one of the following conditions exist:

  – There are no sharing control data sets available.

  – The CF lock structure IGWLOCK00 is unavailable.

  – The SMSVSAM address space fails and cannot be restarted.

  – The SMS address space has a null configuration active. In this case, only existing data sets already assigned to a CF cache structure can be processed.

  – There are no CF cache structures defined in the SMS configuration or currently associated with data sets.

  In addition to VSAM RLS processing being available, the following requirements must be met for a data set to be opened for record-level sharing:

  – If a data set is currently assigned to a CF cache structure, that CF cache structure must be available.

- If a data set is not currently assigned to a CF cache structure, a cache set must be specified on the storage class and at least one of the associated CF cache structure must be available.

- **Fallback from VSAM RLS Processing:** Fallback from VSAM RLS processing involves a number of careful considerations. Plan for the following in case of fallback:
  - There could be outstanding recovery for VSAM RLS data sets. The fallback procedure results in the loss of locks protecting backouts; therefore all recovery processing must be handled first.
  - The CF caches must be quiesced.
  - There are RLS indicators in the catalog that must be reset using the SHCDS CFRESET command.
  - There might be applications active that can only function with VSAM RLS and cannot return to a VSAM NSR/LSR/GSR environment.

    Note: the SMSVSAM fallback command is not intended as a mechanism for normal or abnormal termination of the SMSVSAM server.

  Fallback procedures are detailed in the *DFSMS/MVS DFSMSdfp Storage Administration Reference*. These procedures apply to systems that are maintaining product currency. For issues related to product fallback, see “Product Fallback Considerations” earlier in this topic.

Where to Find More Information

OS/390 Parallel Sysplex Hardware and Software Migration
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Using Data Sets
DFSMS/MVS Macro Instructions for Data Sets
DFSMS/MVS DFSMSdfp Advanced Services
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS Using ISMF
DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMSdss Storage Administration Guide
OS/390 Parallel Sysplex Application Migration
CICS Transaction Server for OS/390 Release Guide
CICS Transaction Server for OS/390 Migration Guide
CICS Recovery and Restart Guide

Migration Planning for DFSMS/MVS Version 1 Release 2

This chapter presents a technical overview of the functions added or enhanced by DFSMS/MVS Version 1 Release 2:

- Backup-while-open improvements
- DFSMShsm recycle enhancements
- DFSMShsm tape I/O improvements
- DFSMShsm virtual storage constraint relief
- Distributed FileManager target server
- Enhanced volume selection for SMS
- IBM 3495 Model M10 Tape Library Dataserver
- IBM 3995 Optical Library Dataserver Models 133 and 113
- OAM objects on tape support
- OAM RMF transaction reporting
- OS/390 UNIX System Services
- Overriding expiration dates when deleting data sets
- QSAM or BSAM compression
- SMS system groups
- UCB constraint relief
- VSAM compression
- VSAM partial release

For each new support, this chapter outlines environmental dependencies and migration considerations and, where appropriate, refers to other books in the DFSMS/MVS library where you can find implementation assistance.

Backup-While-Open Improvements

Backup-While-Open support provides improved data security and integrity for the backup-while-open function.

A backup-while-open enabler, which is an application or component such as CICS, makes a data set eligible to be backed up while it is open. To make the data set eligible, the backup-while-open enabler acquires an exclusive or shared enqueue on the data set, then sets the backup-while-open status to 1xx. A status of 1xx allows the data set to be dumped or copied by another application or component without requiring the normal serialization for VSAM data sets.

A backup-while-open handler, which is an application or component such as DFSMSdss, processes VSAM data sets that are backup-while-open eligible. While the backup-while-open handlers can dump eligible data sets without acquiring the normal VSAM enqueues, the BWODSN enqueue must be obtained. The backup-while-open handler saves, alters, and checks the backup-while-open status, as required, to ensure that the data set can be used without error after the dump and restore sequences are complete.

Functional Components Enhanced
DFSMSdfp, DFSMSdss, DFSMShsm

Software Dependencies
There are no additional software dependencies.

© Copyright IBM Corp. 1979, 1999

Hardware Dependencies
There are no hardware dependencies.

Programming Interfaces Affected
No programming interfaces are affected.

Performance Characteristics
Backup-While-Open provides improved data security and integrity.

Coexistence Issues
There are no coexistence implications for this support.

Other Migration Considerations
Backup-while-open processing can only be used on SMS-managed VSAM data sets, with the exception of the VSAM volume data set (VVDS), linear data sets, and catalogs.

Data sets that are backup-while-open eligible can not be migrated by the DFSMShsm functional component.

Where to Find More Information
DFSMS/MVS DFSMSdss Storage Administration Reference

DFSMShsm Recycle Enhancements

The DFSMShsm recycle function has been enhanced to provide faster recycle processing, more efficient processing, and greater flexibility regarding the number of tape drives dedicated to recycle processing.

Three recycle tape buffers provide faster processing.

Tapes eligible for recycle processing are logically sorted, and those containing the least amount of eligible data are processed first. This allows a greater number of scratch tapes to be returned to the scratch pool early in the recycle process. And when a drive is allocated as an input drive for a tape or a connected set of tapes, it remains allocated until all input tapes have been processed.

One to 15 tape input drives can be dedicated to recycle processing. You can change the number dynamically, even during recycle processing. You can set recycle processing to automatically stop when a predefined net number of tapes are returned to scratch.

In addition, the RECYCLE DISPLAY command can generate a list showing which tapes will be selected for recycle processing.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Programming Interfaces Affected
No programming interfaces are affected.

Performance Characteristics
Throughput is increased by the use of three tape buffers and the multi-tasking capabilities.

Coexistence Issues

There are no coexistence implications for this support.

Other Migration Considerations

There are no other migration considerations.

Where to Find More Information

DFSMS/MVS DFSMShsm Implementation and Customization Guide
DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Storage Administration Reference

DFSMShsm Tape I/O Improvements

Multiple tape buffers provide DFSMShsm tape I/O improvements during migration direct to tape and automatic backup to tape. Two alternating tape buffers provide DFSMShsm tape I/O improvements during tape recycle.

Support includes:

- Multiple output tape buffers for migration and backup to tape using DFSMSdss as the data mover
- Two alternating tape buffers for recycling migration or backup tapes
- Removal of unnecessary in-storage data movements that occur when DFSMShsm software compaction is not used
- Clearing the unused portion of the final 16KB block of tape output in the migration and backup direct-to-tape path

Functional Components Enhanced

DFSMShsm

Software Dependencies

There are no additional software dependencies.

Hardware Dependencies

There are no hardware dependencies.

Programming Interfaces Affected

No programming interfaces are affected.

Performance Characteristics

Throughput is increased for backup and migration to tape by using multiple tape buffers in a manner that allows overlap of I/O processing.

Throughput is increased for recycling of backup and migration tapes by the use of two alternating tape buffers where there was formerly only one.

Coexistence Issues

DFSMShsm Tape I/O Improvement Support can coexist with previous releases of DFSMS/MVS.

Other Migration Considerations

There are no other migration considerations.

Where to Find More Information

DFSMS/MVS DFSMShsm Implementation and Customization Guide

DFSMShsm Virtual Storage Constraint Relief

DFSMShsm virtual storage constraint relief brings the DFSMShsm load module above the 16MB line in extended private storage, which increases the amount of virtual storage available for DFSMShsm processing. This increase in virtual storage increases the number of tasks that can be run simultaneously.

The DFSMShsm move above the 16MB line requires the use of 31-bit addressing. To support this, the DFSMShsm DISPLAY and PATCH commands have been changed from a 3-byte address to a 4-byte address.

Functional Components Enhanced
DFSMShsm

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no hardware dependencies.

Programming Interfaces Affected
The ARCINEXT installation exit is passed parameters with 31-bit addresses. Any DFSMShsm exit routine that needs to address 31-bit addresses needs to switch to AMODE(31) to properly address the data areas, and then switch back to AMODE(24) to return to the caller.

Performance Characteristics
Throughput is increased by DFSMShsm's ability to run an increased number of simultaneous tasks.

Coexistence Issues
Virtual storage constraint relief can coexist with previous releases of DFSMS/MVS.

Other Migration Considerations
There are no other migration considerations.

Where to Find More Information
DFSMS/MVS DFSMShsm Implementation and Customization Guide

Distributed FileManager Target Server

The Distributed FileManager target server allows remote access from systems on an SNA LU 6.2 network to physical sequential data sets, entry-sequenced, key-sequenced, and relative-record VSAM data sets, and PDSE members. The access is based on the Distributed Data Management (DDM) architecture. The Distributed FileManager can be used by any source server that conforms to the DDM architecture, such as the ADSTAR Distributed Storage Manager Data Access Services.

The Distributed FileManager interacts with other systems as an APPC/MVS transaction program. The operator starts the Distributed FileManager with the START DFM system command. APPC/MVS then uses the Distributed FileManager's transaction program profile to set up the environment. The ASCHPMxx member of the system PARMLIB controls APPC/MVS's use of system resources.

Functional Components Enhanced
DFSMSdfp, DFSMSdss

Software Dependencies

The Distributed FileManager requires APPC/MVS and VTAM for communicating with other systems on an SNA LU 6.2 network. RACF is required for checking the authorization of the remote systems to connect to MVS/ESA and to access specific data sets.

Hardware Dependencies

There are no specific hardware dependencies for the Distributed FileManager other than the requirements for the operating system.

Restrictions

There are no special restrictions to using the Distributed FileManager.

Programming Interfaces Affected

No programming interfaces are affected.

Performance Characteristics

Remote data access is necessarily slower than local data access because of the communications paths access requests must take. Restrict use of the Distributed FileManager to those applications for which remote access provides a larger benefit than the performance degradation, if there is a performance degradation. You can use APPC/MVS to tune the performance of the transaction program.

When started, the Distributed FileManager creates an address space. If your system has memory constraints, this might affect performance on the system. The memory used by the Distributed FileManager is pageable.

If a data set used on the MVS system is made available to remote users through the Distributed FileManager, MVS users of the data set might have increased contention for the data set due to the increased number of users that have access to the data set. This can result in delays or access denials for the MVS users, affecting performance for their applications.

Coexistence Issues

The Distributed FileManager maintains DDM attributes for data sets it accesses. DFSMSdss and IEBCOPY maintain these attributes when copying data sets. The IDCAMS EXPORT and IMPORT commands also preserve the attributes.

If a PDSE that contains members with associated DDM attributes is moved to a pre-Version 1 Release 2 system or an MVS/DFP Version 3 system, the attributes are ignored on that system. However, update activity to the PDSE can invalidate the DDM attributes, and if the PDSE is later moved to a system with the Distributed FileManager, remote access to members with invalid attributes can cause access errors.

Data stored in a data set by a remote user is stored in the format of the remote system. For example, ASCII data is not converted to EBCDIC data. If an application requires that the data set be usable by both the remote system and MVS users, it is the application's responsibility to perform the required data conversion.

You can add communications equipment to the system without stopping and restarting the Distributed FileManager.

Other Migration Considerations

Following are some points to consider before using the Distributed FileManager:

- Remote users can create system-managed data sets on MVS if your ACS routines allow it. You can filter on the Distributed FileManager's program name (GDEISASB) to either prevent or allow data set creation. Your ACS routines might not require changes to allow remote users to create data sets.

  The attributes used to create a data set come from the creation request from the remote system and the data class assigned to the data set, if any. If the attributes are not sufficient to create the data set, it is not created and the remote user is notified. You might want to ensure that data classes are assigned to all remotely created data sets so that remote users do not need to understand the data set allocation requirements of MVS, and need only consider their specific needs when creating a data set.

- When starting the Distributed FileManager, use the SUB=MSTR keyword on the START command to make the Master Scheduler the owner of the Distributed FileManager. If you do not use SUB=MSTR, JES is the owning subsystem, and the Distributed FileManager must be brought down before you can bring down JES, if that is ever required.

- In order to make the Distributed FileManager available to remote users, you must make the following modifications to the system:

  – Add the Distributed FileManager to the program properties table.

  – Create the Distributed FileManager transaction program profile for APPC/MVS in SYS1.APPCTP.

  – Define the local logical unit to APPC/MVS.

  – Define the local node to VTAM in SYS1.VTAMLST and the APPC logon mode in SYS1.VTAMLIB.

  – Define the partner logical units on the remote systems, and provide the APPC/MVS logon mode name to the remote applications.

  – Ensure your security program allows the remote system to connect to MVS and use the Distributed FileManager.

  These tasks are explained in *DFSMS/MVS DFM/MVS Guide and Reference*.

- If multiple MVS systems are sharing a data set used by the Distributed FileManager, those systems must be on a GRS ring to ensure proper sharing.

- Remotely accessed data sets can reside only on DASD.

Where to Find More Information

DFSMS/MVS DFM/MVS Guide and Reference
OS/390 MVS Planning: APPC/MVS Management

Enhanced Volume Selection for the Storage Management Subsystem

Storage Management Subsystem's enhanced volume selection makes it possible to combine devices with different performance characteristics in the same storage group. For example, you can include IBM 3390 Model 1, 2, and 3 volumes in the same storage group as IBM 3390 Model 9 volumes and IBM 3995 Model 153

optical volumes. We recommend you do not mix device geometries within a single storage group.

The SMS device selection algorithm is changed to allow you to more selectively target a data set allocation to a specific performance group. In previous releases of DFSMS/MVS and MVS/DFP, all devices in a storage group whose performance was equal to or better than the millisecond response time (MSR) specified in the storage class assigned to the data set were considered for the allocation. For example, if the assigned storage class specifies 25 for the MSR, any device capable of meeting or exceeding that time is considered for the allocation.

In DFSMS/MVS 1.2, and higher releases, MSR is interpreted differently. The primary list of eligible volumes only includes those volumes whose performance approximates the assigned MSR. The specific volume chosen can have a performance that is slightly better or slightly worse than the requested MSR. For example, if the assigned storage class specifies 25 for the MSR, any device whose performance is between 20 and 29 milliseconds is considered for the allocation. Faster or slower devices are only considered if the data set can not be allocated on a device that provides a response time between 20 and 29 milliseconds.

The initial access response time (IART) attribute of the storage class is used to determine whether the data set is to be allocated on removable volumes, for example, optical volumes. If the IART is zero, the data set is only allocated on fixed volumes. If it is non-zero, mountable volumes are preferred, but the data set can be allocated on fixed volumes if no mountable volumes are available.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
There are no additional software dependencies.

Hardware Dependencies
There are no additional hardware dependencies.

Restrictions
There are no restrictions other than coexistence issues.

Programming Interfaces Affected
No programming interfaces are affected.

Performance Characteristics
SMS enhanced volume selection is intended to allow you to more effectively use the variety of devices available on an MVS/ESA system, targeting data that does not require high performance to devices that provide the appropriate level of performance. Data that you target to slower devices should have slower performance than they would have if SMS had redirected the allocation to the fastest device available.

Coexistence Issues
If you mix mountable and fixed volumes in a storage group shared among systems, all sharing systems should be using DFSMS/MVS 1.2, or a higher release. Mixing mountable and fixed volumes in a storage group shared with systems running down-level releases can generate unpredictable results.

Other Migration Considerations

SMS enhanced volume selection changes how the system interprets the MSR and IART storage class attributes. It also takes the availability and accessibility attributes into account when choosing a volume from the secondary volume list if there are no eligible volumes from the primary list available.

In order to take full advantage of SMS enhanced volume selection:

- Consider combining volumes of the same device geometry that are serving the same purpose (for example, your primary storage group volumes) in a single storage group, regardless of their performance characteristics. This simplifies your storage group configuration.
- Examine the storage classes, keeping the following points in mind:
  - If the IART is zero, the data set can only be allocated on fixed volumes.
  - If the IART is non-zero, the data set is allocated on mountable volumes if a mountable volume of the requested MSR is available. If such a volume is not available, and a fixed volume is available that comes closest of all available volumes to meeting the MSR, the data set is allocated on the fixed volume.

    Because fixed volumes can be used with non-zero IART data sets, make sure the MSR values for storage classes with non-zero IARTs accurately reflect the performance required for the data.
  - If an MSR value is not specified, SMS uses the following defaults:

    10 for read bias
    6 for write bias
  - If cache is available for a device, the device's cache performance values are used to determine its MSR. Otherwise, the device's native performance values are used.
  - If a data set is allocated on a device based on its cache performance values, and the device's native performance is lower than the MSR requested, the data set is considered *must cache* when it is opened.
  - An MSR of 999 is interpreted as 25. This ensures that *never cache* data sets continue to be allocated on fixed volumes.
  - When selecting a device based on the MSR values, SMS averages the values when:
    - More than one MSR is specified.
    - An MSR is specified without a corresponding bias.
    - All MSRs and biases are blank.
  - The availability and accessibility attributes are considered for volumes in both the primary and secondary selection lists.
  - SMS tries to allocate the data set on the volume that most closely approximates the MSR target specified in the storage class, even if the closest match is slower than the value specified. The MSR does not define a lower extreme for response time.

Where to Find More Information

DFSMS/MVS DFSMSdfp Storage Administration Reference

IBM 3495 Model M10 Tape Library Dataserver

The IBM 3495 Model M10 Tape Library Dataserver extends the scope of tape library support, introduced in DFSMS/MVS Version 1 Release 1, to include mountable tape cartridges in an SMS-managed manual tape library dataserver. A manual tape library dataserver is an installation-defined set of tape drives and the set of volumes that can be mounted on the drives. An SMS-managed manual tape library dataserver consists of an IBM 3495 Model M10 and one to four IBM 3490 Magnetic Tape Subsystems.

Functional Components Enhanced

DFSMSdfp, DFSMShsm, DFSMSrmm

Software Dependencies

There are no dependencies on any MVS/ESA SP operating system component.

Hardware Dependencies

There are no additional hardware dependencies.

Programming Interfaces Affected

The CBRXLCS macro has a new function, MCE, that allows users to enter volumes into a manual tape library dataserver and to modify volume information in the tape configuration database. The CBRLCSPL macro is changed to map additional fields required for the new function.

A new installation exit, CBRUXVNL, gives you the opportunity to insert a volume into a tape library dataserver during job step setup, device allocation, or library mount processing. The CBRUXNPL mapping macro maps the parameter list for this exit.

Several installation exits with the prefix CBR are changed. For more information on these exits, see *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support* and *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

Performance Characteristics

IBM 3495 Model M10 manual tape library dataserver support extends SMS-management to the non-automated tape storage environment, providing the following benefits:

- System-managed tape outside of an automated tape library dataserver
- Remote ESCON connected tape subsystems in a library
- A mix of manual and automated tape library dataservers within an SMS complex
- SMS control and experience when planning a migration to an automated environment

Coexistence Issues

All software support for the IBM 3495 Model M10 is compatible with support for existing IBM 3495 models.

Where to Find More Information

DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries

IBM 3995 Optical Library Dataserver Models 133 and 113

The OAM component of DFSMS/MVS is enhanced to support two new optical library dataservers, the IBM 3995 models 133 and 113. The model 133 is a channel-attached optical disk library with multi-function optical disk drives (WORM and read-write). The model 113 is an expansion unit that attaches directly to the model 133. These new optical library dataservers allow greater flexibility in the storage of objects.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no dependencies on any MVS/ESA SP operating system component. There is one EREP dependency. EREP provides a Version 3.5.0 PTF to recognize a valid UCB unit type for IBM 3995 models.

Hardware Dependencies

There are no additional hardware dependencies.

Programming Interfaces Affected

No programming interfaces are affected.

Performance Characteristics

The optical disk drives in the IBM 3995 models 133 and 113 are *multi-functional*. These drives can be defined as either library-resident (models 133 and 113) or operator-accessible (model 133) optical disk drives. These drives are capable of reading and writing:

- Single-density WORM 5.25-inch optical disk media
- Double-density WORM 5.25-inch optical disk media
- Single-density rewritable 5.25-inch optical disk media
- Double-density rewritable 5.25-inch optical disk media

The multi-functional optical disk drive capability allows the flexibility of populating the models with any combination of four different types of optical disk media. This allows the user to load different optical disk media types into a single 133 or 113 and to direct data for one application to WORM optical disk media for instance, and data for another application to rewritable optical disk media within the same optical disk drive.

Coexistence Issues

The new models of the IBM 3995 optical library dataservers can coexist with the existing models (111, 112, 131, and 132) on the same MVS/ESA system.

Other Migration Considerations

Two new columns will be added to the library and volume tables in the optical configuration database through the invocation of sample job CBRSMIG4.

Where to Find More Information

DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support

DFSMS/MVS DFSMSdfp Advanced Services
DFSMS/MVS Utilities
DFSMS/MVS Installation Exits
OS/390 MVS JCL User's Guide

OAM Objects on Tape Support

OAM objects on tape support enables OAM to store objects on standard and enhanced capacity cartridge system tapes (IBM 3480, 3490, and 3490E). Although all objects are system-managed, the tapes on which they reside can be either system-managed or non-system-managed.

Functional Components Enhanced
DFSMSdfp, DFSMSrmm

Software Dependencies
There are no dependencies on any MVS/ESA SP operating system component.

Hardware Dependencies
There are no additional hardware dependencies.

Programming Interfaces Affected
You can now use the OSREQ macro with tape.

Performance Characteristics
Objects can now be stored on tape devices, allowing an installation to incorporate various types of storage devices (DASD, optical, tape) into an object storage environment. This support allows a primary copy of an object to be stored on tape cartridges and allows an object that was originally written on DASD or optical storage devices to be migrated to tape. Objects can be among the different types of supported devices.

Coexistence Issues
There are no coexistence considerations for objects on tape.

Other Migration Considerations
A new PARMLIB member (CBROAM*xx*) is used for object to tape processing. This PARMLIB member can contain one or more SETOAM commands. The SETOAM commands can set attributes for all of OAM's object tape processing, or the attributes for a specific object or object backup storage group. These commands can determine the types of tape drives that are to be allocated to the OAM session, storage group assignments, and thresholds for tape drive capacity and volume utilization percentages, among other attributes.

You must modify the OAM cataloged procedure to include an OAM=*xx* parameter to point to the CBROAM*xx* member in order for OAM to be able to place objects on tape.

Two new SAMPLIB members are included in this release to update the optical configuration database in support of object to tape processing. If you have installed the MVS/DFP 3.3.1 PRPQ for object to tape processing, use member CBRSMIG4 to update the database for DFSMS/MVS 1.2. If you do not have the PRPQ installed, use member CBRSAMMT to create the required tape-volume table.

If you are using a tape management system like DFSMSrmm, define appropriate policies to ensure that only OAM uses tapes that contain objects.

OAM writes all objects to data sets named OAM.PRIMARY.DATA or OAM.BACKUP.DATA. See *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support* for more information on how OAM stores data on tape.

Where to Find More Information
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support

OAM RMF Transaction Reporting

The Resource Measurement Facility (RMF) is a tool that allows an installation to obtain reports on workload and transaction activity against specific report performance groups. These reports provide the installation with the ability to monitor and obtain a quick snap-shot of OAM's performance at a given time. The RMF transaction reporting enhancement assists customers using OAM for a variety of applications in performance monitoring, analysis, and tuning.

Functional Components Enhanced
DFSMSdfp

Software Dependencies
RMF is required to monitor OAM.

Hardware Dependencies
There are no additional hardware dependencies.

Programming Interfaces Affected
No programming interfaces are affected.

Performance Characteristics
Reporting RMF transaction completion messages by OAM takes additional resources (CPU cycles). However, the additional processing time needed to report the transaction completion messages to RMF is minimal and should not noticeably degrade OAM performance.

Coexistence Issues
There are no coexistence considerations for RMF transaction reporting

Other Migration Considerations
OAM reports transaction completion messages to the MVS system resource manager (SRM). If RMF transaction reporting is active, the transaction complete messages are passed to RMF. If the appropriate report performance groups are defined, RMF reports statistics related to the report performance groups.

You can obtain two types of reports from RMF:

1. RMF Monitor I Workload Activity Report
2. RMF Monitor II Transaction Activity Report

Where to Find More Information
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support

OS/390 UNIX System Services

OS/390 UNIX System Services (OS/390 UNIX), formerly known as OpenEdition, provides a way for users to access data organized in a hierarchical file system on MVS using a shell similar to that available on UNIX systems. DFSMS/MVS provides the hierarchical file system (HFS) for the OS/390 UNIX environment.

With OS/390 UNIX, users can choose between regular MVS facilities and UNIX-like shell facilities to access data in the hierarchical file system. They can copy data between data sets and hierarchical file system files.

Functional Components Enhanced

DFSMSdfp, DFSMSdss

Software Dependencies

To use the OS/390 UNIX System Services available in DFSMS/MVS, you must have the OS/390 UNIX features that are available with MVS/ESA SP. For other information on the installation, use, and management of OS/390 UNIX, see *OS/390 UNIX System Services Planning*.

Hardware Dependencies

There are no special hardware requirements for OS/390 UNIX besides the basic requirements for DFSMS/MVS.

Restrictions

Hierarchical file system data sets must be system-managed.

Although you can use the binder to create executable files in a hierarchical file system data set, you cannot create overlay modules. Also, if the executable file you are creating has the same name as an existing file in the data set, the existing file is replaced, even if it is not executable, and even if you specify that existing files should not be replaced.

Executable files in an HFS data set cannot have alternate entry points. Any alternate entry points specified are considered true aliases.

When saving an executable file in an HFS data set, the binder does not take the save or stow exits.

Checkpoints cannot be taken in steps that are using OS/390 UNIX System Services.

Programming Interfaces Affected

The IGWASMS callable service can be used to determine if a data set is a hierarchical file system data set.

Performance Characteristics

OS/390 UNIX is not expected to alter the performance of the MVS system. Hierarchical file system data sets are expected to perform better than a logically equivalent PDSE.

Coexistence Issues

Hierarchical file system data sets can only be accessed by systems running OS/390 UNIX. Toleration PTFs are available for currently supported releases of DFSMS/MVS and MVS/DFP that do not support OS/390 UNIX to ensure that systems that cannot process HFS data sets do not try to process them.

Other Migration Considerations

Hierarchical file system data sets are allocated by specifying HFS in the DSNTYPE parameter. You may need to update the ACS routines to ensure all HFS data sets are assigned a storage class.

Where to Find More Information

OS/390 UNIX System Services Planning
OS/390 UNIX System Services File System Interface Reference

Overriding Expiration Dates When Deleting Data Sets

By specifying OVRD_EXPDT(YES) in the IGDSMSxx member of SYS1.PARMLIB, you can indicate whether the system should ignore DASD data set expiration dates when a data set is deleted through JCL, IEHPROGM, ISPF D or DEL line operators, the SCRATCH macro, or through dynamic unallocation.

This makes it possible for a user to delete data sets through JCL that previously required operator intervention. This is especially helpful if you are redirecting tape data sets to DASD volumes when the tape data sets are assigned expiration dates. However, if users at your installation frequently assign expiration dates to DASD data sets, specifying OVRD_EXPDT(YES) can lead to users accidentally deleting data sets they want to keep.

We recommend that you use management classes to assign expiration dates to those data sets that require expiration dates.

This installation default does not apply to the DELETE command (TSO or IDCAMS). Users still must specify PURGE on the DELETE command to delete unexpired DASD data sets. This function is not applicable to ICKDSF because ICKDSF does not test for expiration dates.

Functional Components Enhanced

DFSMSdfp

Software Dependencies

There are no dependencies on other software products.

Hardware Dependencies

There are no dependencies on hardware products.

Restrictions

The installation default only applies to attempts to delete DASD data sets through JCL, IEHPROGM, the SCRATCH macro, the ISPF D and DEL line operators, and dynamic unallocation.

Programming Interfaces Affected

No programming interfaces are affected.

Performance Characteristics

The installation default should not affect performance.

Coexistence Issues

The installation default only applies to the system on which it is defined. It has no affect on connected systems.

Other Migration Considerations

Users might be surprised by the changed behavior of familiar commands if you set an installation default that overrides expiration dates. Be sure they

understand the system change so that they do not unintentionally delete data sets.

The default for the OVRD_EXPDT keyword is NO. Thus, if you do not want expiration dates to be overridden, you do not need to do anything to the system. It handles expiration dates as previous releases of DFSMS/MVS and MVS/DFP handled them.

Where to Find More Information
DFSMS/MVS DFSMSdfp Storage Administration Reference

QSAM or BSAM Compression

BSAM or QSAM compression provides a technique for compressing physical sequential data sets that are processed with BSAM or QSAM to reduce the amount of storage required to house the data on DASD.

BSAM or QSAM compression takes advantage of the hardware-based compression facility available on selected processors, and uses a software compression facility when running on processors that do not have the hardware compression available.

Only SMS-managed extended format data sets can be compressed. A data set is compressed if it is assigned a data class when created that specifies EXT for the DSNTYPE attribute and YES for the COMPRESSION attribute. The only method in which JCL can be used to create a compressed data set is through the DD statement keyword LIKE, where the parameter references a data set that is already in the compressed format.

Functional Components Enhanced
DFSMSdfp, DFSMSdss, DFSMShsm

Software Dependencies
BSAM or QSAM compression requires MVS/ESA SP Version 5.

Hardware Dependencies
Extended format data sets eligible for compression must reside on DASD attached to cached storage controls with the Extended Platform.

For optimal performance, this support uses the data compression feature available on some processors, for example on the IBM 9021 Model 711 and IBM 9121 Model 511 processors. If the hardware is not available, however, it uses the software compression facility to compress data.

Restrictions
Extended format data sets defined as compressible cannot be opened for update processing. Other restrictions for extended format data sets also apply.

Programming Interfaces Affected

User programs and program products should not need changes to access the compressed data. Although there are no syntax changes for BSAM or QSAM macros, there are some changes to the way the following macros are handled for compressed data sets: NOTE, POINT, READ, WRITE, CHECK, and SYNADAF. In addition, certain fields in the DCB and DCBE control blocks might have slightly different meanings for compressed format data sets.

Any program, other than DFSMS/MVS, that performs direct I/O to DASD to change data sets in the compressed format could compromise data integrity. The procedure for running these programs must be modified to access the data using BSAM or QSAM to permit expansion of the compressed data.

Changes to system interactions used by a system programmer or storage administrator are localized to the ISMF and SMF facilities. Compression related information is available in ISMF through the Data Set Application. SMF record types 14 and 15 are modified to record compression information when a data set is closed.

Because users cannot anticipate the amount of space required for compressed data sets, they should consider allocating the space required for uncompressed data, using partial release to release unused space. A user can release unused space directly by including RLSE on the JCL SPACE parameter, or a storage administrator can ensure that management classes that are assigned to compressed data sets specify partial release.

Performance Characteristics

For I/O-bound systems, host-based compression should have a positive effect on all system variables that impact job throughput.

Coexistence issues

Although the BSAM or QSAM compression function can coexist with previous releases of MVS/DFP or DFSMSdfp, compressed data sets can only be allocated and used on a system that has the compression facility available. Read or write access to compressed data through a down-level system results in an ABEND 013-nn. Consider the following environmental conditions:

- **Scenario:** A job running on a down-level system that does not support access method compression attempts to access an extended format data set that is compressed. **Result:** MVS/DFP Version 3 systems fail the open because the data set is in the extended format; DFSMS Version 1 Release 1 fails the open because the data set is compressed.
- **Scenario:** A job runs on a system that has access method compression but does not have the compression facility available. It attempts to allocate a new or existing compressed data set. **Result:** A *new* allocation proceeds but the resulting data set is not compressed; a message is issued. An *existing* allocation abends during the open process.
- **Scenario:** A job accesses compressed data sets on a system that has the software compression facility but does not have a compression-capable processor. **Result:** The application executes successfully but experiences increased cycle utilization.
- **Scenario:** A compressed data set is restored to a system that supports access method compression and has the processor compression facility available, but does not have the prerequisite storage hardware. **Result:** The data set is converted to a non-compressed, non-extended format data set when restored.
- **Scenario:** A compressed data set is restored to a system that supports access method compression, but does not have the compression facility available. **Result:** The target data set is allocated but then fails

during the conversion. Subsequently, the restore process fails and messages are issued.

Other Migration Considerations

Before you convert an application's data to extended format with compression, you should analyze the compressibility characteristics of the data and the performance requirements of the application. Some of the factors for the application are:

- Processor cycles available to the application
- Probable I/O reduction due to compressed data
- Probable reduction in channel, controller, and device busy when processing compressed data
- Probable increased buffering of data in processor and storage control cache

Additional considerations for the data:

- Data set must be SMS-managed.
- Data set must be defined as extended format.
- Primary allocated space must be greater than 8MB if the data set is defined without secondary extents and greater than 5MB if allocated with secondary extents.
- Record length for the data must be greater than 40 bytes.

When DFSMShsm backs up a compressed data set, it does not try to compress it further even if you are using DFSMShsm software compression. Compressed data sets can be written to SDSP data sets if they meet the size requirement.

Where to Find More Information

DFSMS/MVS Using Data Sets
DFSMS/MVS Macro Instructions for Data Sets
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Using ISMF
DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Implementation and Customization Guide
DFSMS/MVS DFSMSdss Storage Administration Guide

Storage Management Subsystem System Groups

An *SMS complex* is a system (an MVS image) or collection of systems that share a common SMS configuration. The systems in an SMS complex share both the active control data set (ACDS) and the communications data set (COMMDS) for that configuration.

The systems allowed to share an SMS policy are defined in the base configuration, which previously accommodated up to eight systems. This support alleviates the eight-system limitation. Although an SMS configuration can still only contain eight names, those names can now be a combination of both system names and system group names. A *system group* is all systems that are part of the same parallel sysplex and are running SMS, minus any systems in the parallel sysplex that are

explicitly defined in the SMS configuration. This enables an enterprise to implement a single SMS policy across multiple systems.

Functional Components Enhanced

DFSMSdfp, DFSMShsm

Software Dependencies

DFSMS/MVS does not require that the systems in a parallel sysplex be connected to a coupling facility. The systems can operate in a “single system parallel sysplex mode” with a common parallel sysplex name (all the COUPLExx parmlib members specify the same COUPLE parallel sysplex name).

Hardware Dependencies

Some SMS complex configurations might require DASD with enhanced connectivity. Because all systems in an SMS complex share the same configuration data sets, the ACDS and COMMDS must reside on DASD devices that are accessible to all systems in the SMS complex. The source control data set (SCDS) must reside on a DASD device that is accessible to the system activating the configuration.

If you specify a large number of systems in an SMS complex, consider connectivity and path requirements for the devices where the following SMS control data sets reside:

- SCDS, backup SCDS, and fallback SCDS
- ACDS, spare ACDS, and fallback ACDS
- COMMDS, spare COMMDS

Restrictions

This support cannot be exploited in a JES3 environment.

Any system connected to optical libraries must be defined by system name in the SMS configuration. Also, any systems not running DFSMS/MVS 1.2, or higher, must be defined by system name. Although you can define systems connected to tape library dataservers with a system group name, we recommend you use the system name.

Programming Interfaces Affected

No programming interfaces are affected.

Performance Characteristics

No increase or degradation in performance is anticipated. However, you might have increased contention for the COMMDS or ACDS depending on how many systems are sharing them.

You should use GRS to convert any hardware reserves on the COMMDS and ACDS to SYSTEMS scope enqueues to avoid contention problems. The major name for the reserve is IGDCDSXS and the minor name is the data set name. Place resource name IGDCDSXS in the RESERVE conversion resource name list (RNL) as a generic entry. This will minimize contention delays and prevent deadlocks which are sometimes associated with the VARY SMS command.

Coexistence Issues

This support can coexist with previous SMS-supported releases provided that systems not running DFSMS/MVS 1.2, or higher, are defined by system name and requisite PTFs are applied to the down-level systems.

If you use a system group name in the SMS configuration, all members of the system group must be running DFSMS/MVS 1.2 or higher. Down-level versions of DFSMS/MVS or MVS/DFP must be defined by system name.

Coexistence restrictions for the following environments are unchanged:

DFSMS/MVS 1.2.0 and MVS/DFP Version 2
DFSMS/MVS 1.2.0 and SMS-capable systems that are not active

Other Migration Considerations

DFSMS/MVS recognizes system group names in its SMS, ISMF, and DFSMShsm operations. Note that:

- There are several differences between a system group in an SMS complex and a parallel sysplex:
  - When a parallel sysplex name is used in an SMS configuration, it must be a name that is also specified on the SYSPLEX parameter in the COUPLExx member of the system PARMLIB. A parallel sysplex named in the SMS configuration represents a system group—all the systems in the parallel sysplex that are not explicitly specified in that configuration.
  - The SMS complex does not need to mirror the parallel sysplex. You can also configure your SMS complex to contain a system group name and individual system names. Although the base configuration accepts multiple system group names, we recommend that you map a single parallel sysplex to a single SMS complex.
  - You can include a system that is a member of a parallel sysplex in the SMS configuration without also including the name of the parallel sysplex in the configuration as a system group. The inclusion of a system in the configuration does not affect other systems that might be included in a parallel sysplex with that system.
  - It is possible to define both the system name and the parallel sysplex name (which encompasses that system) in the SMS configuration. In this case, the more explicit of the two is used, and the system is considered to be defined by its system name—it is not part of the system group. The other systems in the parallel sysplex are still defined for SMS by the parallel sysplex (system group) name.
- SMS storage resources are defined with an SMS status (for example, ENABLE, DISALL, DISNEW) for each system defined in the base configuration. Usually this status defines the availability of the resource from each system's perspective. For those systems represented by a system group name, however, resource availability is known only at the system group level. You cannot control the availability of SMS storage resources at the individual system level unless you explicitly specify that system in the base configuration.
- For most operations, ISMF treats the specification of a system group name as if it were a system name. In the case of the ISMF Storage Group Application, however, processing does differ. When you specify MIGRATE, BACKUP, or DUMP SYSTEM NAME for a pool storage group, ISMF cannot verify that the system names are defined in the base configuration—for a system group, the individual system names

are not present. Take special care entering values into these system name fields to ensure that DFSMShsm processing occurs as expected.

- DFSMShsm processes storage groups for automatic space management, data availability management, and automatic dump processing in the same manner as before; however, it now uses different criteria to determine which system or systems perform the storage group processing:
  - If the system name is blank, any system can process the storage group.
  - If the system name is a system group name, a subset of systems can process the storage group.
  - If the system name is a single system name, a specific system can process the storage group.
- If you have SMS-managed optical or tape libraries, you must explicitly specify all systems that require access to those libraries by system name in the SMS configuration.
- For recovery purposes, leave one of the eight system or system group name slots blank in the SMS base configuration. If a problem occurs that requires that the name of a system that is defined in the configuration as part of a system group be specified explicitly (such as a hardware failure), the system name can be added to the base configuration without disrupting the SMS complex. Use the following steps to recover in these situations:
  1. Use the ISMF Control Data Set Application to add the system name to the SMS configuration.
  2. Activate the configuration.
  3. Use the VARY SMS operator command to make any necessary changes to the configuration.
  4. After correcting the problem, remove the system name from the configuration and reactivate the configuration.
- If you add a system to a parallel sysplex, and that parallel sysplex is defined as a system group in the SMS configuration, the system is automatically added to the SMS complex when you reactivate the configuration. You do not have to change the SMS configuration.

Where to Find More Information

DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS DFSMShsm Storage Administration Guide

Unit Control Block Constraint Relief

The unit control block (UCB) is the software control block that represents a single addressable device to the MVS/ESA operating system. Previously the number of UCBs was limited to 4096 because the device number was represented by a 3-digit field in the UCB. The ultimate number of devices was also limited because UCBs could reside only below the 16MB line.

The UCB constraint relief, increasing device numbers from 3 digits to 4 digits, enables you to define more devices than previously possible on MVS/ESA systems, providing up to 65,536 device numbers. The hardware configuration of your

processor and I/O devices and the virtual storage below 16MB, determines how many devices you can attach to your system. With the UCB prefix now located above 16MB, storage below the line is freed for approximately 1500 more UCBs. Together, these enhancements provide addressability to large numbers of devices needed to support large processors and parallel sysplexes.

Functional Components Enhanced

DFSMSdfp, DFSMShsm, DFSMSdss, DFSMSrmm

Software Dependencies

MVS/ESA SP Version 5 is required to define 4-digit device numbers.

Hardware Dependencies

Support of 4-digit device numbers requires a processor supported by MVS/ESA SP Version 5; at minimum, a processor of 370-XA I/O architecture.

Programming Interfaces Affected

All programs and interfaces that accept device numbers as inputs must recognize device numbers with either 3 or 4 digits. Review installation-written programs and vendor software to identify changes that must be made to support the new 4-digit device numbers.

Performance Characteristics

No increase or degradation in performance is anticipated.

Coexistence Issues

Toleration PTFs for accessing the UCB prefix above the 16MB line are provided for MVS/DFP 3.3 and DFSMS/MVS 1.1.

Other Migration Considerations

To specify a 4-digit device number in JCL, precede the address with a slash (/) in the UNIT parameter: `UNIT=/nnnn`. The slash also works with 3-digit device numbers: `UNIT=/nnn`. The UNIT parameter also accepts a generic unit name (for example, 3480) or esoteric unit name (for example, SYSDA) in addition to device numbers.

No changes to existing JCL are required unless jobs that currently use demand allocation have their devices moved to 4-digit device numbers.

The ISMF Library Management Application requires migration or creation of an Optical Configuration Database with 4-digit device numbers.

You cannot place a CVOL on a volume with a 4-digit device number.

Where to Find More Information

DFSMS/MVS Managing Catalogs
DFSMS/MVS Macro Instructions for Data Sets
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS DFSMSdss Storage Administration Guide
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support
DFSMS/MVS OAM Application Programmer's Reference
OS/390 MVS JCL User's Guide

VSAM Compression

VSAM compression provides a method for reducing space required to store the data component of a key-sequenced data set.

This support is available only for SMS-managed key-sequenced data sets defined in the *extended format*. All alternate indexes for an extended key-sequenced data set must also be in the extended format.

A data set is compressed if, when created, it is assigned a data class with EXT specified for the DSNTYPE attribute and YES for the COMPRESSION attribute. A user can code this DATACLAS on the JCL DD statement for the data set. The only other method in which JCL can be used to create a compressed data set is through the DD statement keyword LIKE, where the parameter references a data set that is already in the compressed format.

Because data is compressed at the record level, this support helps to relieve the current restriction of a 4GB data limit by increasing the amount of user data that can be contained within a single data set. This increase is proportional to the ratio of compression.

Functional Components Enhanced

DFSMSdfp, DFSMSdss, DFSMShsm

Software Dependencies

VSAM compression requires MVS/ESA SP Version 4 Release 3 with Compression Facility PTFs, or MVS/ESA SP Version 5.

Hardware Dependencies

Extended format data sets eligible for compression must reside on DASD attached to cached storage controls, for example, the 3990 Model 3 with the Extended Platform or the IBM 3990 Model 6.

For optimal performance, this support uses the data compression feature available on some processors, for example, on the IBM 9021 Model 711 and IBM 9121 Model 511 processors. If the hardware is not available, however, it uses the software compression facility to compress data.

Restrictions

Control interval (CI) mode processing is not permitted when accessing a sphere containing a key-sequenced data set (the data set and its associated alternate indexes) that is defined to allow for compressed data. Control interval mode processing is only allowed for VERIFY and VERIFY REFRESH processing.

Improved control interval (ICI) processing is not allowed for extended format data sets.

Extended key-sequenced data sets cannot be defined with key ranges (KEYRANGE parameter) or with an imbedded sequence set (IMBED parameter).

You cannot use Hiperbatch processing with extended format data sets.

Programming Interfaces Affected

There are no changes to application program interfaces.

SMF record type 64 is modified to record compression information for a data set.

Performance Characteristics

VSAM compression exploits the host-based compression algorithms—a method that not only reduces space required to house data sets but also has a positive impact on system variables that impact job throughput.

Coexistence Issues

Extended key-sequenced data sets cannot be accessed on releases prior to DFSMS/MVS 1.2 due to new processing requirements. If an extended KSDS is opened for load (create) mode processing on a system that cannot recognize an extended format allocation, the data is written in a manner that makes it unusable on a system that does support extended format data.

Toleration PTFs are provided for previous levels of DFSMSdss and DFDSS to prevent an extended KSDS from being logically dumped, restored, or copied.

Because the data format of an extended format VSAM data set is not compatible with the data format of a standard format VSAM data set, you must use VSAM to access the records of an extended format VSAM data set.

Other Migration Considerations

You might want to convert an application's data to use VSAM compression processing if the application uses data that is highly compressible based on the structure or type of data. If the data is not compressible, or is only compressible to a small degree, the records in the data set can be larger than they would be in an uncompressed data set due to the additional control information required by VSAM. Considerations with regard to the structure of the data include:

- The data from the beginning of the record to the end of the key is not compressed. Thus, the closer the key is to the front of the record, the more compressible is the data set. If the key is within the last 40 bytes of the maximum record length, VSAM does not compress the data set.
- Records should be large relative to the size of the control interval. The length of the actual data, not including the key offset and key length, must be at least 40 bytes in order for the data set to be eligible for compression.
- Smaller control interval sizes might be desirable due to the random structure of the data.
- When a smaller CI size is used, the length of the data could require a spanned data set.

  Records placed in a spanned data set might not need to span control intervals when compression is used. The result is improved performance when VSAM processes the data because the amount of I/O required to get or put the record is reduced.
- To be eligible for compression, the data set must be allocated with at least 5MB primary space, or 8MB primary space if you do not specify a secondary space quantity.

Although locate mode processing is supported, we discourage its use. Performance benefits provided by locate mode processing do not exist with VSAM compression.

Consider assigning a management class that specifies partial release to compressed extended key-sequenced data sets. Because users cannot determine the space required to hold compressed data, they should allocate the data set with the space required for uncompressed data. This could waste DASD space unless partial release is used to allow the system to reclaim unused space.

Where to Find More Information

DFSMS/MVS Managing Catalogs
DFSMS/MVS Using Data Sets
DFSMS/MVS Macro Instructions for Data Sets
DFSMS/MVS Access Method Services for ICF
DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS Using ISMF
DFSMS/MVS DFSMShsm Storage Administration Guide
DFSMS/MVS DFSMShsm Implementation and Customization Guide
DFSMS/MVS DFSMSdss Storage Administration Guide

VSAM Partial Release

You can use partial release to free allocated but unused space in VSAM data sets allocated in the extended format. Space is released from the high-used RBA, if it is on a control area boundary, or the next control area boundary, to the high-allocated RBA. Space is only released from the data component of the data set.

Partial release can be used by specifying RLSE on the JCL SPACE keyword on the DD statement when creating a data set using JCL, by assigning the data set a management class that specifies partial release, by using the PARTREL macro, or by using the DFSMSdss RELEASE command.

Functional Components Enhanced
DFSMSdfp, DFSMShsm

Software Dependencies
There are no other software dependencies.

Hardware Dependencies
Extended format data sets must reside on DASD attached through cached storage controls (for example, the 3990 Model 3 with the Extended Platform or the IBM 3990 Model 6).

Restrictions
Only VSAM data sets allocated in the extended format can use partial release.

Programming Interfaces Affected
The PARTREL macro is modified so that it can be used on VSAM data sets allocated in the extended format. The VSAMTRKS parameter on the PARTREL macro contains the amount of space that was released. Additional new PARTREL keywords are: VSAMTYPE, VSAMFMT, VSAMCLLR, VSAMSEC.

When space is released, the SMF type 64 record contains the amount of space that was released.

The output of the IDCAMS DCOLLECT command contains a field that reports on over-allocated space for VSAM data sets in the extended format.

Performance Characteristics

Performance is not expected to be affected.

Coexistence Issues

If management classes that specify partial release are assigned to data sets that are converted to the extended format, the partial release specification applies to them whereas it did not apply before the data sets were converted. Before converting VSAM data sets to the extended format, determine if you want to use partial release on those data sets, and ensure that the management class assigned to the data sets reflect your intentions.

Users who use direct processing to load VSAM data sets by opening the empty data set, loading one record, and then reopening it for output, should be cautious when using partial release. If partial release is specified, most of the primary allocation is released after the first close, and VSAM must acquire more space for the data set almost immediately after reopening the data set.

Other Migration Considerations

Space is not released from data sets allocated with the guaranteed space storage class attribute. Space is only released if the data set has been opened for output, and all users who have the data set open close it.

The method used for extending VSAM data sets that have had space released is different from the standard method. If space has been released from a data set, and the data set needs to be extended, the system obtains an amount of space which is the difference between the initial primary extent and the current amount of space used by the data set.

Once the data set has reached the size of the primary extent, VSAM obtains secondary extents and handles the secondary extents like the primary extent. For example, once the primary space is filled, VSAM obtains a secondary extent. If that extent is not filled, the excess space is released and the next time the data set is extended, VSAM obtains the amount of space that was released—it does not obtain a full secondary extent. Once that space is filled, VSAM obtains a new full-sized secondary extent.

For multivolume data sets, you can also control the amount of space obtained for the first extent on new volumes using the additional volume amount data class attribute. You can use this attribute even if you do not specify partial release for the data set. You can specify that the initial extent on a new volume be equal to the primary space request or the secondary space request. Additional extents after the first on new volumes are obtained using the secondary space request.

For example, suppose you request 5 cylinders of primary space and 3 cylinders of secondary space, and indicate that the additional volume amount should be primary space, when you allocate a VSAM data set in extended format. The data set is allocated on volume A. After some processing, the system runs out of space on volume A and extends the data set to volume B. The first extent obtained on volume B is 5 cylinders. Any additional extents obtained on volume B are 3 cylinders.

Where to Find More Information

DFSMS/MVS DFSMSdfp Storage Administration Reference
DFSMS/MVS DFSMSdfp Advanced Services

Migration Planning for DFSMS/MVS Version 1 Release 1

This chapter highlights planning information for functions introduced in DFSMS/MVS Version 1 Release 1. These functions include:

- Aggregate backup and recovery support
- BSAM, QSAM, BPAM, and BDAM 31-bit addressing support
- Concurrent copy
- Data set performance statistics
- Device console service improvements
- Dynamic cache management enhancements
- Empty data set support enhancements
- Extended format sequential data sets
- PDSE copy performance improvements
- PDSE cross-system sharing support
- Program management enhancements
- Tape library support
- Volume Mount Analyzer

Aggregate Backup and Recovery Support

Aggregate Backup and Recovery Support (ABARS) enables you to set up SMS management classes to more easily manage your aggregate groups. Aggregate groups are typically groups of data sets that you want backed up together for disaster recovery purposes. SMS management classes include retention attributes for aggregate groups. ABARS support includes:

- The ability to create multiple aggregate backup copies
- The ability to include the ICF user catalog and alias definitions in the aggregate backup
- The automatic calculation of required storage for recovery
- Expanded naming conflict resolution
- A reduction in DASD space required for recovery by allowing recovery to tape
- An option that deletes source user DASD data sets and removes source tape data sets from the catalog after an aggregate has been successfully backed up.

Because these requirements are determined by the management class of the aggregate group, the Aggregate Group Application does not contain the fields that MVS/DFP contained for expiration date, destination, or toleration enqueue failure. If you are migrating from an MVS/DFP system, you should either print or display and record the values of the three deleted aggregate group fields before you IPL the system with DFSMS/MVS.

BSAM, QSAM, BPAM, and BDAM 31-Bit Addressing Support

31-bit addressing support enables programs to transfer larger amounts of data without running out of virtual storage. Programs running in 31-bit mode (AMODE 31) can now issue most BSAM, QSAM and BPAM macros. With AMODE 31, the data, program and save areas can all reside above the 16MB line. DCBs and

© Copyright IBM Corp. 1979, 1999

DECBs must remain below the 16MB line. The control block, DCBE, is introduced to contain optional user parameters to facilitate the 31-bit addressing mode.

The access method macros expand differently on this level of the system but they still work correctly on MVS/DFP systems when running in 24-bit mode. Programs can test the system level in the data facilities area control block (mapped by IHADFA), and switch to the mode that is appropriate for the system.

For more information, refer to *DFSMS/MVS Macro Instructions for Data Sets* and *DFSMS/MVS DFSMSdfp Advanced Services* .

Concurrent Copy

Concurrent copy allows DFSMSdss to backup or copy data sets without locking users out of the data set until the copy is complete. The backup or copy appears as if it were taken instantaneously.

You control the use of concurrent copy with the storage class attribute, ACCESSIBILITY, and the management class attributes, BACKUP COPY TECHNIQUE and ABACKUP COPY TECHNIQUE.

To use concurrent copy, your data must reside on a DASD volume attached through an IBM 3990 Cache Storage Control with the Extended Platform installed. DFSMSdss must be active on your system. For more information on concurrent copy, see the *DFSMS/MVS DFSMSdfp Storage Administration Reference*.

Data Set Performance Statistics

DFSMS/MVS provides statistics in SMF type 42 records that you can use to analyze data set I/O performance. Statistics at the data set level provide information such as the total number of I/O operations and the average I/O response time. These statistics are available for data sets of the following origins and access methods: EXCP, BSAM, QSAM, VSAM, DB2, DIV, and BPAM (except link list libraries).

Additional statistics are recorded for data sets eligible for cache. These statistics include the total number of I/O operations for data sets that are candidates for caching and the number of I/O operations that actually use cache for the data. These same statistics can also be analyzed at the SMS storage class level. Cache statistics are recorded only for data sets residing on DASD volumes connected to an IBM 3990 Storage Control with cache.

Refer to the *DFSMS/MVS DFSMSdfp Storage Administration Reference* for more information.

Device Console Service Improvements

Device console services are improved, allowing you to issue the DISPLAY SMS,OAM and DISPLAY SMS,OSMC operator commands from a TSO terminal in CONSOLE mode. Also, messages for Storage Management Subsystem operator commands that are issued by the system console operator are routed back only to the console on which the command was issued. Additional details are given in the *DFSMS/MVS DFSMSdfp Diagnosis Guide*.

Dynamic Cache Management Enhancements

DFSMS/MVS provides an enhancement to dynamic cache management that reduces the need for manual cache tuning. This enables more data sets to benefit from cache and nonvolatile storage.

An updated cache management algorithm optimizes the selection of data sets that are candidates for caching, improving performance when the cache of an IBM 3990 Storage Control becomes overloaded.

More information is provided in the *DFSMS/MVS DFSMSdfp Storage Administration Reference*and *DFSMS/MVS Implementing System-Managed Storage* .

Empty Data Set Support Enhancements

Support for empty data sets has been enhanced so that users will not receive I/O errors or encounter residual data if they attempt to read certain data sets before the new data has been written. This enhancement applies to all system-managed sequential data sets and all system-managed data sets with undefined data set organizations.

For additional information see *DFSMS/MVS Using Data Sets*.

Extended Format Sequential Data Sets

An extended format sequential data set can be defined and accessed with BSAM and QSAM. An extended format data set consists of *stripes* that reside on separate DASD volumes. A striped data set differs from a standard multivolume data set in which the data is written to one volume at a time, until all the space for that data set on the volume is filled. When data is written to a striped data set, the data is interleaved among the stripes. This technique allows the system to transfer data between DASD and virtual storage at a faster rate than an individual DASD volume can typically handle.

This performance enhancement is used mainly for large, physical sequential data sets that have high I/O activity. Support for extended format data sets requires an IBM 3990 Storage Control with Extended Platform, and ESCON channel attachment.

All extended format data sets must be system-managed. The data set striping characteristic is mostly transparent to an application program; however, there are changes in the SMS data class and storage class constructs that allow users to define extended format data sets.

All sequential data sets accessed through the BSAM and QSAM access methods are eligible for conversion to extended format data sets. To take advantage of striping, you need to define one or more data classes that specify extended format data sets and allow your ACS routines to assign the appropriate data class to data sets allocated with the striped format.

Applications do not need to be changed or recompiled to use striping; however, to improve performance, most users need to define additional buffers. Accumulation allows the system to process BSAM I/O requests more efficiently by not starting the

I/O until a minimum number of buffers are present. BSAM users can request accumulation with the data control block extension (DCBE MULTACC).

Refer to *DFSMS/MVS Using Data Sets* and *DFSMS/MVS DFSMSdfp Storage Administration Reference* for more information on processing extended format data sets.

PDSE Copy Performance Improvements

DFSMS/MVS improves the performance when PDSEs are copied; making it comparable to the performance of copying partitioned data sets with IEBCOPY or DFSMSdss.

Refer to *DFSMS/MVS Using Data Sets* and *DFSMS/MVS Utilities*for more information on copying PDSEs.

PDSE Cross-System Sharing Support

DFSMS/MVS allows users on different MVS/ESA systems to access the same partitioned data set extended (PDSE) simultaneously. Users on multiple MVS systems can concurrently create, read, or replace members of the PDSE.

Refer to “Sharing Access to a PDSE” on page 231 for information on setting up a PDSE sharing environment. PDSE sharing is also detailed in *DFSMS/MVS Using Data Sets*.

Program Management Enhancements

DFSMS/MVS provides the program management binder and program management loader for enhanced program management. The binder relaxes many of the limitations of the MVS/DFP linkage editor and loader, and supports an executable unit called a *program object*. The program object eliminates many of the restrictions of the traditional load module. The loader loads both program objects and traditional load modules.

Refer to *DFSMS/MVS Program Management* for more information.

Tape Library Support

Tape cartridges can now be system-managed in a tape library using SMS storage groups. System-managed tape allows users to define requirements for their tape data in logical, rather than physical, terms. Without requiring changes to programs or JCL, the storage administrator defines the policies the system uses to map those logical requirements to physical tape resources.

A tape library can consist of IBM 3490 or 3490E tape devices and the associated storage groups can contain IBM Cartridge System Tape or IBM Enhanced Capacity Cartridge System Tape. You can use the IBM 3495 Tape Library Dataserver to automate tape cartridge mounts and storage.

The hardware configuration definition (HCD) and the library attachment RPQ are prerequisites for tape library support. Additional planning information is outlined in

DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries and in the *3495 Tape Library Dataserver Installation Planning and Migration Guide*.

There are also special programming requirements for tape library support in a JES3 environment. JES3 installations must have MVS/ESA SP Version 4 Release 3, and JES3 Version 4 Release 2.1, with applicable PTFs. For information about the initialization and setup of library tape drives within a JES3 complex, and other operational considerations, see the *OS/390 MVS Initialization and Tuning Guide*.

Volume Mount Analyzer

The volume mount analyzer (VMA) is a special tool that uses SMF records to provide a comprehensive analysis of tape data. It also identifies suitable data set candidates for use with the tape mount management methodology.

See *DFSMS/MVS Using the Volume Mount Analyzer* for information on using this tool.

Planning to Use DFSMS/MVS Services

Setting up the Interactive Storage Management Facility 213
Storing ISMF Load Module Libraries 213
Library Placement 213
Library Access Control 214
Storing ISMF Distribution Libraries 214
Allocating ISPF Table Libraries 214
Making ISMF Available to TSO Users 215
Modifying the TSO Logon Procedure 215
Setting Up a CLIST 216
Modifying Menus to Invoke ISMF 217

Preparing for System-Managed Storage 219
Catalogs in a System-Managed Storage Environment 219
Indexed VTOCs for System-Managed Volumes 220
VSAM Volume Data Sets for System-Managed Volumes 220
Virtual Storage Requirements for the Storage Management Subsystem . . . 220
Steps for Developing an Implementation Plan 220

Evaluating Program and Data Management Functions 225
Planning for PDSEs 225
Understanding the Structure of a PDSE 226
Directory Structure 226
Record Structure 227
Library Type 227
Defining PDSEs 227
PDSE Storage Requirements 228
Allocating Space for a PDSE 229
Migrating Data From Partitioned Data Sets to PDSEs 230
Other PDSE Conversion Issues 231
Sharing Access to a PDSE 231
PDSE Processing Restrictions 232
Planning to Access Program Objects 234
Processing Restrictions for Program Objects 234
Planning for Extended Format Data Sets 234
Planning for Hierarchical File System Data Sets 235
Creating Hierarchical File System Data Sets 236
Defining the Root File System 236
Creating and Mounting the Root File System 236
Creating Additional File Systems and Directories 237
Adding and Mounting File Systems to the Root File System 237
Managing File System Size 238
Managing File System Activity 238
Transporting a File System 238
Removing (Deleting) a File System 239
Migrating a File System 239
Backing Up File Systems 239
Recovering a Backed-Up File System 239

Planning for Tape in a System-Managed Environment 241
Optimizing Your Current Tape Environment 241

© Copyright IBM Corp. 1979, 1999

Analyzing Tape Usage with the Volume Mount Analyzer 242
Converting Tape Volumes to System Management 243
Conversion Activities 244

Preparing to Implement DFSMSrmm 247
Evaluating Removable Media Management Needs 247
Assessing DASD Space Requirements 248
DASD Space for the DFSMSrmm Control Data Set 248
Calculating DASD Space for the DFSMSrmm Control Data Set 248
Placing the DFSMSrmm Control Data Set 249
DASD Space for the Journal 250
Calculating DASD Space for the Journal 250
Placing the Journal 251
DASD Space for the Problem Determination Aid Log Data Sets 251
Planning DFSMSrmm Authorization and Security 251
Deciding How to Run DFSMShsm with DFSMSrmm 252
Planning for Operating DFSMSrmm 252
Planning for DFSMSrmm Customization 253
Planning to Use DFSMSrmm Utilities 254
Planning Dependencies on Open, Close, End-of-Volume 255

Preparing to Implement DFSMShsm 257
Developing a DFSMShsm Implementation Plan 257
Planning the Processor Environment 257
Planning for Data Security 258
Preparing DFSMShsm Data Sets and Volumes 258
DFSMShsm Data Sets 258
Control Data Sets 259
Journal Data Sets 259
Log Data Sets 259
Control Data Set Availability 260
DFSMShsm-Owned Volumes 260
Planning for DFSMShsm-Managed Data Sets and Volumes 261
Setting Availability Management Policies 261
On-Site Data Set Recoverability 261
Creating Backup Cycles 262
Establishing the Backup Window 262
Setting the Number of Backup Tasks 262
Selecting Backup Parameters 262
Choosing Backup Media 263
Planning for Data Set Recovery 263
Planning for Backup Volume Cleanup 263
On-Site Volume Recoverability 263
Volume Dump Cycle 263
Volume Dump Window 264
Number of Dump Tasks 264
Dump Classes 264
Planning Recovery for DFSMShsm-Owned Volumes 264
Off-Site Recoverability 265
Recovering Critical Applications with ABARS 265
Recovering Noncritical Applications with ABARS 265
Planning for Disaster Recovery without ABARS 265
Setting Space Management Policies 266
Management Class Migration and Expiration Parameters 266

Automatic Primary Space Management Window 266
Automatic Primary Space Management Cycle 267
Number of Automatic Volume Space Management Tasks 267
Interval Migration 267
Automatic Secondary Space Management Window 267
Automatic Secondary Space Management Cycle 267
Migration Level 1 Volumes 268
Setting Thresholds 268
Migration Level 2 Media 268
Small Data Set Packing 268
Backing Up SDSP Data Sets 269
Reorganizing SDSP Data Sets 270
Recovering SDSP Data Sets 270
Pooling Volumes for Non-SMS-Managed Data Sets 270
Pools in Systems Running under JES2 270
Pools in Systems Running under JES3 271
Setting Up Tape Management for DFSMShsm 271
Scratch Pool Environment 273
Output Tape Selection 273
Recycling Tapes 273
Device Selection 274
Tape Operations 274
Fully unattended operation 274
Partially unattended operation 274
Testing DFSMShsm 275
Customizing and Tuning DFSMShsm 276
Coordinating DFSMShsm Processing 277
Sample Plans 279
Sample Setup Plan 279
Sample Plan for Allocating DFSMShsm Data Sets 280
Sample Plan for Maintaining Control Data Sets 281
Sample Plan for Availability Management 282
Sample Plan for Space Management 283
Sample Plan for Customizing and Tuning DFSMShsm 285

Securing DFSMS/MVS Resources 287
Using RACF Profiles 287
Protecting ISMF Functions 288
Protecting Data Management Functions 288
Protecting VSAM Record-level Sharing Functions 288
Protecting DFSMSrmm Functions 288
Protecting DFSMSdss Functions 289
Data and Volume Access 289
DFSMSdss Commands and Keywords 289
Protecting DFSMShsm Functions 289
Data Set Access 289
Tape Access 289
DFSMShsm Commands 290

Setting up the Interactive Storage Management Facility

The Interactive Storage Management Facility (ISMF) helps you manage data and storage interactively. It uses functions provided by DFSMS/MVS, RACF, and DFSORT to perform data and storage management tasks on the system. As an Interactive System Productivity Facility (ISPF) application, ISMF is menu-driven with fast-path access to many of its storage management functions.

You install ISMF as part of the DFSMS/MVS product. In conjunction with the installation procedures outlined in the *DFSMS/MVS Program Directory*, you need to complete the following steps to set up ISMF on your system:

1. Decide where to place the ISMF load module libraries and distribution libraries.
2. Make ISMF available to TSO users, either through a CLIST or JCL logon procedure.
3. Modify one or more menus to allow users to invoke ISMF.

These steps are described in this chapter. Once installed, ISMF can be customized to meet the needs of your storage administrator and users. Authorization for ISMF commands and other operations can be restricted with the RACF program control feature. For more information on tailoring ISMF to your needs, refer to the *DFSMS/MVS DFSMSdfp Storage Administration Reference* and *DFSMS/MVS Using ISMF*.

Storing ISMF Load Module Libraries

ISMF load module libraries are used to set up ISMF during the DFSMS/MVS installation. When following installation procedures in the *DFSMS/MVS Program Directory*, you are given several options for where to store your ISMF load module libraries. In deciding the placement of your libraries, you must consider the performance and access requirements of these load modules.

Library Placement

The placement of your load modules affects the performance of your system. ISMF load modules should be stored on high-performance DASD.

The default location for ISMF load modules is SYS1.DGTLLIB. If you want to use RACF to control access to ISMF functions, consider placing the load modules into the system link library, SYS1.LINKLIB, or into a private load library. If your ISMF load modules are placed anywhere other than SYS1.DGTLLIB, you must identify the location of these modules to SMP/E.

If you make the load modules a part of SYS1.LINKLIB, you must do *one* of the following:

- Copy the load modules from SYS1.DGTLLIB to SYS1.LINKLIB.
- Add the name SYS1.DGTLLIB to LNKLSTnn, which is a member of SYS1.PARMLIB.
- Copy the load modules from SYS1.DGTLLIB to an existing data set whose name is already in LNKLSTnn.

© Copyright IBM Corp. 1979, 1999

Another method for identifying the ISMF load modules to SMP/E is to leave the load modules in SYS1.DGTLLIB and add the data set name to the TSO logon JCL or to a new or existing CLIST for use with TSO. (See “Making ISMF Available to TSO Users” on page 215.)

If you are not using SYS1.DGTLLIB at execution time, and if you have backed up and copied all the load modules, you can delete the SYS1.DGTLLIB data set after installation.

Space requirements for the ISMF libraries are documented in the *DFSMS/MVS Program Directory*. Additional placement considerations are described in *OS/390 MVS System Data Set Definition* and *DFSMS/MVS DFSMShsm Implementation and Customization Guide*.

Library Access Control

You can control access to ISMF functions using the RACF program control feature. To use this feature, you will need to place your ISMF load modules into SYS1.LINKLIB.

The RACF program control feature can protect the entire ISMF component, specific applications, or any of the individual ISMF line operators or commands. You decide who is authorized to perform each ISMF function, setting up an access scheme that applies to either individual users or groups of users.

In addition to controlling ISMF functions, you can use standard RACF checking to limit access to individual data sets, volumes, or catalogs. Used in conjunction with program control, access checking ensures that appropriate ISMF lists and functions are available to users when they are needed.

For procedures on how to restrict ISMF functions, see *DFSMS/MVS DFSMSdfp Storage Administration Reference* . For more information on RACF, see the *OS/390 Security Server (RACF) Security Administrator's Guide*.

Storing ISMF Distribution Libraries

ISMF distribution libraries are used mainly for ISMF maintenance activities. Before you decide where to place your distribution libraries, consider performance, recovery, and cost. Because these libraries are referred to infrequently, their placement is not important for performance. However, it is important that they have long-term integrity because they are used for backup procedures. For information on space requirements of the distribution libraries, refer to *DFSMS/MVS Program Directory*.

Allocating ISPF Table Libraries

Each ISMF user needs a personal partitioned data set (PDS) to use as a table output library. The ISMF SAVE command stores the saved list in this PDS using ISPF table services. Allocate this data set using a DDNAME of ISPTABL; only one PDS can be used in this allocation because concatenated partitioned data sets are not allowed for output. The data set name you choose might be of this form:

```
DSN=userid.ISMF.DGTTABL
```

Place this data set name in the concatenation for the ISPF table input library (ISPTLIB) also, so that the output of the SAVE command is available for subsequent use.

An example of the JCL DD statement for the table output library:

```
//ISPTABL  DD DSN=userid.ISMF.DGTTABL,DISP=OLD
```

An example of the statements for the table input libraries:

```
//ISPTLIB  DD DSN=userid.ISMF.DGTTABL,DISP=SHR
//         DD DSN=SYS1.DGTTLIB,DISP=SHR
```

The table output library, *.ISMF.DGTTABL, has a fixed-block record format and a record length of 80. The block size must be a multiple of 80 and be compatible with the other data sets in the concatenation.

Making ISMF Available to TSO Users

To provide your TSO users with access to the load module library, set up the execution environment by modifying your TSO logon procedure, or by writing a CLIST. Both procedures are described in the following sections. The advantage of using a CLIST over a logon procedure is that the logon procedure is performed only at TSO logon time, whereas different CLISTS can be used to change data set access schemes many times within the same session.

Modifying the TSO Logon Procedure

You can provide TSO users with access to a load module data set by adding DD statements to your logon procedure. Add the load module data set for ISMF to the existing DD statements for the ISPF libraries. You must specify the load module data set before the ISPF data set names in the concatenation. For an example of a concatenation including the ISMF load module data set, see Figure 3 on page 216.

You can include other data sets as part of these concatenations.

```
//ISPLLIB   DD DSN=SYS1.DFQLLIB,DISP=SHR
//          DD DSN=SYS1.DGTLLIB,DISP=SHR
//          DD DSN=SYS1.EDGMOD1,DISP=SHR
//          DD DSN=SYS1.SISPLPA,DISP=SHR
//          DD DSN=SYS1.SISPLOAD,DISP=SHR
//ISPPLIB   DD DSN=SYS1.DFQPLIB,DISP=SHR
//          DD DSN=SYS1.DGTPLIB,DISP=SHR
//          DD DSN=SYS1.EDGPENU,DISP=SHR
//          DD DSN=SYS1.SISPPENU,DISP=SHR
//ISPMLIB   DD DSN=SYS1.DFQMLIB,DISP=SHR
//          DD DSN=SYS1.DGTMLIB,DISP=SHR
//          DD DSN=SYS1.EDGMENU,DISP=SHR
//          DD DSN=SYS1.SISPMENU,DISP=SHR
//ISPSLIB   DD DSN=SYS1.DGTSLIB,DISP=SHR
//          DD DSN=SYS1.EDGSKL1,DISP=SHR
//          DD DSN=SYS1.SISPSENU,DISP=SHR
//ISPTLIB   DD DSN=SYS1.DGTTLIB,DISP=SHR
//          DD DSN=SYS1.EDGTBL1,DISP=SHR
//          DD DSN=SYS1.SISPTENU,DISP=SHR
//SYSPROC   DD DSN=SYS1.DGTCLIB,DISP=SHR
//          DD DSN=SYS1.EDGEXE1,DISP=SHR
//          DD DSN=SYS1.SISPCLIB,DISP=SHR
```

Figure 3. Example of TSO Logon Procedure Specifying ISMF Load Module Data Sets

Setting Up a CLIST

Write a CLIST as an alternative or supplement to modifying JCL logon procedures. The CLIST must consist of a series of ALLOCATE commands for each ISMF and ISPF library. Add the name of the library containing the CLIST to either the concatenation of SYSPROC or the ddname ISPCLIB. Figure 4 on page 217 shows the ALLOCATE statements for each of the libraries.

You can include other data sets as part of these concatenations.

```
ALLOCATE FILE(ISPLLIB) SHARE REUSE     -
          DATASET('SYS1.DFQLLIB'       -
                  'SYS1.DGTLLIB'       -
                  'SYS1.EDGMOD1'       -
                  'SYS1.SISPLPA'       -
                  'SYS1.SISPLOAD')
ALLOCATE FILE(ISPPLIB) SHARE REUSE     -
          DATASET('SYS1.DFQPLIB'       -
                  'SYS1.DGTPLIB'       -
                  'SYS1.EDGPENU'       -
                  'SYS1.SISPPENU')
ALLOCATE FILE(ISPMLIB) SHARE REUSE     -
          DATASET('SYS1.DFQMLIB'       -
                  'SYS1.DGTMLIB'       -
                  'SYS1.EDGMENU'       -
                  'SYS1.SISPMENU')
ALLOCATE FILE(ISPSLIB) SHARE REUSE     -
          DATASET('SYS1.DGTSLIB'       -
                  'SYS1.EDGSKL1'       -
                  'SYS1.SISPSENU')
ALLOCATE FILE(ISPTLIB) SHARE REUSE     -
          DATASET('SYS1.DGTTLIB'       -
                  'SYS1.EDGTBL1'       -
                  'SYS1.SISPTENU')
ALLOCATE FILE(SYSPROC) SHARE REUSE     -
          DATASET('SYS1.DGTCLIB'       -
                  'SYS1.EDGEXE1'       -
                  'SYS1.SISPCLIB')
```

Figure 4. Example of a CLIST That Makes Load Module Data Sets Available to TSO Users

Modifying Menus to Invoke ISMF

To allow users to invoke ISMF, you need to add an option on one of the following interactive menus:

- The ISPF Primary Option Menu
- The SPF Master Application Menu
- A menu of your own choosing

An example for creating an ISMF option on the ISPF Primary Option Menu is shown here. This example is based on an ISPF 4.2 system and assumes that ISR@PRIM, the member containing ISPF primary option menu and processing instructions, is already copied into SYS1.DGTPLIB.

1. Edit SYS1.DGTPLIB(ISR@PRIM) to add entries to the ')AREA SAREA39' and ')PROC' sections of the source code.
2. The ')AREA SAREA39' section lists the options the users see on the ISPF panel. Insert the option letter and text for invoking ISMF, as shown in the partial 'AREA SAREA39' section below.

```
)AREA SAREA39
 0  Settings      Terminal and user parameters
 1  View          Display source data or listings
 2  Edit          Create or change source data
 3  Utilities     Perform utility functions
 .
 .
 .
 10 SCLM          SW Configuration Library Manager
 I  ISMF          Invoke ISMF
```

3. The ')PROC' section contains the code to execute the ISPF options. Insert the code to invoke ISMF, as shown in the partial ')PROC' section below.

```
)PROC
  ...
&ZSEL = TRANS ( TRUNC (&ZCMD,'.')
  0,'PGM(ISPISM) SCRNAME(SETTINGS)'
  1,'PGM(ISRBRO) PARM(ISRBRO01) SCRNAME(VIEW)'
  2,'PGM(ISREDIT) PARM(P,ISREDM01) SCRNAME(EDIT)'
  3,'PANEL(ISRUTIL) SCRNAME(UTIL)'
  .
  .
  .
  10,'PGM(ISRSCLM) SCRNAME(SCLM) NOCHECK'
   I,'PGM(DGTFMD01) PARM(&ZCMD) NEWAPPL(DGT) NOCHECK'
```

4. Save your changes in ISR@PRIM.
5. Verify that your modified ISPF Primary Option Menu looks similar to the one shown in Figure 5. When you select option “I” from this panel, the ISMF Primary Option Menu should display.

```
  Menu  Utilities  Compilers  Options  Status  Help
 ------------------------------------------------------------------------------
                           ISPF Primary Option Menu
 Option ===>

 0  Settings      Terminal and user parameters           User ID . : L841557
 1  View          Display source data or listings        Time. . . : 12:23
 2  Edit          Create or change source data           Terminal. : 3278
 3  Utilities     Perform utility functions              Screen. . : 1
     .
     .
     .
10  SCLM        - SW Configuration Library Manager
 I  ISMF        - Invoke ISMF
     .
     .
     .


 Enter X to Terminate using log/list defaults
```

Figure 5. Example of a modified ISPF Primary Option Menu

Preparing for System-Managed Storage

This chapter provides information to help you begin building your Storage Management Subsystem (SMS). By creating an environment in which storage is managed by the system and controlled by a storage administrator, SMS can help you improve data availability, performance, space utilization, and device installation in your data center.

The following books contain more detailed information to help you implement system-managed storage.

- *DFSMS/MVS Implementing System-Managed Storage* provides information about migrating to, and using SMS as part of your storage management strategy.
- *DFSMS/MVS General Information* provides a high-level overview of SMS components.
- *DFSMS/MVS DFSMSdfp Storage Administration Reference* describes the components and functions of SMS and shows you how to use ISMF to implement a configuration.
- *DFSMS/MVS Using ISMF* describes the storage management tasks an end user can perform under SMS.
- *DFSMS/MVS Managing Catalogs* addresses cataloging in the SMS environment.
- *DFSMS/MVS Using Data Sets* addresses SMS for data sets.
- *DFSMS/MVS NaviQuest User's Guide* helps you understand and implement NaviQuest through the use of ISMF interface and help panels.

Catalogs in a System-Managed Storage Environment

All SMS-managed data sets must be cataloged in integrated catalog facility catalogs. Data sets cataloged in CVOLs or VSAM catalogs cannot be managed by SMS. If your current catalog environment is not positioned for SMS (for example, you have CVOLs or VSAM catalogs), migration to SMS can take longer.

Implementing a few simple guidelines can help you prepare your catalog environment for SMS:

- Discourage the use of JOBCAT and STEPCAT statements because they are not allowed in jobs that reference SMS-managed data sets.
- Protect catalogs of SMS-managed data sets with RACF rather than through password protection. Passwords are ignored for SMS-managed data sets.
- Centralize control of your installation's configuration of catalogs.

For the details of converting CVOLs and VSAM catalogs to integrated catalogs, and information pertinent to integrated catalog facility catalogs, see *DFSMS/MVS Managing Catalogs*.

© Copyright IBM Corp. 1979, 1999

Indexed VTOCs for System-Managed Volumes

An indexed VTOC is a combination of a volume table of contents and an index for direct access into that VTOC. The SMS environment requires indexed VTOCs for all DASD volumes that are SMS-managed.

You can find further information about the VTOC and its index in *DFSMS/MVS DFSMSdfp Advanced Services* and in the *ICKDSF R16 User's Guide*.

VSAM Volume Data Sets for System-Managed Volumes

There is a VSAM Volume Data Set (VVDS) on every SMS-managed volume.

All ICF cataloged VSAM data sets, whether SMS or non-SMS, have entries in the VVDS. The entries contain information about each component of the VSAM cluster on that volume. A VVDS on a SMS-managed volume contains entries for VSAM as well as non-VSAM data sets. The entries for non-VSAM data sets include the SMS storage, management, and data class names.

A VVDS is dynamically allocated by the system, if one does not exist on the volume, when the first data set is allocated on the volume.

Virtual Storage Requirements for the Storage Management Subsystem

Jobs that use system-managed data sets require more virtual storage space than equivalent jobs using non-system-managed data sets due to the additional control information the system requires.

If users are experiencing virtual storage abends, for example, abends 878 and 80A, try one of the following to circumvent the abends:

- Have the user *decrease* the region size requested for the job. This makes more room available below 16MB for required system control blocks.
- Split the job into two or more jobs.
- For JES2 systems, create a job class that places the scheduler work area above 16MB, and have affected users specify this class in the CLASS parameter on their JOB statement. After picking a class, use the JOBCLASS statement with the SWA=ABOVE keyword to create the class in the JES2 initialization stream. See *OS/390 JES2 Initialization and Tuning Reference* for information about the JES2 initialization stream.
- For JES3 installations, use the CIPARM statement in the JES3 initialization stream to specify that the scheduler work area should be placed above 16MB. See *OS/390 JES3 Initialization and Tuning Reference*for information about the JES3 initialization stream.

Steps for Developing an Implementation Plan

Converting to system-managed storage is often a big step and should be given the same planning consideration as other major projects. Spend time first planning and developing your SMS configuration to achieve your storage management goals. A starter set (a simple model configuration) is provided with DFSMS/MVS to help you begin this process.

Just as storage management needs differ from site to site, planning needs also vary. Some installations moving to a system-managed storage environment are in a position to exploit the full potential of SMS—for example, their storage is already pooled and their data is centrally-managed. For other installations, the move to administrator-controlled storage is more gradual, as new storage management policies evolve.

The tasks in the chart beginning on page 221 can help you prepare for SMS implementation. Consider each item when you develop your own implementation plan. Books that give you additional planning information for each task or subtask are referenced in this chart.

A sample DFSMS implementation project plan is available in the *DFSMS/MVS Implementing System-Managed Storage*, giving you detailed steps for converting to a system-managed environment. You can bypass some of these steps by using the Fast Implementation Technique to get most of your data onto SMS in a short period of time with minumum effort.

Table 7 (Page 1 of 4). SMS Implementation Planning Tasks and References

Planning Tasks	Subtasks	References
Understand product requirements and system considerations	Determine appropriate product prerequisites and corequisites	*DFSMS/MVS General Information*
	Understand multisystem considerations for SMS and non-SMS coexistence	*DFSMS/MVS Implementing System-Managed Storage*
	Understand JES3 considerations	*OS/390 JES3 Introduction*
Establish implementation responsibilities	Establish project team responsibilities for: Storage administration Software and hardware planning User group interaction	*MVS/ESA SML: Leading a Storage Administration Group*
	Determine prerequisites for migration (for example, establishing data set naming standards or other policies).	*DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Data* *MVS/ESA SML: Managing Storage Groups*
Determine education needs for users	Provide instruction on how to use the services of SMS	*DFSMS/MVS General Information* *DFSMS/MVS Implementing System-Managed Storage*
	Inform users of prerequisite changes, if any	*DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Data*
Determine data set requirements	Determine which data sets will be SMS-managed and when	*DFSMS/MVS DFSMSdfp Storage Administration Reference* *DFSMS/MVS Implementing System-Managed Storage*
	Understand user data requirements for performance, availability, and security	*DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Data* *MVS/ESA SML: Managing Storage Groups*

Table 7 (Page 2 of 4). SMS Implementation Planning Tasks and References

Planning Tasks	Subtasks	References
Determine data set requirements (continued)	Determine catalog requirements	*DFSMS/MVS Managing Catalogs* *DFSMS/MVS Using Data Sets*
	Determine which partitioned data sets to convert to PDSEs	*DFSMS/MVS Using Data Sets* *MVS/ESA SML: Managing Data* *DFSMS/MVS Program Management*
	Analyze use of passwords (SMS ignores passwords for data sets) and consider protecting the data with RACF	*MVS/ESA SML: Managing Data*
	Evaluate JCL issues Although no changes are required for SMS conversion, plan for strategic changes (for example, eliminating the use of the UNIT parameter)	*OS/390 MVS JCL User's Guide* *DFSMS/MVS Implementing System-Managed Storage*
	Plan for coexistence of SMS and non-SMS data	*DFSMS/MVS Implementing System-Managed Storage*
Set up storage groups, aggregate groups, storage classes, management classes, and data classes	Determine the volumes to be contained in each storage group Consider pooling requirements	*DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Storage Groups*
	Determine how many aggregate groups will be required Evaluate which data sets to back up and recover for disaster recovery and determine a method of recovery	*DFSMS/MVS DFSMSdfp Storage Administration Reference*
	Determine storage classes Consider performance and availability requirements	*DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Data* *MVS/ESA SML: Managing Storage Groups* *DFSMS/MVS DFSMSdfp Storage Administration Reference*
	Determine management classes Consider space management needs and requirements for backup/recovery	*DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Data* *MVS/ESA SML: Managing Storage Groups* *DFSMS/MVS DFSMSdfp Storage Administration Reference*
	Define data classes Consider space and service requirements	*DFSMS/MVS DFSMSdfp Storage Administration Reference*

Table 7 (Page 3 of 4). SMS Implementation Planning Tasks and References

Planning Tasks	Subtasks	References
Design Automatic Class Selection (ACS) routines	Determine which system input variables to use as filter criteria to assign storage groups, storage classes, management classes	*DFSMS/MVS DFSMSdfp Storage Administration Reference* *DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Data*
	Determine requirements for user specification of management class, storage class, and data class	*DFSMS/MVS DFSMSdfp Storage Administration Reference* *DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Data*
	Understand how to assign RACF authorization to management classes, storage classes	*DFSMS/MVS DFSMSdfp Storage Administration Reference* *OS/390 Security Server (RACF) Introduction*
	Implement concurrent copy for backup and copy use, deciding which data sets to allocate to storage classes that ensure placement on storage controllers that support concurrent copy of snap shot.	*DFSMS/MVS DFSMSdfp Storage Administration Reference*
Set up recovery procedures for SMS conversion	Plan for recovery of SMS control data sets	*DFSMS/MVS DFSMSdfp Storage Administration Reference*
	Plan for recovery of SMS-managed data	*DFSMS/MVS DFSMSdfp Storage Administration Reference* *DFSMS/MVS Implementing System-Managed Storage* *MVS/ESA SML: Managing Data*
Implement the planned configuration	Use ISMF to define storage groups, storage classes, management classes, data classes, and aggregate groups	*DFSMS/MVS DFSMSdfp Storage Administration Reference*
	Write and translate ACS routines using the ISMF ACS application	*DFSMS/MVS DFSMSdfp Storage Administration Reference*
	After validating and testing the configuration, activate a Source Control Data Set (SCDS) through ISMF or with a SETSMS operator command	*DFSMS/MVS DFSMSdfp Storage Administration Reference*
	Use NaviQuest to do ACS testing in batch processing mode	*DFSMS/MVS NaviQuest User's Guide*
Migrate data to SMS-management	Use DFSMSdss in-place conversion	*DFSMS/MVS DFSMSdss Storage Administration Guide*
	Use DFSMSdss data set copy	*DFSMS/MVS DFSMSdss Storage Administration Guide*
	Use ICKDSF	*ICKDSF R16 User's Guide*
	Use DFSMShsm migrate, recall	*DFSMS/MVS DFSMShsm Managing Your Own Data*

Table 7 (Page 4 of 4). SMS Implementation Planning Tasks and References

Planning Tasks	Subtasks	References
Maintain and monitor SMS	Monitor space usage through ISMF	*DFSMS/MVS DFSMSdfp Storage Administration Reference*
	Evaluate performance	*DFSMS/MVS Implementing System-Managed Storage*

Evaluating Program and Data Management Functions

This chapter highlights the newer program and data management functions available in DFSMS/MVS. It provides planning details to consider as you replace a partitioned data set (PDS) with a partitioned data set extended (PDSE). It also gives an overview of planning considerations for hierarchical file system (HFS) data sets, and describes the advantages of extended format data sets, program objects, and 31-bit addressing in the DFSMS/MVS environment.

Planning for PDSEs

The PDSE expands the capabilities of system-managed storage to help you improve system performance and productivity.

The main advantage of using a PDSE over a partitioned data set is that the PDSE uses DASD space more efficiently. The size of a partitioned data set directory is fixed, regardless of the number of members in it, while the size of a PDSE directory is flexible and expands to fit the stored members. Also, the system reclaims space automatically whenever a member is deleted or replaced, and returns it to the pool of space available for allocation to other members of the same PDSE. The space can be reused without compressing the data set.

Other advantages of PDSEs:

- The system requires less time to search directories. The PDSE directory is indexed and is searched using that index. The directory can also be cached in virtual storage.
- You can create multiple members at the same time.
- Sharing occurs at both member and data set levels. This makes it easier to maintain the integrity of the PDSE when modifying separate members at the same time.
- PDSEs can expand up to 123 extents per volume. An extent is a continuous area of space on a DASD storage volume, occupied by, or reserved for, a specific data set.
- Program objects stored in PDSE libraries can be data in virtual (DIV) mapped into storage. This makes it possible for the program management loader to load only part of a program into storage before passing control to it.
- PDSE program libraries allow more control information to be stored in the directory than partitioned data sets do.
- PDSE program libraries offer superior buffer management and caching capability.
- Management of alias entries in PDSE libraries is better than that offered by partitioned data set libraries. There is no possibility that aliases can point to members that have been deleted or replaced.
- PDSEs can be shared by all the processors in a multisystem complex. A user sharing a PDSE can read existing members and create new members or new copies of existing members while other users on the same or other processors are also sharing the PDSE. When a member is being updated, sharing is restricted to one processor.

© Copyright IBM Corp. 1979, 1999

PDSEs and partitioned data sets share some important similarities. They are processed using the same macros and access methods (BSAM, QSAM, BPAM); and, records are stored in members that are described in a directory. Significant differences between PDSEs and partitioned data sets are listed in Table 8.

Table 8. Differences Between PDSEs and Partitioned Data Sets

PDSE Characteristics	Partitioned Data Set Characteristics
Data set has a 123-extent limit	Data set has a 16-extent limit
Directory is open-ended and indexed by member name; faster to search directory	Fixed-size directory searched sequentially
Records are logically reblockable in a device-independent manner. Track record address (TTR) simulated as a system key.	Device-dependent TTR addressing and block size
Dynamic space allocation and reclaim	Must compress data set to reclaim space
Data sets and members can be shared	Only data sets can be shared
Create multiple members at the same time	Create one member at a time

To fully exploit the potential of PDSEs and smooth the transition from partitioned data sets, you need to know how to:

Define a PDSE structure
Read a PDSE directory
Concatenate PDSEs and partitioned data sets
Place programs into PDSEs
Retrieve, modify, and process a PDSE member
Use the SYNCDEV macro to synchronize data to DASD

These topics are introduced here and discussed further in *DFSMS/MVS Using Data Sets* and in *DFSMS/MVS Program Management*.

Understanding the Structure of a PDSE

You can use a PDSE in place of a partitioned data set to store data. You can also use a PDSE to store executable programs, known as program objects. Only the program management binder can create program objects.

Externally, a PDSE appears similar to a partitioned data set; most partitioned data set and PDSE interfaces are indistinguishable. A PDSE is a system-managed data set divided into multiple members, each described by one or more directory entries. Member and directory data is stored on a device in blocks. Each member has a unique name, 1 to 8 characters long, and can have several aliases. Member records of a PDSE are written or retrieved sequentially.

Directory Structure

You can use BSAM or QSAM to read a PDSE directory sequentially. The directory is searched and maintained by the BLDL, FIND, and STOW macros.

When accessed sequentially, the PDSE directory appears to be constructed of 256-byte blocks containing sequentially ordered entries. (The PDSE directory is presented like a partitioned data set directory even though its internal structure is different.) Directory entries are arranged by name in alphanumeric collating sequence.

The PDSE directory is also indexed, permitting faster searches for members. The member name and starting location, or track/track/record (TTR), are used as search keys. For PDSEs, the TTR is a token that simulates a track or record location, but does not represent an actual track or record location. TTRs in the directory can change if you move the PDSE.

The directory is expandable—you can keep adding entries up to the directory's limit (522,236 members) or until the data set runs out of space. The system acquires the space it needs for the directory entries from the storage available to the data set.

For a partitioned data set, the directory size is determined when the data set is allocated. There can be fewer members in the data set than the directory can contain, but when the preallocated directory space is full, you must copy the partitioned data set to a new data set before new members can be added.

Note: Directories for PDSEs containing large program objects are not completely compatible with the format of partitioned data set directories.

Record Structure

PDSE and partitioned data set members are accessed sequentially. Each data member can contain a maximum of 15,728,639 logical records. PDSE members are reblockable. Logical records are written to DASD in physical blocks and can span physical records. The user- or system-defined block size is saved in the data set control block (DSCB) when the data is written, and becomes the default block size for input. The system reconstructs the original user- or system-defined block size when reading the PDSE members. These reconstructed blocks are called *simulated blocks*. There is a logical end-of-file mark at the end of each PDSE member.

You can either calculate the block size yourself and specify it in the BLKSIZE parameter of the DCB or let the system determine it for you. The block size makes no difference in space usage for PDSEs, but it does affect the buffer size of a job using the PDSE.

Library Type

There are two types of PDSE libraries: data libraries and program libraries. The first member stowed determines the library type. Once the library type has been determined, only members of the same type can be stored in that library.

The major differences between data libraries and program libraries are:

- Program libraries are always defined with DCB attributes RECFM=U and BLKSIZE greater than or equal to 4096. A default size of 4096 is used if the BLKSIZE value specified is less than 4096.
- Members of program libraries can only be created by the program management binder or IEBCOPY. BSAM support is limited to READ only, and STOW ADD or REPLACE is not allowed.

Defining PDSEs

If you install DFSMS/MVS while system-managed storage is active, you can allocate PDSEs in system-managed storage groups. To define PDSE data set types, specify DSNTYPE=LIBRARY in a data class definition, in a JCL DD statement, in the LIKE keyword, or in a TSO/E ALLOCATE command. Your storage administrator can also assign DSNTYPE=LIBRARY as an installation default value. This default value is assigned in the SYS1.PARMLIB member IGDSMSxx.

Table 9 shows how SMS checks DSNTYPE in the JCL, the LIKE keyword in the DATACLAS, and the installation default to determine whether to allocate a data set as a PDSE or a partitioned data set. If no DSNTYPE is found, or if a DSNTYPE of PDS is processed first, the resulting data set type is a partitioned data set.

Table 9. Defining a PDSE using the DSNTYPE keyword

DSNTYPE in JCL or TSO/E	DSNTYPE in DATACLAS	Installation Default	Resulting Data Set Type
none	none	LIBRARY	PDSE
none	LIBRARY	not used	PDSE
LIBRARY	not used	not used	PDSE
LIBRARY	partitioned data set	not used	PDSE

PDSE Storage Requirements

When planning for PDSE processing, you should be aware of the following storage requirements:

Directory Requirements

The storage used for PDSE directory information is obtained from storage that is available to the data set. Because the directory expands dynamically, storage for the directory is obtained as needed. Directory storage is not limited to the primary extent, but can be obtained from any available storage.

Information (such as attributes, statistics, and status) is retained in the directory for each PDSE member. The directory storage required to support any single member is variable, as is the storage required to support alias names.

The minimum directory size is 5 pages, even if there are no members in the PDSE. For a medium-sized PDSE consisting of 150 members, approximately 12 pages of directory storage are required.

Deleting a PDSE member can, in some cases, increase the amount of directory space used by the remaining members.

Member Data Requirements

Program objects residing in PDSE program libraries could require more space than equivalent load modules in partitioned data set libraries. This is because initialized data areas must be accounted for in program objects. Some program objects can also be smaller than their load module counterparts, because a consistent block size of 4096 is used for all records.

If expanded storage is available, and the PDSE storage class is defined with a 3 millisecond response time, PDSE member data is stored in an expanded storage-only Hiperspace or in processor cache storage.

Data Area and Other Support Requirements

PDSE support builds some data areas and other work areas in ECSA and ESQA during IPL. Storage requirements increase as the number of PDSE members grows, or as the number of users sharing PDSE members increases.

Table 10 shows the minimum storage usage after IPL and examples of how this storage grows after PDSE processing.

Table 10. Storage requirements for PDSE support.

System Storage Area	Minimum Requirements	Increase for PDSE Source Data Sets (Example 1)	Increase for PDSE Program Libraries (Example 2)
ECSA	1.7MB	3.4MB	2.0MB
ESQA	0.9MB	1.2MB	2.6MB

Notes:

1. Example 1: When copying a 1000–member PDS to a PDSE, storage usage increases mostly in ECSA, as shown. Based on this example, approximately 1.7MB of ECSA and 0.3MB of ESQA is required per 1000 PDSE members connected.
2. Example 2: When copying a 1000–member program library, most of the virtual storage growth is in ESQA. Based on this example, approximately 0.3MB of ECSA and 1.7MB of ESQA is required per 1000 program objects connected.

As the examples in the table above demonstrate, a 1000-member source PDSE requires approximately 1.7MB storage in ECSA and 0.3MB in ESQA and a 1000-member PDSE program library requires 0.3MB of ECSA and 1.7MB of ESQA. This is in addition to the minimum 1.7MB of ECSA and 0.9MB of ESQA acquired during IPL. PDSEs containing more members take correspondingly more ECSA and ESQA storage.

Based on PDSE usage in your installation, you need to set a sufficiently high limit on ECSA and ESQA usage. When the ECSA or ESQA that is allowed by your installation is exhausted, additional storage is obtained by the system in CSA and SQA instead. This could severely affect the operation of other subsystems and could require an IPL for resolution.

Consider adding 50MB to ECSA or ESQA storage, depending on how PDSEs are used in your installation.

Allocating Space for a PDSE

This section explains how to use the SPACE JCL keyword to allocate primary and secondary storage space amounts for a PDSE. The PDSE directory can extend into secondary space. A PDSE can have a maximum of 123 extents but cannot extend beyond one volume. A fragmented volume might exhaust extents more quickly because you get less space with each extent.

For example, the following data set is allocated as a PDSE:

```
//NEWLIB DD UNIT=SYSDA,VOL=SER=SMSPAC,DISP=(NEW,KEEP),
// DSN=SOME.DATA,DSNTYPE=LIBRARY,SPACE=(CYL,(1,1,1))
```

The allocation varies, depending upon how your system defines defaults for data class, storage class, and management class.

Because directory space is not preallocated for PDSEs (as it is for partitioned data sets), you do not need to estimate the number of PDSE members to be created. While the number of directory blocks specified on the SPACE keyword is not used for PDSEs, it is saved and is available for conversion utilities. If you specify DSORG=PO when allocating the data set, you can omit the directory quantity. However, you need to specify at least one of DSNTYPE, DSORG, or directory space on the SPACE keyword in order to allocate a PDSE. These values can be supplied by a data class.

See *DFSMS/MVS Using Data Sets* for examples of the SPACE keyword and *OS/390 MVS JCL User's Guide* for more information on how to allocate space.

Migrating Data From Partitioned Data Sets to PDSEs

You can use IEBCOPY or DFSMSdss to convert partitioned data sets to PDSEs. You can convert the entire data set or individual members, and also back up and restore PDSEs. To copy one or more specific members, use the SELECT control statement.

In this example, IEBCOPY copies members A, B, and C from USER.PDS.LIBRARY to USER.PDSE.LIBRARY.

```
//INPDS   DD  DSN=USER.PDS.LIBRARY,DISP=OLD
//OUTPDSE DD  DSN=USER.PDSE.LIBRARY,DISP=OLD
//SYSIN   DD  *
          COPY OUTDD=OUTPDSE
          INDD=INPDS
          SELECT MEMBER=(A,B,C)
/*
```

In the following example, DFSMSdss COPY copies all partitioned data sets beginning with the qualifier of TEST from volume AAAAAA to volume BBBBBB and converts them to PDSEs. You must specify INDY and OUTDY for the input and output volumes unless the ddnames are defined for the volumes.

```
COPY DATASET (INCLUDE(TEST.**)  -
              BY(DSORG = PDS))  -
    INDY(AAAAAA)                -
    OUTDY(BBBBBB)               -
    CONVERT(PDSE(**))           -
    DELETE
```

Note: Once you have migrated to a PDSE, you cannot access the data from a system without MVS/DFP Version 3 Release 2 or higher installed. This might affect your disaster recovery plans.

When copying partitioned data set load modules into a PDSE program library, the program management binder is called to convert the load module into program object format. Copying in the opposite direction also invokes the program management binder to perform the conversion. You should note, however, that program objects exceeding the limitations of load modules cannot be converted successfully.

See *DFSMS/MVS DFSMSdss Storage Administration Guide* for more information on using DFSMSdss and *DFSMS/MVS Utilities* on using IEBCOPY to convert partitioned data sets to PDSEs.

You should convert a PDSE to a partitioned data set when:

- Shipping the PDSE to a system that does not support PDSEs
- An application does not run against a PDSE
- Sharing the PDSE with a system that does not support PDSE access

You can use IEBCOPY or DFSMSdss to convert a PDSE to a partitioned data set. Specify a DSNTYPE of PDS in the JCL or data class definition.

Other PDSE Conversion Issues

Copying members

If you attempt to copy members of partitioned data sets containing user TTRs or note lists to a PDSE, you will get an error message.

Using the SYNCDEV macro

If an application converted for PDSE use invokes the SYNCDEV macro, make sure the application can process the return codes and reason codes for partitioned data sets. The correct return code is 4, indicating that SYNCDEV does not support partitioned data sets.

Using programs that interrogate the VTOC format-1 DSCB

The DS1LSTAR field of the VTOC format-1 DSCB is not maintained for PDSEs allocated or accessed on a DFSMS/MVS system. Consequently, DS1LSTAR is meaningless for a PDSE residing on DASD shared between a DFSMS/MVS system and an MVS/DFP system.

Programming Interface Information

Previously, DS1LSTAR contained the high— allocated track number relative to the data set in the first 2 bytes, and contained a constant in the third byte of the field. Programs that read the DS1LSTAR field for PDSEs could require modifications. An alternative method should be used for determining the high allocated track number for a PDSE. For example, you can derive the number of allocated tracks from the extent information in the format-1 and format-3 DSCB chain. DSCB information is described in *DFSMS/MVS DFSMSdfp Advanced Services*.

End of Programming Interface Information

Sharing Access to a PDSE

You can define the extent to which a PDSE is shared with the PDSESHARING keyword in SYS1.PARMLIB member IDGSMSxx.

If you do not specify the PDSESHARING keyword or you specify PDSESHARING(NORMAL), PDSEs can be shared by all systems in a parallel sysplex for input only.

If you specify PDSESHARING(EXTENDED), PDSEs can be shared by all systems for input and output, provided the systems are set up in a parallel sysplex environment using the cross-system coupling facility (XCF).

PDSE extended sharing is limited to a single sysplex because PDSE extended sharing uses the cross-system coupling facility (XCF) to implement its sharing protocol, and XCF only operates within a single sysplex. If you are sharing PDSEs between multiple sysplexes you should specify PDSESHARING(NORMAL) on each of the systems. You might experience system hangs if you use PDSESHARING(EXTENDED) with multiple sysplexes.

Access to a PDSE is serialized with an enqueue using major names SYSZIGW0 and SYSZIGW1. If you use a serialization program other than GRS, whether the systems are in parallel sysplex mode or not, you need to set up the resource name list (RNL) to explicitly include global resource names SYSZIGW0 and SYSZIGW1, to ensure data integrity across the multiple systems.

Note: When GRS is used, DFSMS/MVS specifies RNL=NO on the ENQ macro issued for PDSEs, bypassing all RNL processing.

To make full use of PDSE sharing at the member level you need to complete the following steps for every system in the parallel sysplex:

1. Verify that the system is in parallel sysplex mode and not XCF-local mode. Refer to *OS/390 MVS Setting Up a Sysplex*.
2. Modify SYS1.PARMLIB member IGDSMSxx to specify PDSESHARING(EXTENDED).
3. Do one of the following to effect the changes in SYS1.PARMLIB:
   - Issue the SET SMS=xx command, identifying the SYS1.PARMLIB member that starts the migration to the EXTENDED protocol.
   - Re-IPL or reset the system.

   Note: Expect to receive informational messages from each system until you reach the last system, which issues a message to indicate that PDSE extended sharing is active.

To change the sharing option back to normal, follow these steps for each system that is running with extended sharing:

1. Change SYS1.PARMLIB member IGDSMSxx member to specify PDSESHARING(NORMAL) or remove the PDSESHARING keyword to allow the system to default to normal sharing.
2. Do one of the following to effect the changes in SYS1.PARMLIB: issue the SET SMS=xx command, re-IPL the system, or reset the system.

To ensure that the complex does not continue with extended sharing, you must reset all systems simultaneously.

For more information about the PDSESHARING keyword, see *DFSMS/MVS DFSMSdfp Storage Administration Reference* . For assistance on resource serialization, see *OS/390 MVS Planning: Global Resource Serialization*.

PDSE Processing Restrictions

The following restrictions apply to PDSEs:

- Although you cannot store load modules in a PDSE, you can store executable program objects into it. Program objects are functional replacements for, and remove many restrictions of, load modules. Program objects are created by the program management binder.
- Do not use PDSEs if your application depends on the system returning short blocks as they were written or sequential access method record null segments. You can create short blocks for PDSE members, but their block boundaries are not shown when the data set is written. For example, if you use the TRUNC macro with QSAM to create short blocks, the block boundaries are not saved when the data set is read. If the TRUNC macro is used, a message is written to the job log and an indicator is set in record type 15 of the SMF data set. See *OS/390 MVS System Management Facilities (SMF)* for more information.
- Null record segments, only created with variable blocked spanned records, are not saved when the data set is written. On the first write of a null record segment, a message is written to the job log and an indicator is set in record

type 15 of the SMF data set. See *OS/390 MVS System Management Facilities (SMF)* for more information.

- A checkpoint data set cannot be a PDSE. Although PDSEs are processed more efficiently than partitioned data sets, checkpoint/restart does not support PDSEs. A failure occurs if a checkpoint is requested when a DCB is open to a PDSE, or if a partitioned data set was opened at checkpoint time but changed to a PDSE by restart time.

  These restrictions do not apply to job and step libraries, or to data sets in the link list that are PDSEs. A job step can also open a PDSE and load a program object from it as long as the PDSE is closed before a checkpoint is taken.

- You can use the BSP macro to backspace the current member one simulated block; however, you cannot backspace beyond the start of a PDSE member nor backspace within the PDSE directory. Issuing the BSP macro while pointing to the middle of a spanned record moves the pointer to the beginning of the record. If the pointer is at the beginning of a record when the BSP macro is issued, the pointer will skip to the beginning of the previous record.
- You can use keyed BSAM for reading the PDSE directory only if the key length specified in the DCB is 0 or 8. You cannot use keyed BSAM for writing PDSE members.
- You cannot write or update the PDSE directory using BSAM or QSAM. Instead, use the STOW macro. This maintains integrity by preventing users from accidentally destroying the directory.
- Aliases for data members must point to the beginning of the member.
- Deleting the primary member name deletes all aliases.
- EXCP and XDAP are not supported.
- You cannot back down a DFSMS/MVS system to an MVS/DFP 3.1 system and still read a PDSE or use objects in a PDSE. PDSEs can only be accessed from a system that supports PDSEs.
- Do not specify a PDSE in the master JCL (MSTJCL00 in SYS1.LINKLIB).
- Note lists are not supported for PDSEs. When using STOW with a PDSE, do not supply a list of TTRs in the user data field of the directory.
- The CHECK macro does not guarantee that the data has been synchronized to DASD. Use the SYNCDEV macro or the storage class parameter SYNCDEV=YES to guarantee synchronizing data when open for update.
- The write-check OPTCD DCB parameter is ignored for PDSEs.
- Do not use the TRKCALC macro because results could be inaccurate, although processing is allowed to continue.
- Program libraries required during IPL cannot be PDSEs. This set includes, but is not limited to: SYS1.NUCLEUS, SYS1.LPALIB and SYS1.SVCLIB.
- SYS1.PARMLIB cannot be a PDSE.
- DB2 DBRM libraries cannot be PDSEs. (See *DB2 Installation Guide* for more information on DBRM libraries.)
- PDSEs cannot generally be accessed in cross-memory mode.

Planning to Access Program Objects

Program objects, created by the program management binder and stored in PDSE program libraries, are not intended to be processed as data. Although they can be read using the BSAM access method, their format is proprietary and not part of any application interface. DFSMSdfp program management does, however, provide a few methods for processing program objects as data:

- A processing program can establish a dialog with the program management binder and obtain object data (for example: ESD, TEXT) in a public format.
- A processing program can invoke the fast data access utility, IEWBFDAT, to obtain the same object data in the same format returned by the binder.
- You can use the program management transport utility to unload a program library or a single program object into a public format similar to that returned by the binder. This transport file can then be processed on any system, regardless of the type or level of operating system available.

These methods are described in more detail in *DFSMS/MVS Program Management*.

Processing Restrictions for Program Objects

There are some restrictions for using certain software products to process program objects:

- Although the AMASPZAP service aid has been enhanced to support program objects, it does not allow program objects to be modified in place; instead, a new version is written into the library. If the library is managed by library lookaside (LLA), then an LLA refresh is necessary to access the replacement module from JCL or programs other than the binder.
- ISPF/PDF provides only limited support for PDSE program libraries. You can create and delete libraries, and list their members. You can delete or rename individual members, and browse, but not edit, program objects. In addition, the program management binder is invoked automatically from the foreground or background linkage editor panels.
- PDSE program libraries or program objects should not be exported to down-level or non-MVS systems using TSO TRANSMIT because the receiving system will not be able to process them.

Planning for Extended Format Data Sets

Extended format data sets have a data set name type (DSNTYPE) of EXTENDED, for example, VSAM extended format key-sequenced data sets (KSDS). All extended format data sets must be SMS-managed.

Extended format data sets provide for the detection of errors that can occur when a physical block of data is written incompletely to a device. Data sets in the extended format are also eligible for compression, a technique used to reduce the space required to house a given piece of data on any type of storage media.

Data striping, available with extended *sequential* data sets, gives you the additional capability of distributing data records in a data set across multiple system-managed

DASD volumes. Transferring data concurrently to multiple devices improves I/O performance and reduces the batch window.

For information on converting to extended format sequential data sets and functions available with this data type, see the following sections of this book:

- "Extended Format Sequential Data Sets" on page 205
- "QSAM or BSAM Compression" on page 191

Migration considerations for VSAM extended format data sets are described under:

- "VSAM Extended Format Data Set Enhancements" on page 72
- "VSAM Extended Addressability" on page 157
- "VSAM Load Enhancements" on page 112
- "VSAM System-Managed Buffering" on page 113
- "VSAM Partial Release" on page 200
- "VSAM Compression" on page 198

Planning for Hierarchical File System Data Sets

A hierarchical file system (HFS) data set is a data set that contains a POSIX-compliant hierarchical file system, which is a collection of files and directories organized in a hierarchical structure that can be accessed using the OS/390 UNIX System Services. It cannot be accessed by standard MVS access methods.

HFS data sets are structured like a tree, based on a root directory with various subdirectories. The files within an HFS data set are identified by their path and file names.

DFSMShsm can automatically back up HFS data sets if it is using DFSMSdss as its data mover, but it cannot back up individual files within an HFS data set.

Once you evaluate OS/390 UNIX and decide to install it in your enterprise, you may proceed with planning tasks such as:

- Deciding which DASD devices will contain the HFS data sets
- Deciding how to control access to them
- Structuring the file systems
- Determining backup, restore, and expiration date policies
- Determining HFS naming policies (file names can be up to 255 characters long and path names can be up to 1023 characters)
- Defining data classes, management classes, and coding ACS routines and JCL statements for HFS data sets

Creating Hierarchical File System Data Sets

An HFS data set is allocated by specifying HFS in the DSNTYPE parameter. You can also define a data class for HFS data sets. All HFS data sets must be system-managed, and an individual HFS data set can reside on only one DASD volume.

RACF or an equivalent security product must be installed and active on your system to use OS/390 UNIX or HFS data sets.

Defining the Root File System

During installation of OS/390 UNIX, code the ROOT statement in the BPXPRMxx member of SYS1.PARMLIB to identify the HFS data set containing the root file system you want MVS to logically mount when you start OS/390 UNIX System Services.

```
ROOT   FILESYSTEM('OMVS.ROOT')
       TYPE(HFS)
       MODE(RDWR)
```

The root file system is the starting point of the overall file structure. It consists of the root directory and any related HFS files or subdirectories. After installation is complete and the MVS system is running, you can create (allocate) an HFS data set, which will contain the root file system.

Creating and Mounting the Root File System

Create an HFS data set to contain the root file system by running a job to allocate the data set. During allocation, OS/390 UNIX builds a basic root directory, which you may alter to meet your specific needs. The parameters for creating the data set are similar to those for creating a PDSE. The difference is that you specify DSNTYPE=HFS to designate an HFS data set.

```
//STEP1  EXEC PGM=IEFBR14
//MKFS   DD   DSNAME=OMVS.ROOT,DISP=(NEW,CATLG),
//            DSNTYPE=HFS,SPACE=(CYL,(1,1,1))
```

Note: This data set must be created on an SMS-managed volume. This can be accomplished either through the ACS routines or by specifying the STORCLAS parameter in the JCL.

The data set must reside on one volume. Like PDSEs, HFS data sets can expand to as many as 123 extents of DASD space.

Write operations present the greatest exposure to file system damage. For this reason the root file system should be small, minimizing the amount of write activity to it, thus offering the least exposure to damage.

Additionally, if all users' files are in file systems that are mounted on the root file system, rather than defined as part of the root itself, and users are denied write access to the root, then the root is further protected from inadvertent damage. Damaged user directories or files can be unmounted and replaced without causing OS/390 UNIX System Services to fail.

Creating Additional File Systems and Directories

After allocating an HFS data set for the root file system and logging on as a TSO/E user, you can define additional directories in the root file system using the MKDIR command. For example, to create the /u/joe directory, issue:

```
MKDIR /u
MKDIR /u/joe
```

These directories can be used as mount points for additional mountable file systems. You can use as a model an IBM-supplied REXX EXEC that creates directories, pseudo-TTY pairs, and device files. Interactive users and application programs can then add files to those additional file systems.

Adding and Mounting File Systems to the Root File System

You can create other mountable file systems with their own directory and data file structures, and mount them on a directory in the root file system or in another file system. Each file system can be logically mounted to a directory (mount point) in another file system by using the TSO/E MOUNT command. Use the UNMOUNT command to unmount a file system. To create and mount additional file systems:

1. Allocate an additional HFS data set by using either the TSO/E ALLOCATE command (example shown) or a JCL DD statement similar to that shown at "Creating and Mounting the Root File System" on page 236.

   ```
   ALLOCATE DSNAME('OMVS.USER.JOE') NEW DSNTYPE(HFS) BLKSIZE(0)
     LRECL(0) RECFM(U) DSORG(PO) SPACE(1,1) CYLINDERS
   ```

   The new data set is allocated with an empty root directory.

2. Have an authorized user enter a TSO/E MOUNT command to logically mount the new file system in the directory of an existing file system.

   ```
   MOUNT FILESYSTEM('OMVS.USER.JOE') TYPE(HFS) MOUNTPOINT('/u/joe')
   ```

You can specify additional file systems to be logically mounted automatically every time OS/390 UNIX is started by adding MOUNT commands to the BPXPRMxx member of SYS1.PARMLIB. The following restrictions apply to mounting file systems:

- The mount point must be a directory.
- Any files in the directory are not accessible while the file system is mounted.
- Only one mount can be active at any time for a mount point.
- A file system can be mounted on only one directory at a time.

You can also create special HFS files to:

- Represent hardware devices (character special files).
- Allow the use of alias names for HFS files (symbolic links).
- Send data from one process to another so that the receiving process reads the data first-in-first-out (FIFO special files, also called named pipes).

The term *process* as used here, is defined as either a program that is created by the fork function, or a program that requests OS/390 UNIX System Services.

Managing File System Size

A file system resides on only one volume. File system size increases as users add files and extend existing files. Eventually, a file system can outgrow the space on its volume. In this case, the storage administrator or system programmer responsible for HFS data sets can either make more space available on the volume by moving individual HFS files to other file systems that have space available, or do one of the following:

- Remove other data sets from the volume on which the full file system resides.
- Move the entire full file system to another volume as follows:
  1. Have an authorized user enter a TSO/E UNMOUNT command to logically unmount the file system.
  2. Allocate an HFS data set with a different data set name on a volume that has adequate space available.
  3. Use the DFSMSdss DUMP function to logically dump the old file system.
  4. Use the DFSMSdss RESTORE function to restore the dumped file system, with a new name, to a volume that has sufficient space. If you want to maintain the original file system name, delete the existing file system first, and then restore it using DFSMSdss without renaming it.
- Remove files from the full file system by either deleting them or by moving them to another file system on another volume. If it is impossible to remove the initially chosen files from a particular directory in the file system, it may be possible to remove other files from a different directory in the same file system—the objective being to reduce the size of the file system.
- Create a new file system on another volume and move some files from the full file system to the new file system. To avoid problems that might result from this approach, define symbolic links using the original names.

Managing File System Activity

If activity for a file system becomes so extensive that accesses are slow, you can do one of the following:

- Move the file system to a volume that processes I/O more quickly, because it has, for example, a faster channel or a cached storage control.
- Move a subtree from the full file system into a new file system on a different volume. Mount the new file system on the now-empty directory, which was the head of the subtree. This divides I/O activity between two volumes. To avoid failures from this action, define symbolic links using the original names.

Check the DDNAME parameters of the MOUNT statements in the BPXPRMxx member of SYS1.PARMLIB to make sure they are correct for the next time you start OS/390 UNIX.

Transporting a File System

You might also want to copy a data set to a storage medium that can be physically transported to another location. To do that, do one of the following:

- Use either the PAX, CPIO, or TAR shell commands to copy the file system in tape archive (TAR) format.

- Have an authorized user logically unmount the file system, allocate an HFS data set with a different data set name, and use the DFSMSdss dump utility to copy the old file system to the new data set.

Removing (Deleting) a File System

If the file system (HFS data set) is mounted to another file system, logically unmount it using the TSO/E UNMOUNT command against the HFS data set containing it.

Do one of the following to remove the file system:

- Use the DELETE command (IDCAMS or TSO) with the SCRATCH parameter
- Execute an IEFBR14 job with DISP=(OLD,DELETE) specified for the HFS data set

Migrating a File System

If a file system is unmounted and remains so for a predetermined time, DFSMS/MVS can migrate it to a lower priority storage medium. DFSMS/MVS automatically recalls a migrated file system from migration storage if a mount command is issued for the file system.

If you plan to migrate HFS data sets, migrate them only to level 1 (DASD) storage. Recalling an HFS data set that was migrated to tape could adversely affect performance because of the time required to physically mount the volume.

Backing Up File Systems

DFSMShsm provides automatic backup facilities for HFS data sets. You can back up mountable file systems by periodically backing up the HFS data sets that contain them; the data sets can be restored if necessary. DFSMShsm is also used for migrating and restoring unmounted file systems.

You can manually back up a mountable file system, including the root file system, periodically with DFSMSdss data set dumps. To do this, issue the DFSMSdss DUMP command. This quiesces activity against the specified HFS data set, then invokes backup processing. When the backup is complete, the file system is unquiesced and user activity can resume.

Retain periodic DFSMSdss data set dumps of file systems in case a program fails and damages files and directories. Keep this backup in another area or different building in case the computer site experiences physical damage.

There is no facility for automatically backing up individual files within an HFS data set. You can manually back up files with the PAX, CPIO, and TAR shell commands.

Recovering a Backed-Up File System

If a file system is damaged, recover by replacing it with a saved version created earlier by a backup. To recover a backed up file system, the operator must:

1. Notify all users of the damaged file system that they must stop activity against it. Because of the file system damage, all activity has probably stopped already.

2. Have an authorized user enter the TSO/E UNMOUNT command with the IMMEDIATE option to logically unmount the damaged file system. If the unmount fails, reenter the UNMOUNT command with the FORCE option.
3. Use the DFSMSdss dump utility to restore the backed up file system to a replacement file system (HFS data set).
4. Have an authorized user enter a TSO/E MOUNT command to logically mount the replacement file system.
5. Issue a broadcast message to all users or a message to all OS/390 UNIX users when they invoke the shell, telling them that you have mounted a back-level file system and informing them of the mount point. Users must recreate and add any files added since the file system was backed up.

Planning for Tape in a System-Managed Environment

Converting to system-managed tape can help you optimize your installation's tape subsystem operation and media utilization by:

- Allowing the system to manage the placement of tape data
- Automating tape mount and demount operations
- Using the full capacity of the tape media
- Taking advantage of hardware and software compaction
- Exploiting new technologies for removable media.

System-managed tape allows users to define requirements for their tape data in logical, rather than physical, terms. Without requiring changes to programs or JCL, the storage administrator defines the policies the system uses to map those logical requirements to physical tape resources, which can include mixed device types (such as IBM 3490 or 3490E transports), mixed media (Cartridge System Tape or Enhanced Capacity Cartridge System Tape), and tape library dataservers.

This chapter introduces what you need to do in planning for a system-managed tape environment. More implementation considerations, and actual sample plans for managing tape data and volumes under SMS, are presented in the *DFSMS/MVS Implementing System-Managed Storage* .

Optimizing Your Current Tape Environment

Depending on your installation's current tape practices, you could maximize the use of your tape resources and even improve batch job throughput by implementing tape mount management (TMM).

Tape mount management is a methodology for managing tape data sets within the DFSMS storage hierarchy:

1. Tape data sets are categorized according to size, pattern of usage, and other criteria, so that appropriate DFSMS policies can be assigned to tape mount management candidates.
2. Data sets written to tape are intercepted at allocation and, if eligible for tape mount management, redirected to a system-managed DASD buffer. The buffer serves as a staging area for these data sets until they are written to tape. The location of the data is transparent to the application program.
3. DFSMShsm periodically checks the occupancy of the DASD buffer storage group to ensure that space is available when needed, migrating data sets to a lower level of the storage hierarchy when they are no longer required on primary DASD volumes.
4. DFSMShsm eventually moves the data to tape, using single-file format and data compaction to create full tape cartridges.
5. If an application later requests a data set, DFSMShsm automatically recalls it from where it resides in the storage hierarchy, and allocates it on primary DASD for access.

This process can significantly reduce tape mounts and the number of cartridges required to store the data. Operations can benefit from a decreased number of

© Copyright IBM Corp. 1979, 1999

random tape mounts, while applications benefit from improved job throughput (because the jobs are no longer queued up on tape drives).

Note: You cannot use tape mount management for OAM objects that are written to tape.

Converting tape data sets to system-managed storage requires careful planning. For tape mount management, you should include the following major tasks in your implementation plan:

1. Analyze the current tape environment.
2. Simulate the proposed tape mount management environment.
3. Implement advanced cartridge hardware.
4. Select DASD volumes to satisfy buffer requirements.
5. Define SMS classes and storage groups for candidate data sets.
6. Create the ACS routines, or add to the current routines.
7. Tune DFSMShsm operations.

Although JCL changes are not usually necessary for implementing tape mount management, you do need to determine the impact of jobs that leave tape data sets uncataloged, and special expiration date codes that are used by some tape management systems, if these practices exist in your current environment. These, and other planning considerations are described in *DFSMS/MVS Implementing System-Managed Storage* .

"Tape Mount Management Enhancements" on page 149 describe the expanded tape mount management support for data set stacking, unit affinity and volume reference processing.

Analyzing Tape Usage with the Volume Mount Analyzer

The *volume mount analyzer* (VMA) is a tool that can help you analyze your installation's current tape environment.

The volume mount analyzer produces reports that profile your tape mount workload and tape media usage, so that you can decide whether tape mount management might benefit your installation. In addition, the volume mount analyzer identifies data sets that, because of their large size, can benefit from advanced cartridge hardware and media technology.

The volume mount analyzer uses your installation's SMF data to analyze tape mount activity and to produce reports that help you:

- Identify trends and other information about tape mount events, including data set name, job, program, and data set size (bytes transferred).
- Evaluate the tape hardware configuration.
- Quantify the benefits of tape mount management in terms of library and tape mount reduction.
- Determine which data sets are good candidates for tape mount management.
- Determine data class and management class requirements for tape mount management candidates
- Develop ACS routine filters to select tape mount management candidates and exclude other data sets that must remain on tape.

- Determine the size of the DASD buffer and the high and low thresholds needed for the buffer's storage group.

To run the volume mount analyzer, you must have DFSORT installed.

In a volume mount analyzer study, you:

1. Determine a representative time period for the study; usually a one-month cycle that includes peak loads as well as routine processing.
2. Collect the required SMF record types during the study period to record tape information used by the volume mount analyzer.
3. Run the volume mount analyzer's SMF data extractor program, GFTAXTR, to reduce the amount of input data.
4. Run the volume mount analyzer program, GFTAVMA, to generate the summary and detailed reports.
5. Analyze the results. At this point you might want to modify parameters for GFTAVMA, and rerun the program until you are satisfied with the results.
6. Determine whether you want to implement tape mount management.

For detailed procedures on performing a volume mount analyzer study and interpreting the results, see *DFSMS/MVS Using the Volume Mount Analyzer*. For information on implementing tape mount management using the volume mount analyzer reports, consult *DFSMS/MVS Implementing System-Managed Storage* and *MVS/ESA SML: Managing Data*.

Converting Tape Volumes to System Management

Installing tape library dataservers and converting your volumes to system management offers the following benefits:

- Tape mounts and demounts are automated: DFSMS/MVS and the tape library dataserver work together to automate the retrieval, storage, allocation, and control of standard and enhanced capacity cartridge system tape volumes.
- Tape subsystem performance is improved through the enhanced device selection process used for system-managed tape.
- Compaction and media interchange attributes are more easily specified for system-managed tape volumes.
- Disaster recovery procedures are simplified: data can be written to tape volumes at an offsite tape library dataserver during backup processing and, in the event of a local disaster, rebuilt from those offsite volumes.

When you allocate a new data set in the automated environment, the system selects an appropriate tape cartridge from a scratch tape pool in the tape library dataserver. If you request a data set that is already stored on a cartridge in a tape library dataserver, DFSMS/MVS enables the dataserver to automatically locate, mount, demount, and store the correct tape cartridge for you.

DFSMSrmm also plays an important role in this environment, enabling the tape library dataserver to automatically recycle tapes back into the scratch pool when the data sets on the tapes expire, eliminating the need for you to manually change these tapes back into scratch tapes. DFSMSrmm invokes the object access method

(OAM) to update the volume's status during the recycle process. It also causes OAM to eject cartridges from the tape library dataserver for the purpose of moving volumes offsite in preparation for disaster recovery.

Conversion Activities

To prepare your installation for system-managed tape, you need to include the following major tasks in a conversion plan:

1. Evaluate job dependencies on the JCL UNIT parameter.

   Consider the impact of certain SMS restrictions on JCL UNIT parameter usage. Because the device used to satisfy an allocation request is selected from device pool information associated with a given tape library:

   - Demand allocation is not supported. For example, UNIT=520 (where 520 is the address of a 3490 drive) is not valid.
   - The UNIT keyword is not actually used to select a tape device. However, this keyword is available to your ACS routines to use for filtering purposes.
   - Devices requested using unit affinity (such as UNIT=AFF=DD1), are honored only if the volumes reside in the same tape library and use compatible devices.

   Except for demand allocation, JCL changes are not required to use tape library dataservers.

2. Define the tape environment to MVS.

   Tape drives in a tape library dataserver are defined using the hardware configuration definition. A library ID defined to the ISMF Library application links the system-managed tape library definition to the tape library dataserver. The library ID is defined to HCD by specifying the LIBID and PORTID parameters for each library device. HCD help text for the LIBID and PORTID parameters explains how you can obtain the IDs.

3. Define OAM.

   Tape library support uses OAM to define system-managed tape libraries and volumes. You need to update various PARMLIB members to define OAM. See *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.

4. Define the classes, storage groups, and associated ACS routines.

   You can use data classes established when you implemented tape mount management to control the categories of tape data sets that are allocated on system-managed tape volumes. You can also define special classes for managing DFSMShsm-owned volumes in a tape library, and for tapes that hold objects.

   If your tape management application supports the pre-ACS interface, you can use the information available to you (for example, destination) when coding the ACS routines to direct tape allocations to specific libraries, to a DASD pool (for tape mount management), or to keep them outside of the system-managed environment.

   Tape storage groups, defined under ISMF, associate the tape libraries to tape storage groups. A scratch tape volume becomes system-managed when it is entered into a system-managed library. A scratch tape becomes part of a storage group when a system-managed data set or object is allocated on the

volume. System-managed volumes are assigned to a tape storage group. Tapes that contain objects also belong to either an object or an object backup storage group.

After you define a tape storage group, you must set the status for the storage group on each system that uses the tape library dataserver.

5. Create the tape configuration database.

   You must define one general volume catalog. One or more specific volume catalogs can also be defined based on your installation's requirements. The collection of your installation's general and specific volume catalogs is the *tape configuration database*.

   DFSMS/MVS determines which catalog to update based on the first character of the volume serial number. Naming conventions for volume serial numbers can help you balance catalog update activities.

   You can use access method services to define volume catalogs and use standard ICF support for backup and recovery.

   Before you define your tape libraries, make sure that update authority for your tape configuration database is restricted to storage administrators.

6. Define the tape libraries.

   Create a logical tape library definition for each grouping of tape volumes associated with a collection of tape devices. This definition becomes part of your active SMS configuration and a library entry is generated in the tape configuration database. It is created using a new ISMF application that is invoked from the Library Management application. ISMF also allows you to redefine the tape library from information in the tape configuration database.

7. Create any system-managed tape exits, if required.

   There are several DFSMSdfp installation exits that you can use specifically with tape library dataservers. DFSMSrmm also uses selected DFSMSdfp exits to manage some of its tape processing.

   To find out how these exits affect library management, see *DFSMS/MVS Installation Exits* and *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* .

8. Translate and validate the new SMS configuration.

   Use the same translation and validation steps you follow for most other SMS configurations. The new configuration is considered valid if all tape libraries associated with tape storage groups exist for SMS.

9. Test the new SMS configuration.

   Use ISMF to write and run test cases to verify that your new, or modified, ACS routines properly assign the new tape classes and storage groups.

10. Activate the new SMS configuration.

    Activate the new system-managed tape configuration as you would other SMS configurations.

11. Start OAM.

    Place the start-up PROC for OAM in MVS PROCLIB. You must have OAM running on all processors that will use the automated libraries.

12. Enter volumes in the tape library dataserver.

Add your tape cartridges to the tape library dataserver before you begin testing allocations.

13. Test library usage for SMS tape allocations.

    Perform selective allocations to check library usage before converting the rest of your tape data to system-management.

14. Put the tape library dataserver into production.

    Consider converting your tape data to system management by category of data, using the following suggested order:

    Large temporary data sets
    DFSMShsm-owned volumes
    Offsite volumes
    Active volumes
    Backup volumes

DFSMS/MVS Implementing System-Managed Storage expands on these tasks, giving you more detailed procedures and examples. You should use this book in planning the conversion to system-managed tape. See *DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support* for information on using tapes to hold OAM objects.

"Preparing to Implement DFSMSrmm" on page 247 and "Preparing to Implement DFSMShsm" on page 257 also discuss tape planning issues.

See the following books for help during various stages of the installation and conversion process:

3495 Tape Library Dataserver Installation Planning and Migration Guide
OS/390 JES3 Initialization and Tuning Guide
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries

Preparing to Implement DFSMSrmm

This chapter helps you plan for DFSMSrmm implementation. Use it to assess your removable media environment and to estimate the amount of DASD space you need for DFSMSrmm data sets. The chapter introduces DFSMSrmm utility programs, security options, and programming interfaces. It also describes DFSMSrmm's interaction with DFSMShsm and its dependency on Open/Close/End-of-Volume processing.

see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* for more information about DFSMSrmm implementation.

Evaluating Removable Media Management Needs

Answer these questions to assess your current tape management practices and to anticipate future requirements. If you have plans to make changes to your removable media library, such as increasing the number of volumes, include these changes in your answers. Use this information to assess DASD space requirements for your DFSMSrmm implementation.

Table 11 (Page 1 of 2). Evaluating Removable Media Management Needs

Task	Subtask
Determine the number of resources you have in your removable media library.	How many volumes do you have in your removable media library? A *volume* is any type of removable media, such as a tape cartridge or an optical disk. Add an average of five volumes in your count for each software product in your installation. How many shelf locations or slots do you maintain in your removable media library and in your storage locations? A *shelf location* is a single space on a shelf where you store a volume. Count all shelves in the library and in your storage locations. For DFSMSrmm subcommands and the ISPF dialog, shelf locations in the removable media library are called *rack numbers*. Shelf locations in storage locations are called *bin numbers*. How many data sets do you have on removable media? Count any data sets on your removable media. How many different individuals or groups use removable media? DFSMSrmm can keep track of owners and of removable media in the DFSMSrmm control data set.
Determine the number of requests submitted to your removable media library.	How many scratch tape mounts are performed daily? A *scratch* tape mount is a non-specific tape mount as, for example, when someone requests a blank tape. How many non-scratch tape mounts are performed daily? A *non-scratch* tape mount is a specific tape mount as, for example, when someone requests a tape he or she owns or a software product tape.

© Copyright IBM Corp. 1979, 1999

Table 11 (Page 2 of 2). Evaluating Removable Media Management Needs

Task	Subtask
Determine the types of activities taking place in your media library.	What activities are performed to support disaster recovery and vital records management? How many volumes enter and leave your removable media library daily? This includes volumes moving to storage locations for disaster recovery and vital records, as well, as foreign tapes entering your library. How many volumes are returned to scratch daily? This number can be used to calculate the space required for the journal. How many volumes expire daily? This number can be used to calculate the space required for the journal.
Determine the number of information changes that might be made to DFSMSrmm information.	This number can be used to calculate the space required for the journal. Changes include information about data sets, owners, software products, or volumes made by using the DFSMSrmm TSO subcommands or DFSMSrmm ISPF dialog.

Assessing DASD Space Requirements

This section helps you assess how much DASD space you need for DFSMSrmm-owned data sets: the DFSMSrmm control data set and the journal and the problem determination aid log data sets.

DASD Space for the DFSMSrmm Control Data Set

The DFSMSrmm control data set is a VSAM key-sequenced data set that contains the complete inventory of your removable media library. DFSMSrmm records all changes made to the inventory, such as adding or deleting volumes, in the control data set. Multiple systems that share DASD can use the same control data set.

Calculating DASD Space for the DFSMSrmm Control Data Set

Use the numbers you developed when evaluating your resources and removable media activities with the information in Table 12 to calculate DASD space for the DFSMSrmm control data set.

Table 12. DFSMSrmm Control Data Set DASD Space Requirements

Control Data Set Content	DASD Space
Control record	1KB
Data sets	416KB for every 1000
Shelf locations in the library that do not contain volumes	140KB for every 1000
Shelf locations in storage locations	140KB for every 1000
Owners	38KB per 1000
Software products, average five volumes per product	420KB for every 1000
Volumes	1 MB for every 1000
Vital record specifications	212KB for every 1000

After calculating the previous figures, increase the total by approximately 50% for free space in the VSAM key-sequenced data set. For example, if the calculated size is 3000KB, increase it by 50% to 4500KB to allow for free space.

Placing the DFSMSrmm Control Data Set

If you place the control data set on a volume that is system-managed, use a storage class with the GUARANTEED SPACE attribute.

If you are running DFSMSrmm on more than one system, create the control data set on a shared volume, ensuring that all systems have access to it. The control data set must be cataloged in a shared user catalog. Put the control data set and journal on different volumes. Separating the two data sets optimizes data integrity, because the journal is a copy of the changes made to the control data set and is used during control data set recovery.

If you plan to use DFSMSdss to back up the DFSMSrmm control data set, consider placing the control data set on a concurrent copy capable volume. Use of concurrent copy for the DFSMSrmm control data set backup allows tape activity which updates the control data set to continue during the backup operation.

DFSMSrmm uses RESERVE/RELEASE to control access to the control data set and ensure integrity. If your installation is using global resource serialization, add the name of the RESERVE enqueue name to the global resource serialization conversion list. This prevents the enqueue that global resource serialization uses from being propagated around the global resource serialization ring. The hardware reserve ensures that the control data set cannot have simultaneous updates to it. Both the SYSZRMM MASTER.RESERVE and SYSZVDDS resources must be treated the same to avoid lockouts between systems on these resources. For more information about DFSMSrmm using global resource serialization, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

Although using global resource serialization is optional, global resource serialization enhances DFSMSrmm performance.

Use IBM 3990 Storage Control with cache for the control data set, and use DASD fast write to improve performance. This is beneficial if you are running DFSMSrmm in a multihost environment where control data set buffers are always refreshed.

You can also place the control data set on a device that supports the dual copy feature of dual copy-capable IBM 3990 Storage Control and use the storage class AVAILABILITY=CONTINUOUS attribute. Duplexing is different from journaling, and you should not replace journaling with duplexing. Duplexing records the same data in two places to provide protection against hardware failure. Journaling provides protection against accidental loss of critical data by allowing you to reconstruct it using journal contents.

Ensure you place the control data set index and data components on the same volume because DFSMSrmm processing does not support putting the index and data components on separate volumes.

Note: To avoid a potential deadlock on the volume where the DFSMSrmm control data set is placed, you should consider the other data that you place on the volume. A deadlock can occur when a program other than DFSMSrmm reserves the volume where the control data set resides and requests

DFSMSrmm for service. If the DFSMSrmm control data set requires additional extents because of the request, then a deadlock can occur.

DASD Space for the Journal

The journal is a sequential data set that contains a chronological record of changes made to the DFSMSrmm control data set. DFSMSrmm uses the journal to perform any necessary forward recovery of the control data set.

Calculating DASD Space for the Journal

Use the numbers you developed when evaluating your resources and removable media activities with the information in Table 13 to calculate DASD space for the DFSMSrmm journal.

For example, plan for 8.3KB of DASD space for each scratch mount. Only plan space for those scratch mounts that are recorded in the journal. To ensure that the journal has enough space when there is an unexpected increase in tape activity, we recommend increasing the calculated size by 50% or more. The journal size limit for DFSMSrmm 1.1.0 and DFSMSrmm 1.2.0 is 32,764 tracks. The journal size limit for DFSMSrmm 1.3.0 or higher is 65,535 tracks.

Table 13. DFSMSrmm Journal DASD Space Requirements

Journal Content	DASD Space
Changes by users	1.5KB for each change made
Data sets	1.5KB for each data set retained by a vital record specification
Expiring volumes	1.5KB for each expiring volume
Non-scratch mounts	6.7KB for each mount
Scratch mounts	8.3KB for each mount
Volumes	1.5KB for each volume retained by a vital record specification
Volume checked in or checked out	2.6KB for each volume in or out of the library
Volumes returned to scratch	3.0KB for each volume returned to scratch
Volumes to and from storage locations	3.5KB for each volume moved to or from a storage location
Vital record specifications	1.3KB for each vital record specification created

Convert the final figure into a space allocation. The journal has a record format of variable-length blocked records. You do not need to specify the record format information when allocating the journal, because DFSMSrmm sets the correct values when it opens the journal data set.

To calculate the space required, divide the total KB of space by 4, as allocation will be by average record size using a 4K value.

Placing the Journal

Place the journal and control data on different volumes. If the selected volume is system-managed, use the STORCLAS parameter. Select a storage class with the GUARANTEED SPACE attribute to ensure that the journal is allocated to a specific volume. If the volume is not system-managed, remove the STORCLAS parameter.

You can share the journal between systems. If your control data set is shared, your journal must be shared.

DASD Space for the Problem Determination Aid Log Data Sets

The problem determination aid (PDA) facility gathers DFSMSrmm processing information to enable analysis to pinpoint module flow and resource usage related to DFSMSrmm problems. The PDA facility is required for IBM service because it traces module and resource flow. The PDA facility consists of in-storage trace, optional DASD log data sets, EDGRMMxx parmlib member options, and operator commands to control tracing.

DFSMSrmm accumulates problem determination information at specific module points in the form of trace data, and it records this data in main storage. At predetermined intervals, the trace data is scheduled for output to DASD. The DFSMSrmm trace recording function receives the trace data scheduled for output and writes this data to a file on DASD. DFSMSrmm recognizes optional log data sets by their DD names, EDGPDOX and EDGPDOY. Recording takes place in the data set defined by EDGPDOX. When that data set is filled, the two data set names are swapped, and recording continues on the newly renamed data set.

When this data set is filled, the names are again swapped, and the output switches to the other data set, thus overlaying the previously recorded data. The larger the data sets, the longer the period of time that is represented by the accumulated data.

We recommend an initial PDA log data set size of 20 cylinders. You can adjust the size as you gain more experience with the amount of activity that is traced in your installation. *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* includes worksheets you can use to calculate the data set sizes.

Planning DFSMSrmm Authorization and Security

You can choose the authorization levels of users for all DFSMSrmm functions. DFSMSrmm uses System Authorization Facility (SAF) for its authorization checking.

For its security checking, DFSMSrmm uses RACF. DFSMSrmm provides support for using RACF standard tape volume security protection, using any combination of RACF TAPEVOL and TAPEDSN options. DFSMSrmm can create volume profiles, change them, and delete them on registration, expiration, or release of volumes. To cause DFSMSrmm to create a RACF TAPEVOL profile, you can use the access list DFSMSrmm provides. You can use this access list for authorization checking on non-RACF systems.

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

- Audit trail of access which includes information about RACF user ID, group, and job name
- Operator confirmation of use
- Erasure of data on volume release and recycle

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

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

Deciding How to Run DFSMShsm with DFSMSrmm

Run DFSMSrmm with DFSMShsm to enhance the management of the volumes DFSMShsm uses. For example, DFSMSrmm can manage the movement of tapes that must be sent out of the library for disaster recovery. Using DFSMSrmm and DFSMShsm together, you can use the scratch tape pool rather than a DFSMShsm tape pool.

DFSMSrmm provides the DFSMSrmm EDGTVEXT and EDGDFHSM programming interfaces for use by products like DFSMShsm and ADSM for using DFSMSrmm tape management. DFSMSrmm treats DFSMShsm like any other user and retains volumes based on retention period and vital record specifications. DFSMShsm automatically calls EDGTVEXT so you no longer need to use ARCTVEXT to communicate with DFSMSrmm. A benefit of this interaction is that DFSMSrmm can prevent DFSMShsm from overwriting its own control data set backup, automatic dump, ABARS, and copies of backup or migration tapes. Although DFSMShsm checks its own migration and backup tapes, DFSMSrmm checks them as well.

Before you can use DFSMSrmm with DFSMShsm, you are required to authorize DFSMShsm to STGADMIN.EDG.MASTER, STGADMIN.EDG.OWNER.user, and STGADMIN.EDG.RELEASE. If you have multiple DFSMShsm USER IDs, for example in a multi-system environment, and any DFSMShsm ID can return tapes to scratch status or return tapes to the DFSMShsm tape pool, you must authorize each DFSMShsm USER ID. Define STGADMIN.EDG.OWNER.hsmid for each DFSMShsm USER ID and give the other DFSMShsm USER IDs UPDATE access to it.

When using DFSMSrmm with DFSMShsm, you can use expiration dates or define DFSMSrmm vital record specifications to identify how DFSMShsm volumes should be retained. For more information about DFSMSrmm and DFSMShsm interaction, see *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*.

Planning for Operating DFSMSrmm

Consider these tasks when planning for operating DFSMSrmm.

Set up procedures for your operators to follow to start and stop DFSMSrmm, respond to DFSMSrmm messages, and to perform tasks like labeling tapes. *DFSMS/MVS DFSMSrmm Guide and Reference* includes a chapter called DFSMSrmm Operator Procedures that you can use to help develop your own pro-

cedures. The chapter includes information on starting and stopping DFSMSrmm and operator responses to DFSMSrmm messages.

Set up DFSMSrmm so you can maximize the use of DFSMSrmm functions. *DFSMS/MVS DFSMSrmm Implementation and Customization Guide* provides information on topics including:

- Using the DFSMSrmm EDGRMMxx parmlib member to control DFSMSrmm operation and functions.
- Using the DFSMSrmm utilities to perform DFSMSrmm inventory management to process the policies you have defined for the resources that DFSMSrmm manages for you.
- Setting up DFSMSrmm so you can use the catalog status tracking function to ensure that your control data set and catalogs are synchronized.

Planning for DFSMSrmm Customization

DFSMSrmm uses DFSMSdfp installation exits and provides installation exits for managing volumes and data sets. To use the installation exits, DFSMSrmm provides programming interfaces. The installation exits help you tailor information being converted to DFSMSrmm. It is especially important to plan for these exits if you have an existing media management system.

Table 14 lists the installation exits used with DFSMSrmm. Before installing DFSMSrmm, ensure that there are no conflicts between how DFSMSrmm uses these exits and how your installation currently uses them.

Table 14. Installation Exits Used with DFSMSrmm

Exit	Description
CBRUXCUA	DFSMSrmm uses OAM's change use attribute exit to manage the volumes in system-managed tape libraries.
CBRUXEJC	DFSMSrmm uses OAM's cartridge eject exit to manage the volumes in system-managed tape libraries.
CBRUXENT	DFSMSrmm uses OAM's cartridge entry exit to manage the volumes in system-managed tape libraries.
CBRUXVNL	DFSMSrmm uses OAM's volume-not-in-library installation exit to process tape volumes that are not resident in system-managed tape libraries but that are needed for processing to continue.
EMODVOL1	DFSMSrmm uses the DFSMSdfp End-of-Volume label editor to control tape-volume labeling.
IATUX71	DFSMSrmm uses the JES3 exit to determine the processing needed for JES3 messages and tape drive display processing.
IGXMSGEX	DFSMSrmm uses the DFSMSdfp MSGDISP exit to update tape drive displays.
OMODVOL1	DFSMSrmm uses the DFSMSdfp Open Volume label editor to control tape-volume labelling.
IFG019VM	DFSMSrmm provides a sample exit you can choose to use so that your operator can supply tape label information.

DFSMSrmm provides object code for EMODVOL1, called EDGEMVOL, and for OMODVOL1, called EDGOMVOL. After completing DFSMSrmm processing, these exits transfer control to any existing exits defined by your installation that are named as EMODVOL1 or OMODVOL1. If you are now using OMODVOL1 or EMODVOL1 and the CSECT names are OMODVOL1 or EMODVOL1, you do not have to do anything to modify the exits. DFSMSrmm correctly includes your existing code.

Planning to Use DFSMSrmm Utilities

DFSMSrmm provides utilities that you can plan for now and implement after installation using the directions in *DFSMS/MVS DFSMSrmm Implementation and Customization Guide*. They include:

EDGHSKP

Provides the following inventory management functions:

- Synchronization of the ICF catalogs with the DFSMSrmm control data set.
- DFSMSrmm vital record processing to determine which volumes to retain and what volume moves are required based on vital record specifications. You can run a trial run of inventory management vital record processing to test policy changes before DFSMSrmm uses them in a production run. Trial run vital record processing does not change data set and volume information in the DFSMSrmm control data set. Using trial run processing, you have an opportunity to analyze the effect that your movement and retention policies will have. Based on your analysis, you can then determine if you need to change vital record specifications before performing production run vital record processing. This is helpful when you are defining your initial set of policies and when you need to make changes as you gain more experience with DFSMSrmm.
- Expiration processing to identify volumes ready to be released.
- Storage location management processing to assign shelf locations to volumes being moved to storage locations.
- Control data set and journal backup.
- Report extract data set creation. You can tailor your own reports by producing an extract data set from the control data set and creating a report from it with EDGRPTD or the EDGRRPTE exec.

EDGAUD and EDGRPTD

Help you get information about your removable media library and storage locations. You can obtain information about volume movement and data sets defined to DFSMSrmm, and audit trail information about volumes, shelf assignments, and user activity.

EDGBKUP and EDGUTIL

Back up and recover the control data set, check the integrity of the information contained in it, and back up the journal. Use the EDGBKUP utility to back up and recover the control data set and journal, and the EDGUTIL utility to create, update, verify, and mend the control data set.

DFSMSrmm backup utilities check whether the control data set is in use, tell the DFSMSrmm subsystem that backup or recovery is in process, and provide a way to forward recover.

Use DFSMSrmm backup utilities rather than other backup utilities, such as access method services REPRO, because DFSMSrmm provides the necessary serialization and forward recovery functions.

EDGINERS
Helps you erase and initialize tape volumes either automatically or manually. You can choose to use EDGINERS to replace the DFSMSdfp utility IEHINITT to take advantage of DFSMSrmm initialization and erase functions. EDGINERS writes BCD labels on 7–track tape volumes and ASCII (ISO/ANSI) on tape cartridges or 9–track tape volumes. EDGINERS provides support for ANSI Version 3 VOL1 and HDR1 labels and for ANSI Version 4 VOL1 and HDR1 labels.

You schedule these utilities and run them independently or as a sequence of events. Use your own scheduling facility, such as OPC/ESA. For example, run vital record processing daily, but storage location management processing and movement reports weekly.

Planning Dependencies on Open, Close, End-of-Volume

DFSMSrmm provides object code included in open, close, end-of-volume load modules. These do not modify any open, close, end-of-volume code at installation time, but for correct execution, have a dependency on the open, close, end-of-volume code. If you have any products installed that modify any open, close, end-of-volume code, check exactly what they modify.

The open, close, end-of-volume load modules that DFSMSrmm link-edits into are:

IFG0194A
IGC00020
IGC0005E
IGC0001I
IGC0002C

If you are replacing an existing media management system, analyze how your product interacts with open, close, end-of-volume compared to how DFSMSrmm interacts with open, close, end-of-volume.

Preparing to Implement DFSMShsm

DFSMShsm provides automatic management of low-activity and inactive data, and automatic backup and recovery of active data in both SMS-managed and non-SMS-managed environments. This section will help you to plan your installation of DFSMShsm by discussing the requirements for your environment and policies for availability and space management.

A sample setup plan, “Sample Setup Plan” on page 279, can help you create your overall implementation plan. Other sample plans are included at the end of the chapter and referred to in the appropriate areas of the text.

For more details about any of these topics, you can read the *DFSMS/MVS DFSMShsm Implementation and Customization Guide* . Other reference books will be cited when additional information could be helpful.

Developing a DFSMShsm Implementation Plan

You should implement DFSMShsm in stages, so that each function can be tested and tuned before you add the next one. We recommend that you start your implementation in the following order, whether you select one, or a combination, of these functions:

- Automatic backup
- Automatic space management
- Automatic dump
- Aggregate backup and recovery support (ABARS)

Planning the Processor Environment

You can use DFSMShsm in either a single- or a multi-processor environment.

In a single-processor environment, the processor can run all DFSMShsm functions either concurrently or sequentially. The operating system's internal serialization sufficiently protects control data sets from being changed while DFSMShsm is processing them.

In a multi-processor environment, where processors share the control data sets, different processors can perform each function, or all the processors can perform all the functions.

You should designate one processor as a primary processor. This processor then backs up the control data sets and journal and migrated data sets; moves backup versions of data sets from migration level 1 (ML1) volumes to backup volumes; and deletes expired dump copies.

In a multi-processor environment, you must specify that automatic backup and automatic dump should run in the primary processor. To run backup and migration concurrently, you can run multiple tasks for each of the functions, distributing them to different processors. All the necessary volumes must be available to all the processors.

© Copyright IBM Corp. 1979, 1999

You can, however, use attributes in the storage groups for SMS-managed volumes, or DFSMShsm parameters for non-SMS-managed volumes, to direct backup and dump operations to a nonprimary processor.

Planning for Data Security

You should include security in your DFSMShsm implementation plans to prevent unauthorized access to data resources. DFSMShsm, together with RACF, offers several ways to protect your DFSMShsm-managed data and functions. For more information on RACF, see “Securing DFSMS/MVS Resources” on page 287.

Preparing DFSMShsm Data Sets and Volumes

DFSMShsm uses dedicated data sets and logs for internal record keeping, reporting, and performance analysis. The starter set provided with DFSMShsm allocates the appropriate data sets, but you define the devices on which the data sets are allocated. Ensure that all DFSMShsm data sets are allocated on low-use, mounted volumes. This enhances performance by reducing data set contention, and by simplifying checkpoint and recovery procedures for the control data sets. Figure 6 gives you an overview of the DFSMShsm environment for these data sets and logs.

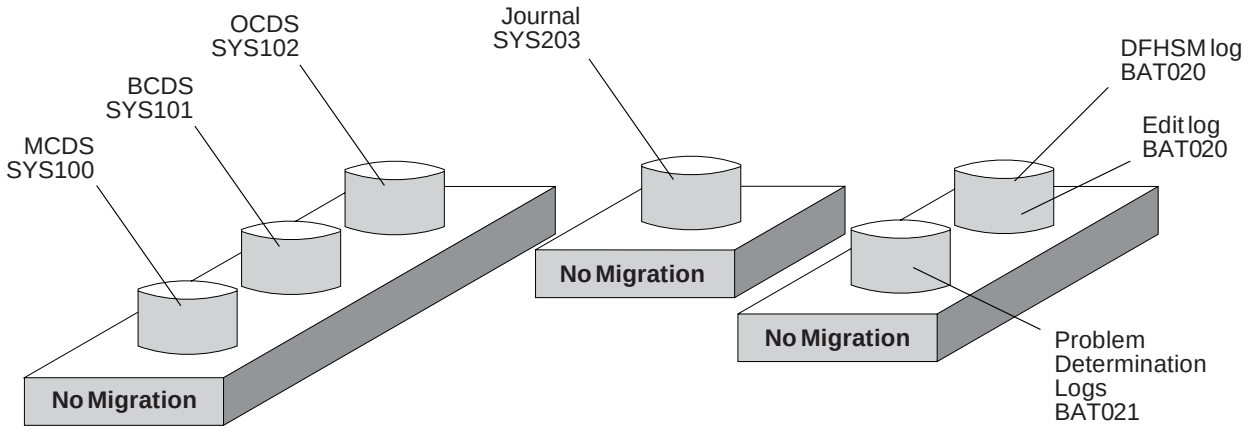

Figure 6. Overview of DFSMShsm Installation Environment for Control Data Sets, Journal, and Logs

DFSMShsm Data Sets

DFSMShsm uses the following control data sets, journal data set, and control data set backup copies:

- The migration control data set (MCDS) records activity to migration volumes. This data set is always required.
- The backup control data set (BCDS) is required to record activity for availability management.
- The offline control data set (OCDS) records tape activity, and is required if you are storing DFSMShsm-owned migration and backup data sets on tapes.

- The journal data set is used to recover the control data sets.
- Log data sets are used for recording DFSMShsm activity and logging errors for problem determination.
- The control data set backup copy data sets are optional but are highly recommended. They are combined with the journal data set to recover any control data set that becomes damaged.

For a planning example, see “Sample Plan for Allocating DFSMShsm Data Sets” on page 280.

Control Data Sets

The control data sets are VSAM key-sequenced data sets; consider the following recommendations when you allocate these data sets:

- If the control data sets are SMS-managed, assign a storage class with the GUARANTEED SPACE attribute so that the data sets can be placed manually.
- If possible, allocate the control data sets on volumes that have cache and DASD fast write capability.
- If possible, allocate the index and data components of each control data set on the same volume to eliminate lockouts.
- Be sure to allocate the control data set backup copies on volume strings different from the volume strings on which the active control data sets reside.
- Protect the control data sets with RACF to prevent any other program from modifying them.
- Do not allocate the control data sets on volumes that contain JES3 data sets or other system data sets. Otherwise, requests for JES3 data sets or system data sets can conflict with a DFSMShsm request for the control data sets and cause performance problems. A system lockout can occur in this situation.
- All control data sets must be allocated on mounted volumes or on permanently resident storage volumes. In a multi-processor environment, these volumes must be shared when DFSMShsm is active on all processors.

Journal Data Sets

To enable you to recover a lost or damaged control data set to the point of failure, DFSMShsm maintains a journal of each transaction that occurs to the control data sets. Should a control data set be damaged or lost, you can acquire a backup version of the control data set and merge the transactions recorded in the journal to bring the control data set back to current status.

When allocating journal data sets, consider selecting volumes with DASD fast write capability. Be sure to allocate the journal and journal backup copies on a volume string that does not contain any other control data sets.

Log Data Sets

The starter set or DFSMShsm itself allocates the required log data sets. Each system that is running DFSMShsm has its own set of log data sets, so that the log data sets need not share volumes with each other. The log data sets DFSMShsm uses are:

- DFSMShsm logs (LOGX and LOGY)
- Edit log

- Activity logs
- Problem determination aid logs (PDOX and PDOY).

Control Data Set Availability

DFSMShsm control data sets and journals provide record keeping for DFSMShsm activities. Backup copies of these data sets allow you to retrieve data should the active control data sets become lost or damaged.

DFSMShsm backs up the control data sets and journal each time automatic backup is performed on the primary processor. You can also back up the control data sets by command. In planning for control data set and journal availability, you must consider:

- If you will back up the CDSs in a concurrent copy session.

  Concurrent copy allows point-in-time copies while DFSMShsm is processing user data sets. To back up control data sets in a concurrent copy session, you must have the volumes connected to a IBM 3990 Storage Control with cache.

- How many versions of the control data sets you want to keep.

  You should keep several versions to ensure that you have at least one good backup version if you discover errors in the active data sets. If you use the DIAGNOSE command periodically, keep more backup versions than the number of days between DIAGNOSE commands.

- The media on which you want to store the backup versions.

  DASD makes it possible to recover the control data sets without tape mounts. On the other hand, tape backup versions require less space.

- What data mover you want to use to back up the control data sets.

  If you select DFSMShsm as the data mover, control data sets are backed up sequentially. If you want to use concurrent copy for backup, however, you must specify DFSMSdss as the data mover. DFSMSdss is recommended, as it performs integrity checking on the index component during backup, providing early warning of problems with the KSDS structure.

A sample plan for control data set maintenance is shown in “Sample Plan for Maintaining Control Data Sets” on page 281.

DFSMShsm-Owned Volumes

You must decide which volumes to assign to DFSMShsm as *owned* volumes. These volumes cannot be placed under SMS management, and users should not be allowed to allocate data sets on them. The volumes are used as ML1 volumes, and can also be used as backup or migration level 2 (ML2) volumes.

In selecting volumes to be owned by DFSMShsm, although you might use the same type of DASD volume for migrated and backed up data sets as those for your users’ data sets, remember that DFSMShsm stores each in about half the space that the data occupies in primary storage. Also, you do not need to use dual copy or DASD Fast Write, because DFSMShsm does not require these services.

You must select some DFSMShsm-owned DASD volumes as ML1 volumes. Even if you are not implementing space management as the first function, DFSMShsm requires ML1 volumes because:

- DFSMShsm stores VTOC copy data sets on ML1 volumes when it performs the backup and dump functions.
- When you issue a command to back up a data set, DFSMShsm stores the backup version on an ML1 volume until automatic backup runs the next time on the primary processor.

Planning for DFSMShsm-Managed Data Sets and Volumes

To use the services that DFSMShsm provides, you need to define the data sets or volumes that will be DFSMShsm-managed.

For SMS-managed volumes, DFSMShsm manages each data set on a volume according to the management class attributes associated with the data set. Additional parameters are set for volumes in a storage group to provide automatic dump, migration, and backup services, which use DFSMShsm functions.

For non-SMS-managed volumes, DFSMShsm manages each data set on a volume according to the attributes of the volume on which the data set resides.

Common data types for which you can use DFSMShsm services:

Application data sets and TSO data sets: Include space management, availability management, and ABARS functions; place in storage groups that provide automatic migration, backup, and dump services.
System data sets: Include availability management functions; exclude from space management.
Database data sets: Exclude from DFSMShsm-management. This recommendation is for the data only, and not the programs.

Setting Availability Management Policies

Complete availability management includes on-site data set and volume recoverability, and off-site recovery from disaster. On-site data set recoverability is the ability to recover individual data sets that are lost or corrupted. On-site volume recoverability lets you recover the data sets on a DASD volume that becomes unusable. Off-site disaster recovery lets you recover and continue operations if your computing center becomes inoperable.

On-Site Data Set Recoverability

On-site data set recoverability depends heavily on automatic incremental backup, although individual data sets can be recovered from dump copies. The most recent copy of an active data set is likely to be a backup version rather than a dump copy. Additionally, recovery of an incremental backup version requires tape mounts only for the backup volumes that contain the data set. Recovery from a dump copy requires all the tapes in the dump that contain the data set to be mounted from the first tape through the tape containing the data set.

For on-site data set recoverability, plan for each of these:

- Backup cycles
- Backup window
- Number of backup tasks
- Management classes and backup attributes

- Backup media
- Data set recovery
- Backup volume cleanup.

For more detail, see *DFSMS/MVS DFSMShsm Storage Administration Guide*.

Creating Backup Cycles

If you select automatic backup, DFSMShsm attempts to start it every day. However, you can choose a backup cycle of up to 31 days that causes automatic backup to start only on those days in the cycle that you designate. Thus, you can specify a daily backup excluding Sunday, every other day, twice a week, or any other schedule that you find appropriate. For quick recovery, run backup frequently.

Establishing the Backup Window

With DFSMShsm, you can specify start and stop times for automatic backup. You should select a time when DFSMShsm does not interfere with major production or interactive processing.

The backup window interacts with the number of backup tasks. Having more backup tasks running helps ensure that processing is completed within the window.

When you select a backup window, if backup starts in the primary processor, it immediately backs up the control data sets and no other DFSMShsm function can run. Therefore, start backup early enough so that it finishes backing up the control data sets before any other concurrent automatic function starts.

You can reduce the time that DFSMShsm functions are suspended during control data set backup by backing them up as concurrent copies.

Setting the Number of Backup Tasks

With DFSMShsm, you can run up to 15 backup tasks concurrently in a single processor. Doing this allows you to process more volumes in a given time. However, when you choose the number of tasks, consider the number of target devices available. Each task requires an output device on which it can store backup versions. There are other automatic functions that can run at the same time, and you must ensure adequate resources.

Selecting Backup Parameters

You can select both the number of backup versions that you keep for each data set and the number of days between backups. DFSMShsm allows you to maintain up to 100 backup versions of the data sets eligible for incremental backup.

For non-SMS-managed volumes, these specifications are system-wide. That is, for each system running DFSMShsm for backup, each backed up data set keeps the same number of backup versions and can be backed up only as often as any other data set backed up by that system.

For SMS-managed storage, the number of versions and frequency of backup are attributes in the management class. Sample management classes and descriptions of their effects are given in both the *DFSMS/MVS Implementing System-Managed Storage* and the *DFSMS/MVS DFSMShsm Storage Administration Guide*. Use these descriptions to decide what values you want to use in your own management classes.

Choosing Backup Media

You can back up data to DASD volumes, but it is better to back up data to tapes. Although backing up to DASD volumes eliminates the wait for tape mounts, tape backups are inexpensive, have greater storage ability, are easier to clean up. In addition, the duplex tape function can be used to simultaneously created an original backup tape and an alternate. The alternate can either be taken offsite, or written directly to a remote tape library.

Planning for Data Set Recovery

One advantage of DFSMShsm is that users can recover their own data sets by issuing the HRECOVER command. DFSMShsm searches its records, requests that the correct tape be mounted, and recovers the data set. Be sure that you provide education for users, and ensure that they have *DFSMS/MVS DFSMShsm Managing Your Own Data* available.

Planning for Backup Volume Cleanup

When you delete an outdated or invalid data set, the backup versions still exist. If you do not remove them, you will accumulate volumes full of backup versions that will never be recovered. You can use the DFSMShsm EXPIREBV command to expire backup versions of deleted and unchanged data sets. Run EXPIREBV on a regular basis to keep your backup volumes clean.

On-Site Volume Recoverability

The dump function of DFSMShsm allows you to recover a complete volume from one small group of tapes mounted in sequence. Forward recovery with the APPLYINCREMENTAL command then lets DFSMShsm bring the volume up to date with tape mounts just for those data sets that were changed since the volume was dumped.

The incremental backup data sets are scheduled to be processed by data set recovery tasks, rather than being processed by the volume recover task. Anywhere from one to 64 individual data set recovery tasks can be performed concurrently, as defined by the installation. This multitasking capability for individual data set recoveries can speed volume recoveries. If more than one volume is being recovered, the incremental data set recovery requests for one volume can be scheduled, and the recovery process can immediately begin on the next volume, lessening the time needed for the total recovery process.

In planning for the dump functions, you must consider the:

- Volume dump cycle
- Volume dump window
- Number of dump tasks
- Dump classes

Volume Dump Cycle

DFSMShsm attempts to start dump processing every day if automatic dump is specified. You can choose a dump cycle of up to 31 days, and you can select the days in the cycle when you want DFSMShsm to perform dump processing. Create the dump cycle and the number of times dump processing occurs based on how much data activity you want to process with forward recovery. That is, you should establish your dump cycle so that if a recovery of a volume is necessary, DFSMShsm will not require excessive tape mounting for forward recovery.

Volume Dump Window

With DFSMShsm, you can select start and stop times for automatic dump; choose a time period when DFSMShsm will not interfere with major production or interactive processing. You can specify that DFSMShsm dump processing run as long as possible in the available time. If it completes all volumes before the end of the time period, it ends, and the processing resources become available for any other task. If dump processing runs concurrently with backup processing, begin dump processing after backup has started to back up the control data sets.

The dump window interacts with the number of dump tasks. Increasing the number of dump tasks running helps to ensure that dump processing completes within the dump window.

Number of Dump Tasks

You can run up to 15 dump tasks concurrently in a single processor, increasing the number of volumes DFSMShsm processes at a given time. When you select the number of tasks, consider the number of output devices available: each task requires an output device for each concurrent dump copy that is created. The number of dump copies each task makes depends on the number of dump classes you select for the volume being processed. For extended performance, you can run multiple dump tasks in multiple processors.

Dump Classes

You can select up to five dump classes to manage how volumes are dumped. Dump classes control:

- When in the dump cycle the volume is dumped.
- Whether the tapes are automatically reused when the dumps expire.
- Whether users can recover their data sets from the dump copy.
- Instructions for the operator, such as to send the dump to off-site storage.
- When the data on the dump copy becomes invalid.

You might want to use multiple dump classes if you have volumes with differing requirements. For example, you might want to dump some volumes weekly and others biweekly. You might also want multiple dump classes if you want to send dump copies off site for use with off-site disaster recovery.

Planning Recovery for DFSMShsm-Owned Volumes

Even though there are backup copies of data sets stored on ML1 and ML2 volumes, you need to be able to recover those volumes if one is lost. You also want to replace a backup if it becomes damaged. DFSMShsm can make dump copies of ML1 volumes and can copy backup and ML2 tapes. For DFSMShsm to make copies of backup and ML2 tapes, the tapes to be copied from must be full, single-file tapes. Partial tapes are not automatically selected for copying, and multiple-file tapes cannot be copied. You should put ML1 volumes into dump classes that dump them on a regular schedule, and you should develop a schedule for copying backup and ML2 tapes.

Off-Site Recoverability

When you plan off-site disaster recovery, consider the types of data, the types of recovery for critical applications, and the recovery sites.

You probably have both critical and noncritical data and applications. The critical ones are those that must continue to run on schedule for your business to survive. Noncritical applications can generally wait for recovery until your site has returned to normal operations.

When you establish a plan for disaster recovery, you must consider:

- Using ABARS to recover critical applications
- Using ABARS to recover noncritical applications
- Security for ABARS backup and recovery
- Disaster recovery without ABARS
- Using the duplex tape function to concurrently create original and alternate tapes, with the capability of writing the alternate tapes directly to off-site locations.
- Being a recovery site for another site's applications

Recovering Critical Applications with ABARS

To use ABARS, you must decide which applications and data sets are critical to your business, and how you will transport backup information to the recovery site. When you have chosen the applications, you can use the aggregate backup assist utility tool to produce a selection data set that is used as input to the ABARS ABACKUP function.

You do not need either SMS or DFSMShsm active to define an aggregate group. Both must be active for performing the aggregate backup of the data sets. The data sets to be backed up need not be SMS-managed, but they must be cataloged in an integrated catalog.

You can send the backed-up data sets to the recovery site either by physically moving the tapes or by sending them by the NetView* file transfer program or an equivalent product.

Recovering Noncritical Applications with ABARS

You can plan to recover noncritical applications as you did for the critical applications, but you do not require a recovery site. The noncritical applications can be recovered after your main data center has returned to operation.

Planning for Disaster Recovery without ABARS

If you have uncataloged data sets or data sets cataloged in non-ICF catalogs, you cannot use ABARS for disaster recovery. Provide backup sets by:

- Making a standalone-recoverable copy of the operating system.
- Making extra dump copies of your primary volumes by dumping to more than one dump class.
- Making extra dump copies of your ML1 volumes.
- Using the duplex tape function to simultaneously create original and alternate tapes. The alternate tapes can be written directly to remote tape libraries.
- Using TAPECOPY to make copies of your backup and ML2 tapes.

- Backing up your control data sets or making copies of their backup versions.

You can then ship the tapes to a recovery site equipped with DASDs identical to those from which the dump copies were made. The dump process creates DFSMSdss physical dumps that require restoration to a device identical to that on which the dump copy was made.

Setting Space Management Policies

In planning for space management, you want to balance DASD space availability for data set allocations without frequently recalling migrated data sets. You also need to ensure that you can complete your space management processing in the time allotted. See “Sample Plan for Space Management” on page 283 for guidelines.

Management Class Migration and Expiration Parameters

Space management parameters, or *attributes* in SMS management classes, control how DFSMShsm makes space available on your primary volumes. In a non-SMS environment, the parameters apply to all the data sets on a volume. In an SMS environment, the management class attributes apply on a data set basis so that different data sets on a volume can be managed for different requirements. The parameters control:

- Whether unused space in sequential and partitioned data sets is released when a data set is closed, or is released when primary space management is performed
- When data sets expire
- How long a data set should remain on a primary volume if it is not opened
- How long a data set should remain (combined time) on the primary and level 1 volume if it is not opened
- The level to which a data set moves on its first migration

Sample management classes illustrating these concepts are shown in *DFSMS/MVS Implementing System-Managed Storage*. Sample parameters illustrating both SMS and non-SMS activities are described in *DFSMS/MVS DFSMShsm Storage Administration Guide*.

Automatic Primary Space Management Window

With DFSMShsm, you can choose start and stop times for automatic primary space management. Specify a time period when DFSMShsm cannot interfere with major production or interactive processing. Also require that automatic primary space management processing runs for as long as possible in the time available. If automatic primary space management completes processing before the end of the specified time, it ends.

When automatic backup starts on the primary processor, it first backs up the control data sets. No other function can start while the control data sets are being backed up. Therefore, start automatic primary space management some time after you start automatic backup.

The automatic primary space management window interacts with the number of automatic volume space management tasks. Increasing the number of automatic

volume space management tasks running helps ensure that the processing completes within the window.

Automatic Primary Space Management Cycle

With DFSMShsm, you can choose a cycle of up to 31 days and select the days that automatic primary space management does not run. It is best to run automatic primary space management every day unless you have a special reason for skipping a day in the cycle.

Number of Automatic Volume Space Management Tasks

With DFSMShsm, you can run up to 15 automatic volume space management tasks concurrently in a single processor. For a detailed discussion of volume migration task contention, see the *DFSMS/MVS DFSMShsm Storage Administration Guide*.

If you are migrating data sets from a primary volume directly to migration level 2 tape, you must have a tape device available for each automatic volume space management task plus at least one other drive available for recall tasks.

Other resources to consider are processor cycles, channel paths, and access to the control data sets and journal.

In planning the number of tasks for automatic volume space management, you must also consider the number of resources required by other DFSMShsm tasks that are running concurrently.

Interval Migration

Interval migration is the process of migrating data sets from level 0 volumes when a specified volume threshold level is met or exceeded during the day. To use interval migration, you must set high enough thresholds on primary volumes so the migration happens when there is unusually high activity. Set the low thresholds low enough so space is available for data set allocations throughout the day on ordinary days.

Automatic Secondary Space Management Window

You can select any time of day that you have tape devices available to run automatic secondary space management. Do not, however, run automatic secondary space management concurrently with automatic primary space management if you are migrating data sets to SDSP data sets.

Automatic Secondary Space Management Cycle

This cleans the migration control data set and moves aged data sets from ML1 to ML2. It also deletes all recalled and expired data sets from SDSP data sets. With DFSMShsm, you can choose a cycle of up to 31 days and what days automatic secondary space management should run. When you select the cycle, consider how frequently your ML1 volumes become full and how many data sets expire or are recalled.

Migration Level 1 Volumes

These volumes store data sets that migrate from primary storage or copies of VTOCs from volumes that are backed up or dumped. They can also store backup versions created by a data set backup command.

You should choose your highest capacity volumes for ML1 volumes. Prepare the volumes by removing them from SMS management.

Setting Thresholds

If you want automatic secondary space management to migrate aged data sets from ML1 volumes to ML2 volumes, you must set a threshold value for each ML1 volume. Secondary space management compares all the ML1 volume occupancies against the given levels; if the occupied space exceeds the threshold, all eligible ML1 data sets are moved to ML2.

When you set thresholds on ML1 volumes, you should maintain equal free space on all of the volumes. If you use different device types, calculate the percentages so that there will be an equal amount of space available on each.

For more information on thresholds, see see *DFSMS/MVS DFSMShsm Storage Administration Reference*.

Migration Level 2 Media

Although you can use DASD for ML2 volumes, we strongly recommend that you use tape, because:

- ML2 is for data you do not expect to use for a long time.
- Tape media is less expensive per MB than DASD.
- SMS only supports ML2 tape for its "Direct to ML2" support.

Small Data Set Packing

Small data set packing (SDSP) can be a great space saver on your ML1 volumes. With it, you can migrate data sets that occupy 400K or less to VSAM key sequenced small-data-set-packing data sets. Each migrated data set becomes one or more records of the SDSP data set. Thus, more user data sets can be packed into fewer tracks of the ML1 DASD. Figure 7 on page 269 illustrates this.

For correct operation you must have more SDSP data sets than you have automatic primary space management tasks. To use SDSP, you must do some manual operations, such as:

- Backing up the SDSP data sets regularly
- Reorganizing the SDSP data sets when they become full
- Recovering the data sets if their ML1 volumes are damaged.

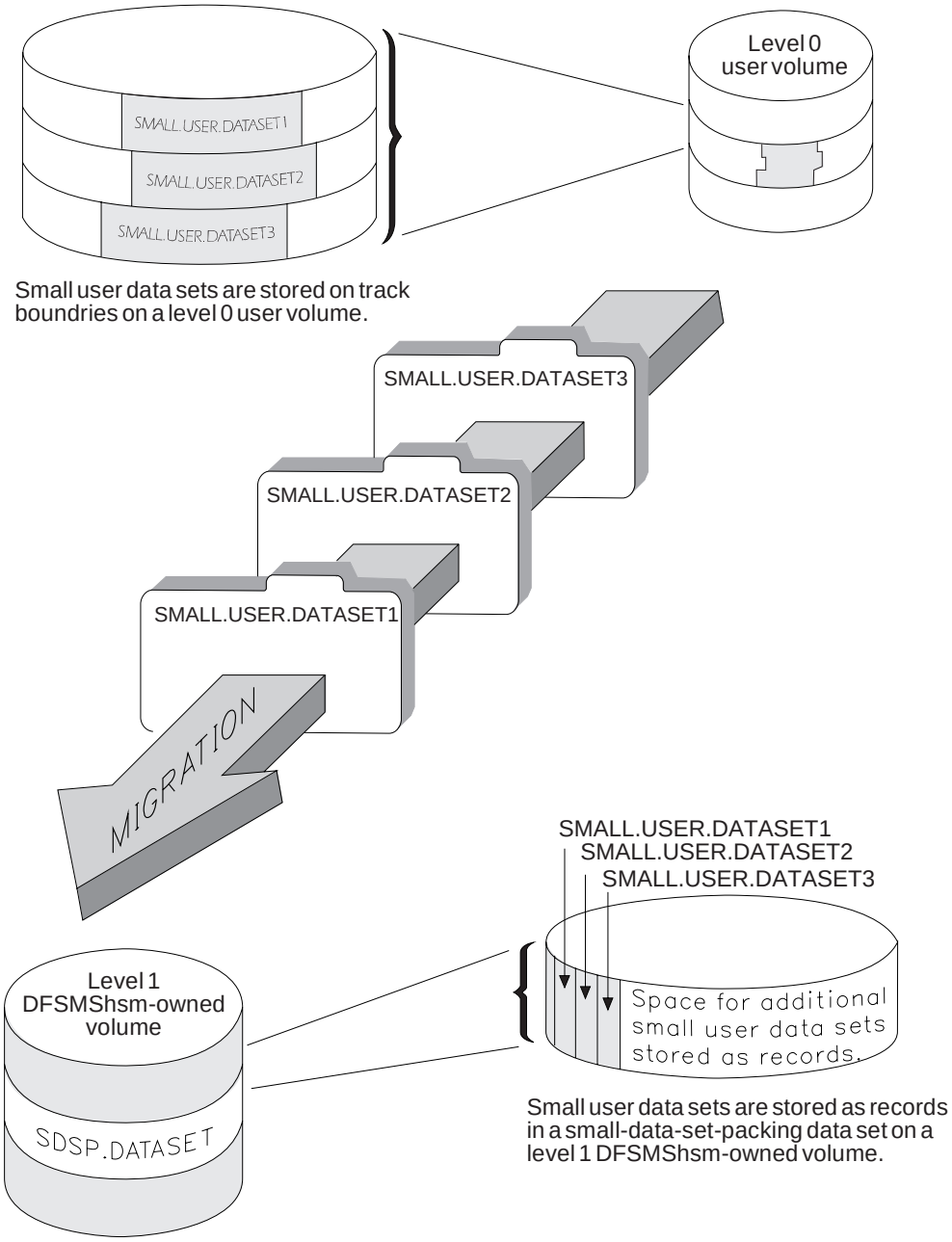

Figure 7. Small-Data-Set-Packing Data Set Overview

Backing Up SDSP Data Sets

Back up your SDSP data sets daily, preferably at the end of automatic primary space management. Frequent backups are recommended because of the large number of user data sets stored in an SDSP data set. You can back up SDSPs either by using the DFSMSdss DUMP command or by using access method services EXPORT commands. Individual data sets that the SDSP data set contains can be backed up with DFSMShsm BACKDS command.

Reorganizing SDSP Data Sets

Migration of user data sets into an SDSP causes VSAM control interval and control area splits. Erasing the VSAM records of recalled and deleted data sets in an SDSP makes space reusable. However, because redistribution of SDSP free space leaves gaps, the SDSP eventually appears full even though there could be some free space within it. When the high-allocated relative byte address equals the high-used relative byte address, the SDSP appears full to a migration request if the SDSP cannot be extended.

How often you reorganize SDSPs depends on the size and number of them and on the number of data sets that regularly migrate to SDSPs in your environment. Use the TSO LISTC command every week to determine the status of SDSPs as reported by the high-used and high-allocated relative byte addresses.

Use the access method services EXPORT and IMPORT commands to reorganize the data sets. Because the reorganization affects only DFSMShsm-owned data sets, you can schedule it for any time that will not interfere with recall requests.

Recovering SDSP Data Sets

How you recover an SDSP data set depends upon how you backed it up. If you use the DFSMSdss DUMP command for backup, you must use the DFSMSdss RESTORE command for recovery. If you use an access method services EXPORT command for backup, you must use an IMPORT command for recovery. If you use a DFSMShsm BACKDS command to backup individual data sets that the SDSP contains, you must use a DFSMShsm RECOVER command for recovery. See *DFSMS/MVS DFSMShsm Storage Administration Guide* for instructions on recovering SDSP data sets.

Pooling Volumes for Non-SMS-Managed Data Sets

To ensure that recalled non-SMS-managed data sets are recalled to the correct volumes, you can organize your non-SMS-managed volumes into pools. You can define either pools that accept for recall only specified data set names or pools that accept for recall any data set that migrated from them.

A non-SMS-managed volume that is automatically managed by DFSMShsm can be in any number of data set or volume pools. It can also be available for general recall use if it is online, permanently resident or reserved, and is specified as available for automatic recall.

In a multiple-processor environment:

- A data set or volume pool should be the same in all processors.
- Volumes that are available for general recall must be accessible to all processors.

Pools in Systems Running under JES2

In systems running under JES2, there are two kinds of pools—data set pools and volume pools—also known as user-defined pools. You can define them either during the startup procedure or after DFSMShsm has been started.

Pools in Systems Running under JES3

In systems running under JES3, there are three kinds of pools. In addition to the user-defined pools, DFSMShsm configures a general pool. This is a collection of all automatically managed volumes that are:

- Online
- Permanently resident or reserved
- Specified for automatic recall
- Mounted with the use attribute of storage or private

Setting Up Tape Management for DFSMShsm

A tape library is a named collection of storage groups, tape devices, tape cartridges, and a tape library dataserver. Tape libraries are defined by a combination of SMS constructs and DFSMShsm commands. You can process any DFSMShsm tape function in an SMS-managed tape library.

Individual tape cartridges can also be managed with a tape management system, such as DFSMSrmm. For more details on DFSMSrmm, see “Preparing to Implement DFSMSrmm” on page 247.

Whether or not you are managing tapes in a tape library, you must clearly define the way a tape is managed throughout its life cycle. Figure 8 on page 272 illustrates this cycle, from its initial entry into the scratch pool to its recycling and return to the scratch pool.

This section will help you to further define your DFSMShsm tape management policies by explaining scratch pools, output tapes, security, and recycling.

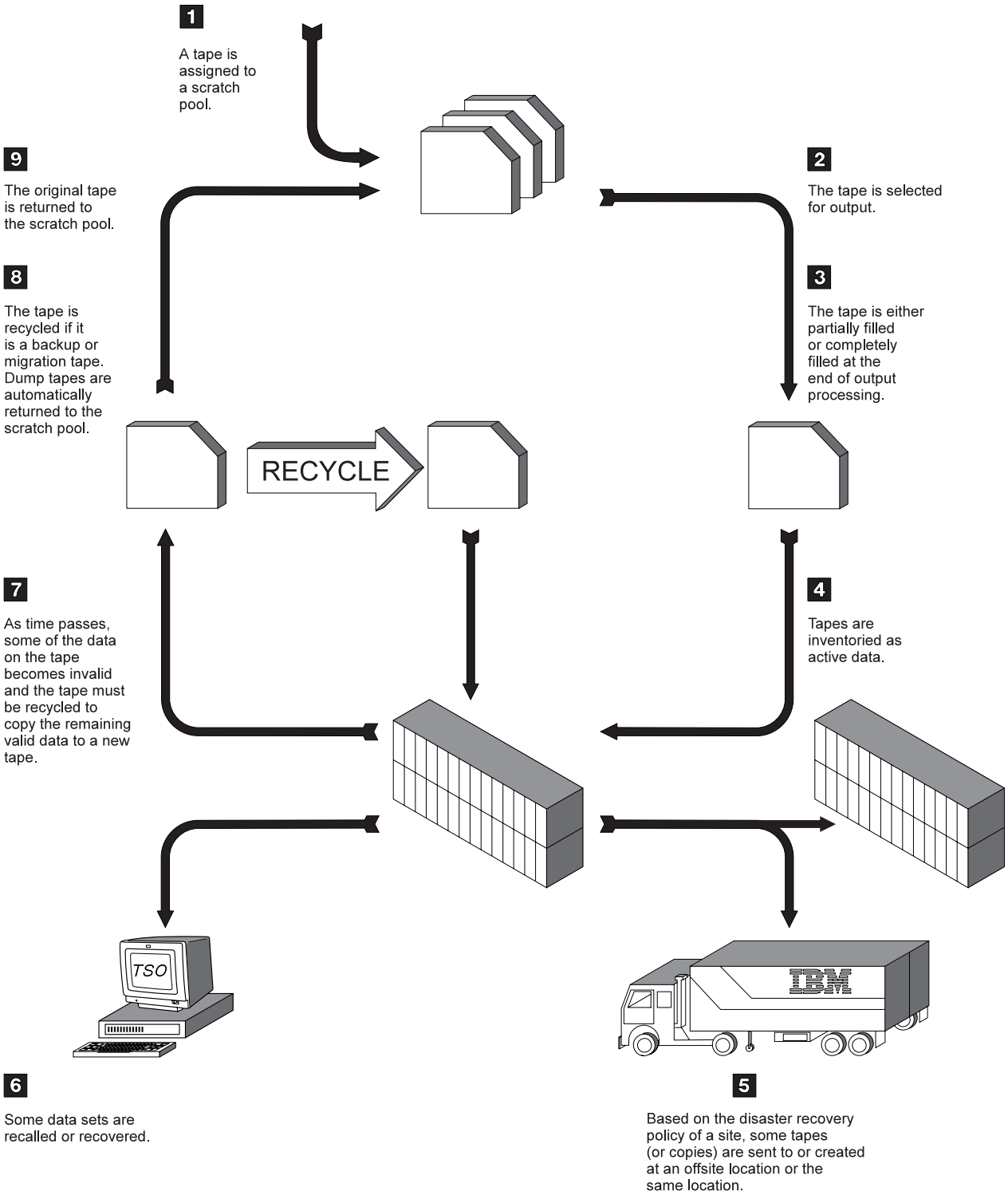

Figure 8. Life Cycle of a Tape

Scratch Pool Environment

There are two types of scratch pools:

- Global scratch pools, which we recommend for both SMS-managed tape libraries and for non-SMS tape environments that make full use of cartridge loaders. In this environment, all tapes in the installation are managed by a tape management system.
- In the *specific* scratch pool environment, DFSMShsm keeps a record of all of its tapes, both those containing valid data and those that are empty. When tapes no longer contain valid data, DFSMShsm returns them to the specific scratch pool.

Output Tape Selection

DFSMShsm selects tapes for output processing depending upon the scratch pool you have defined and the way you direct DFSMShsm to mark tapes that are only partially full. Global scratch pools offer an advantage because all scratch tapes are candidates for output selection. In a specific scratch pool, each scratch tape is specifically known to DFSMShsm and only one tape is a candidate for output selection. Marking tapes full allows DFSMShsm to always select a fresh scratch tape for tape output processing; marking them partially full allows DFSMShsm to fully use all the tape media.

Recycling Tapes

You should decide how often to run the recycling process, and how little valid data a tape can contain before it is recycled. You can either set a regular schedule for recycling, or you can wait until more tapes are needed. Because recycle can be run independently for ML2 tapes and for backup tapes, the schedules can be independent of each other. Because recycle runs only on DFSMShsm-owned tapes, you can schedule it for whenever your tape drives are available. Figure 9 on page 274 illustrates the concept of recycle processing.

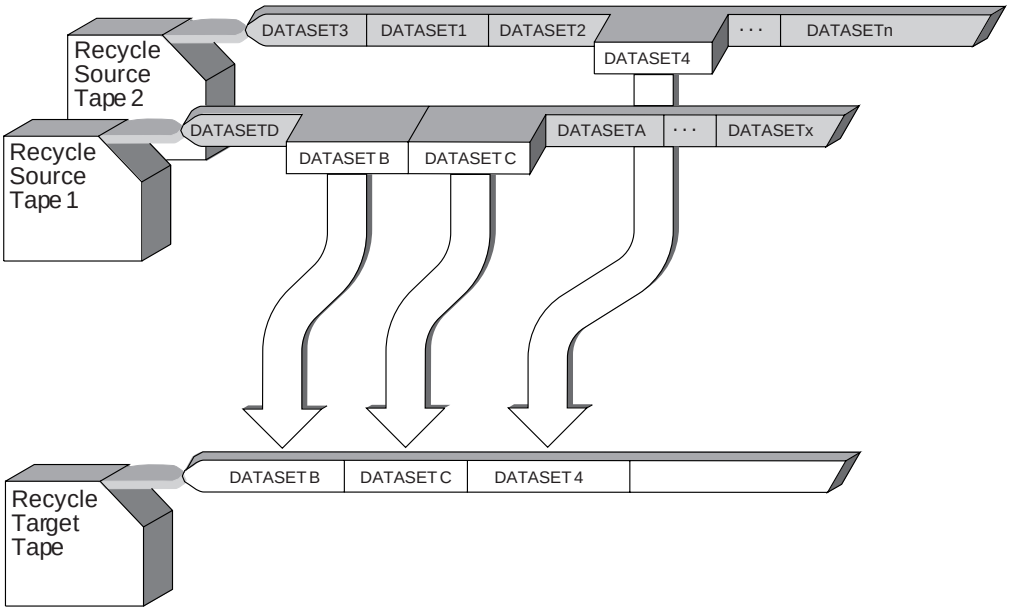

Figure 9. Tape Efficiency Through Recycle Processing. As a result of recycle processing, valid data sets from source tape 1 are copied to the recycle target tape. After all valid data sets from source tape 1 are copied, valid data sets from source tape 2 are copied to the recycle target tape. Source tapes that have been recycled are returned to the scratch pool.

Device Selection

When you choose tape devices for DFSMShsm functions, you can maximize the use of high-performance devices by selecting them for your heaviest tape processing loads. You maximize the use of older devices by using them for tape input processing that always requests a specific tape—requiring operator intervention.

If you manage tapes in a tape library, high-performance devices (3490, 3490E, and 3590-1) are selected because of their association with the tape library. If you do not use a tape library, high-performance devices can still be selected when you group them together by an esoteric name that associates the devices with the DFSMShsm functions you choose.

Tape Operations

To optimize the performance of your tape environment, you can choose automated tape processing, or you can use operator intervention.

Fully unattended operation

This is for use in SMS-managed tape library environments, and offers the highest level of tape processing automation. It needs no human intervention.

Partially unattended operation

This is for use in non-SMS-managed tape environments, and uses cartridge loaders. It applies only to tape output processing because output processing requests nonspecific scratch tapes. Input processing always requests a mount for a specific tape volume, thus eliminating the possibility of automating the process with cartridge loaders.

You can run partially unattended operations with either a global scratch pool or a specific scratch pool. Partially unattended operation for migration or backup can be either:

Initial selection

- In a global scratch pool

  The operator fills the cartridge loader with scratch tapes before DFSMShsm selects a tape device and a tape for the task. If DFSMShsm finds an empty or partially filled tape available, it selects one as the initial tape mounted for that task.

- In a specific scratch pool

  The operator creates a “pull list” of known output tapes and loads them into the cartridge loader.

Subsequent selection

- In a global scratch pool

  DFSMShsm mounts the next scratch tape from the cartridge loader with no operator intervention.

- In a specific scratch pool

  DFSMShsm takes the next scratch tape from the cartridge loader with no operator intervention (the operator has loaded scratch tapes that were identified with the pull list into the cartridge loader).

You can schedule dump and incremental backup to occur when the system is unattended, and space management functions when an operator is present. To do this:

- Ensure that at least one tape drive is available for each task that could require one. For example, if you have three dump tasks running, each requires a tape drive.
- Remember that recalls from ML2 tape volumes and recovers from dump tapes or incremental backup tapes will fail. If you know that a job requiring a data set that is migrated to tape will run when the system is unattended, you must recall that data set before the unattended operation starts.
- Ensure that each task requests a scratch tape. You must use the PARTIALTAPE(MARKFULL) parameter of the SETSYS command to prevent having partially written tapes.

Testing DFSMShsm

Three facilities are used to ensure that DFSMShsm is installed correctly and can function in your data center. See *DFSMS/MVS DFSMShsm Implementation and Customization Guide* for detailed instructions and examples.

- Run the *installation verification procedure (IVP)* to ensure that DFSMShsm is correctly installed. Run it after you finish the SMP/E installation of the DFSMShsm modules into the appropriate system library data sets.
- Run the *functional verification procedure (FVP)* to test and verify major DFSMShsm functions such as backup, dump, migrate, recall, recovery, and restore. The FVP creates test data sets and exercises basic DFSMShsm processing.

- Run the *starter set* to establish a functional, sample DFSMShsm environment. Creating the starter set allows you to test DFSMShsm before you put it into a production environment. Running it creates a parameter library startup member, which can be expanded and tested as you implement additional DFSMShsm functions.

Customizing and Tuning DFSMShsm

DFSMShsm provides many installation exits to allow you to customize DFSMShsm processing at your site. They can give you additional control (for example, how certain kinds of data sets are processed). A sample plan for customizing and tuning is shown in “Sample Plan for Customizing and Tuning DFSMShsm” on page 285. For more information about the DFSMShsm exits, see *DFSMS/MVS Installation Exits*.

You can tune your system temporarily by issuing specific PATCH commands after DFSMShsm has been started, or you can do it permanently by including the commands in the DFSMShsm startup PARMLIB member. Tuning options include:

- Allowing password-protected generation data sets to be migrated and scratched without checking the password at roll-off time
- Scratching generation data sets at roll-off time, regardless of their expiration dates
- Using DFSMShsm in a JES3 environment that performs main device scheduling only for tapes
- Preventing migration or altering the prevent-migration activity for JES3 setups
- Replacing HSMACT as the high-level qualifier for activity logs
- Increasing the I/O buffers for recalls of data sets that migrated with optimum DASD blocking
- Using the DFSMSdss compress option for DFSMShsm dumps
- Resetting the wait time for the start up of the ABARS secondary address space
- Enabling ABARS ABACKUP and ARECOVER to wait for a tape unit allocation
- Changing the RACF FACILITY class ID for the console operator's terminal
- Handling original equipment manufacturer data in the data set VTOC entry
- Allowing DFSMShsm automatic functions to process volumes other than once per day
- Changing the default tape data set names that DFSMShsm gives to tapecopy and full volume dump tapes
- Preventing interactive TSO users from being placed in a wait state during a data set recall
- Changing the default number of recall requests for a data set residing on a volume in use by recycle or tapecopy
- Changing the default number of buffers DFSMShsm uses to recall data sets from ML1 volumes

There are additional options for monitoring and tuning DFSMShsm functions with the DFSMS Optimizer. See the*DFSMS Optimizer User's Guide and Reference* for more information.

Coordinating DFSMShsm Processing

This section shows you how to coordinate automatic processing of DFSMShsm functions. Figure 10 on page 278 graphically demonstrates this coordination, while the sample plans in Table 18 on page 282, and Table 19 on page 284 describe planning tasks used to automate the functions shown in this illustration.

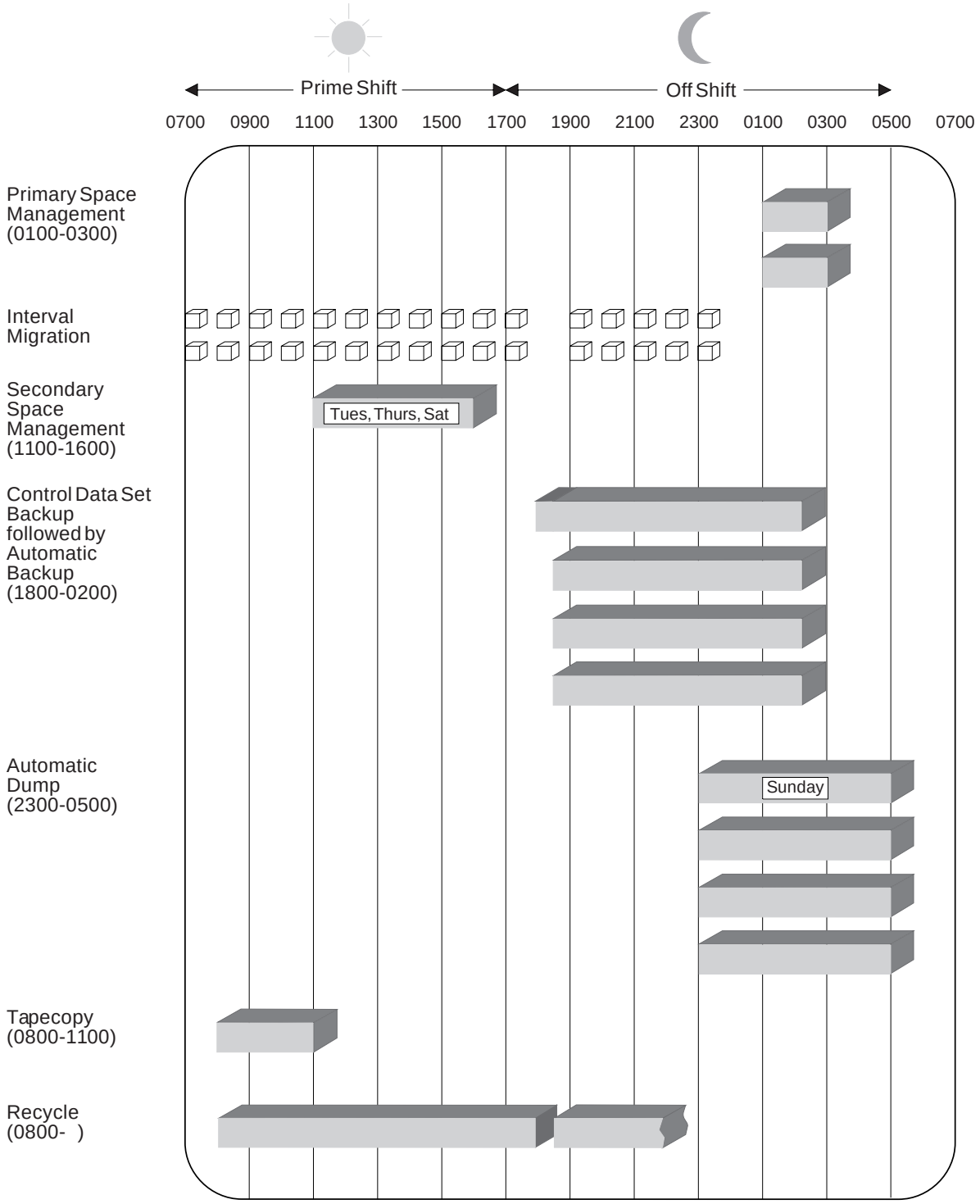

Figure 10. Coordinating DFSMShsm Processing Functions

Note the following:

- Interval migration, which runs hourly if any volumes meet or exceed their specified threshold, runs throughout the day and is only suspended for control data set backup and primary space management processing.

- Secondary space management runs during normal business hours because movement from level 1 volumes to level 2 volumes does not affect primary user volumes. Notice that secondary space management runs during a period of lower system use.
- Tape copy processing runs after the functions that write to output tapes have completed their processing. Tape copy processing begins at the start of the prime shift when tape operators are present to mount tapes.
- Backup is the most critical function to place in the time spectrum and is generally a good function to start with.
- Primary space management runs during periods of low user activity because, as DFSMShsm moves data from primary to level 1 volumes, it needs exclusive access to the users' volumes. Because DFSMShsm data ages at midnight, primary space management runs soon after.
- Recycle processing is done when operators and tape drives are available.
- Control data set backup occurs as the first step of automatic backup processing and occurs only on the primary processor. Interval migration and recycle processing (and any other functions that run concurrently with CDS backup) are suspended while DFSMShsm backs up its control data sets.

Sample Plans

The sample plans in this section can assist you in developing your installation's plans for:

- Setting implementation objectives
- Allocating DFSMShsm data sets
- Maintaining DFSMShsm control data sets
- Setting policies for availability management
- Setting policies for space management
- Customizing and Tuning DFSMShsm.

Sample Setup Plan

The sample plan in Table 15 on page 280 provides you with a framework for setting up your own objectives before you implement DFSMShsm. It includes RACF security and the Global Resource Serialization program. It uses three processors and five storage groups:

Storage group 1	Batch programs and data sets
Storage group 2	TSO programs and data sets
Storage group 3	System data sets and database program data sets
Storage group 4	Data sets that are the database data
Storage group 5	Programs and data sets for a special development project

Table 15. Example of a Setup Plan

Task	Subtask
Determine functions to be implemented	**Availability Management** Volumes to be managed: Storage groups 1, 2, 3, and 5 Implementation schedule: Begin as soon as SMP/E installation is complete. Have all storage groups managed in four weeks. Processor performing dump processing: 1 Processor performing backup processing: 1 **Aggregate Backup and Recovery** Aggregates to be managed: Critical aggregates: payroll, accounts receivable, order processing, manufacturing control system Noncritical aggregates: special development program, application development Implementation schedule: Begin two weeks after completion of availability management. Complete in four months. **Space Management** Volumes to be managed: Storage groups 1, 2, and 5 Implementation schedule: Begin two weeks after completion of ABARS. Complete in two months. Processor performing automatic primary space management: 2 Processor performing automatic interval migration: 2 Processor performing automatic secondary space management: 3
Determine DFSMShsm-owned volumes	DASD volumes to be used for DFSMShsm: 1 volume each from storage groups 1 and 2 to be used as ML1 volumes for backup and dump VTOC data sets. Add more volumes as needed for migration.

Sample Plan for Allocating DFSMShsm Data Sets

In this sample plan, the system setup is as follows:

- The index and data component of each control data set is on the same volume.
- The sizes of the control data sets are as follows (calculated as illustrated in *DFSMS/MVS DFSMShsm Implementation and Customization Guide*):

MCDS	105 cylinders on an IBM 3390
BCDS	324 cylinders on an IBM 3390
OCDS	17 cylinders on an IBM 3390
Journal	25 cylinders on an IBM 3390

- Control data sets are allocated on different volumes in storage group 3 and conform to the recommendations in “DFSMShsm Data Sets” on page 258. Allocating the control data sets on different volumes reduces contention.
- Control data set backup copies are stored on tape; therefore, you do not need to preallocate files for backup processing.
- Log data sets are allocated on volumes in storage group 1. The management class with which they are associated prevents them from migrating.

Table 16 on page 281 and Figure 6 on page 258 show a sample of the planning in this section.

Table 16. Sample Plan for Installing DFSMShsm and Allocating DFSMShsm Data Sets

Task	Attributes
Define software environment	Operating system: MVS/ESA SP Version 5 Storage control system: DFSMS/MVS Version 1 Job entry subsystem: MVS/ESA SP-JES2 Version 5 Data mover system: DFSMSdss Security system: RACF Version 1 Release 9.0 Resource serialization: GRS Installation facility: SMP/E Release 6
Implement panel support	ISMF
Determine volumes for the control data sets, journal, and logs	MCDS: SYS100 BCDS: SYS101 OCDS: SYS102 Journal: SYS203 DFSMShsm log: BAT020 Edit log: BAT020 Problem determination logs: BAT021 Activity logs: As determined by the storage groups with which they are associated.
Test DFSMShsm	Implement DFSMShsm on a single-processor test system before implementing DFSMShsm in a production environment.

Sample Plan for Maintaining Control Data Sets

Table 17 shows the sample plan for maintaining the DFSMShsm control data set backup environment.

Table 17. Sample Plan for Defining the DFSMShsm Control Data Set Maintenance Environment

Task	Subtask
Backing up control data sets in a concurrent copy session	Ensure that you have concurrent copy-capable IBM 3990 storage controls for DASD strings containing the control data sets.
Deciding how many versions of control data sets to keep	Keep four backup versions of each control data set and the journal.
Determining what media to use for backup versions of the control data sets	Back up to tape. Use the ARCCBEXT exit to move the versions of the BCDS, MCDS, and OCDS for one backup onto one tape. Use a separate tape for concentrating the journal backup versions.
Deciding on a data mover for control data set backup	Use DFSMSdss as the data mover for control data set backup in a concurrent copy session.

Sample Plan for Availability Management

The following sample plan is based on these assumptions:

- Daily backup for both on-site data set and volume recovery
- Weekly dump for on-site volume recovery
- Early evening backup and late dump to avoid dumping volumes that have not yet been backed up
- Eight tape drives available for backup and dump processing
- Four management classes are defined
- Users can recover their own data sets
- TAPECOPY used to recover DFSMShsm-owned volumes on site
- ABARS is used to provide for disaster backup

Table 18 (Page 1 of 2). Sample Plan for Defining the DFSMShsm Availability Management Environment

Task	Parameters or attributes
Select backup parameters	Backup window: 1800 to 0200 Backup cycle: Run every day. Backup tasks: 4 System to run backup: 1 Backup volume cleanup: Every other Friday from 0800 to 1300 Backup parameters of management classes For standard data sets • Backup frequency: 0 • Number of backup versions for existing data sets: 2 • Number of backup versions for deleted data sets: 1 • Retain days only backup version of deleted data sets: 0 • Retain days extra backup versions: 30 For generation data sets • Backup frequency: 1 • Number of backup versions for existing data sets: 2 • Number of backup versions for deleted data sets: 0 • Retain days only backup version of deleted data sets: 0 • Retain days extra backup versions: ___ For large data sets • Backup frequency: 0 • Number of backup versions for existing data sets: 2 • Number of backup versions for deleted data sets: 1 • Retain days only backup version of deleted data sets: 0 • Retain days extra backup versions: 30 For data sets that are not to be migrated • Backup frequency: 0 • Number of backup versions for existing data sets: 2 • Number of backup versions for deleted data sets: 1 • Retain days only backup version of deleted data sets: 0 • Retain days extra backup versions: 30

Table 18 (Page 2 of 2). Sample Plan for Defining the DFSMShsm Availability Management Environment

Task	Parameters or attributes
Select dump parameters	Dump window: 2300 to 0500 Dump cycle: Run every Sunday. Dump tasks: 4 System to run dump: 2 Dump classes Dump class ONEWEEK: • Reuse the tapes automatically when the dump copy becomes invalid. • Allow the users to recover data sets from the dump copy. • Dump the volumes to this class with 7 days between each dump. • Do not reset the data-set-changed bit. • Retain these dump copies for 21 days. Dump class TWOWEEK: • Do not reuse the tapes automatically when the dump copy becomes invalid. • Do not allow users to recover data sets from this dump copy. • Dump the volumes to this class with 14 days between each dump. • Do not reset the data-set-changed bit. • Retain these dump copies for 42 days. • Send the tapes from this dump class off site. Recoverability for DFSMShsm-owned volumes: • Dump ML1 volumes to dump classes ONEWEEK and TWOWEEK. • Run tapecopy function daily from 1300 to 1600.
Select the disaster recovery parameters	Recovery site: Sister site at Dallas for critical applications Critical applications: Payroll, Order Processing, Accounts Receivable Critical applications to be backed up after each run. Noncritical applications: Special Development, Application Program Development Noncritical applications to be backed up on Wednesdays and Saturdays. Use facility class to authorize the appropriate users to critical applications. Use facility class to authorize all DFSMShsm-authorized users.

Sample Plan for Space Management

This is a sample plan for you to use when you define DFSMShsm space management. The following parameters are assumed:

- Automatic primary space management occurs every day.
- Automatic primary space management provides 25% of the capacity of the volumes used for batch processing and 20% for interactive processing and for the special development project.
- If a volume becomes 95% full during the day, it becomes eligible for interval migration.
- Automatic secondary space management occurs three times a week.
- Automatic secondary space management runs during prime shift.
- Six tape drives are available for use for automatic primary space management.
- Four management classes are defined.

- ML2 volumes will be tape.
- Small data set packing is used.
- There are no non-SMS-managed primary volumes.

Table 19 (Page 1 of 2). Sample Plan for Defining the DFSMShsm Space Management Environment

Task	Parameters or attributes
Select automatic primary space management parameters	Automatic primary space management window: 0001 to 0300 Automatic primary space management cycle: Run every day. Automatic primary space management tasks: 2 System to run automatic primary space management: 3 Primary volume thresholds: Batch volumes • High threshold: 95% • Low threshold: 75% Interactive processing volumes and special development volumes • High threshold: 95% • Low threshold: 80%
Determine space management attributes of management classes	For standard data sets: • Expire after days nonusage: 395 • Expire after date or days: No limit • Release unused allocated space from eligible non-VSAM data sets: I • Primary days nonusage: 15 • Level 1 days nonusage: 60 For generation data sets: • Expire after days nonusage: No limit • Expire after date or days: No limit • Release unused allocated space from eligible non-VSAM data sets: I • Primary days nonusage: 15 • Level 1 days nonusage: 0 • GDGs on primary: 2 • Rolled off GDS action: Expire For large data sets: • Expire after days nonusage: 760 • Expire after date or days: No limit • Release unused allocated space from eligible non-VSAM data sets: I • Primary days nonusage: 15 • Level 1 days nonusage: 0 For data sets that do not migrate: • Expire after days nonusage: 395 • Expire after date or days: No limit • Release unused allocated space from eligible non-VSAM data sets: No • Primary days nonusage: ___ • Level 1 days nonusage: ___
Select automatic secondary space management parameters	Automatic secondary space management window: 1000 to 1500 Automatic secondary space management cycle: Run on Tuesday, Thursday, and Saturday. ML2: 3480 XF tapes.

Table 19 (Page 2 of 2). Sample Plan for Defining the DFSMShsm Space Management Environment

Task	Parameters or attributes
Select the SDSP parameters	ML1 volumes: SDSP data set on each SDSP data set sizes: 100 cylinders Sizes of data sets to be stored in SDSP data sets: 200K bytes Backup of SDSP data sets: At the end of automatic primary space management Reorganization of SDSP data sets: Saturdays from 0800 until finished.
Determine non-SMS-managed volume attributes	System to process: ___ Method of management: ___ Thresholds: ___ Recall attributes: ___

Sample Plan for Customizing and Tuning DFSMShsm

Table 20 assumes that:

- Sequential and partitioned data sets are reblocked during recall and recovery.
- A tape management program is in use in the computing center.
- DFSMShsm indicates when primary volumes need to be defragmented.
- ABARS is used to send data sets off site for disaster recovery.
- This data center will be the disaster recovery site for another data center in the company.

Table 20 shows the formal statement of our customizing planning.

Table 20. Sample Plan for Customizing and Tuning DFSMShsm

Task	Subtask
Determine exits needed	Reblock exit Tape volume exit Space management volume exit ABARS backup error exit ABARS migration level 2 data set exit ABARS data set skip exit ABARS conflict resolution exit.
Normal operating environment customizing with DFSMShsm supported patches	

Table 21. Sample Plan for DFSMShsm Tape Management

Task	Subtask
Select the tape environment	Recycle backup and ML2 tapes on Friday starting at 1500.

Securing DFSMS/MVS Resources

The Resource Access Control Facility (RACF) controls access to data processing resources such as DASD volumes, tape volumes, and data sets. All DFSMS/MVS functional components support RACF protection. See “Products Recommended for DFSMS/MVS Full-Function Support” on page 14 for the level of RACF you should run with DFSMS/MVS. Consult the following references for more information on RACF:

- *OS/390 Security Server (RACF) Introduction*
- *OS/390 Security Server (RACF) General User's Guide*
- *MVS/ESA SML: Managing Data*.

Using RACF Profiles

You should use RACF protection, rather than data set password protection, because RACF protection is easier to use and offers greater system resource security. In addition, SMS ignores data set passwords. If RACF protection is applied to a data set that is already password protected, password protection is bypassed and access is controlled solely through RACF.

RACF retains information about users, resources, and access authorities in profiles stored in a special RACF database:

Generic profiles make it easy for you to protect multiple data sets with similar security requirements, without having to specify the requirements for each individual data set.

Discrete profiles contain security information about a single data set or other type of resource.

Generic profiles contain security information about one or more data sets or other resources that have similar characteristics and therefore require a similar level of protection.

SMS, DADSM, open/close/end-of-volume, checkpoint/restart, IEHMOVE, and the integrated catalog facility all automatically check data sets on DASD for RACF generic or discrete profiles, to verify access authority. This is referred to as *RACF always-call*, because RACF is always called by these functions to verify access to a data set.

For VSAM data sets, DFSMS/MVS checks only the cluster's profile for authorization to access any of the cluster's components. Although you can have a discrete profile defined for each of the data and index components of a data set, DFSMS/MVS does not check these profiles. If the cluster does not have its own profile, you must consider the individual VSAM components unprotected.

© Copyright IBM Corp. 1979, 1999

Protecting ISMF Functions

The storage administrator can use the RACF program control facility to establish authorization levels for all ISMF functions or for individual applications, functions, line operators, and commands. A message informs users who are denied access because of insufficient authority. RACF also allows the storage administrator to control the use of storage and management classes. See “Library Access Control” on page 214 of this book for more installation-related details and *DFSMS/MVS DFSMSdfp Storage Administration Reference* for more information on using RACF services to protect ISMF functions.

Protecting Data Management Functions

The *erase-on-scratch* operation can be controlled by RACF options and data set profiles for non-VSAM data sets and for VSAM data sets cataloged in integrated catalogs. Erase-on-scratch prevents unauthorized access to sensitive DASD data by automatically erasing a data set when it is scratched.

Open, close, end-of-volume uses RACF to protect tape data sets, unlabeled tapes, and tapes using bypass label processing. This support provides protection for both DASD and tape data sets under the same generic profile.

By defining the appropriate RACF FACILITY class profiles, you can protect certain IDCAMS commands and keywords, and catalog functions, from unauthorized use. For more information on the RACF FACILITY classes you can define, see *DFSMS/MVS DFSMSdfp Storage Administration Reference*.

Protecting VSAM Record-level Sharing Functions

Applications, such as CICS, using the VSAM record-level sharing (RLS) commit protocol interface, are capable of releasing locks that protect uncommitted RLS updates. You need to restrict the use of this interface through the RACF facility class SUBSYSNM. Use the SUBSYSNM class to authorize the CICS subsystems that are allowed to use the VSAM RLS commit protocols. Read access is required.

The IDCAMS SHCDS command provides recovery functions that can also release locks protecting uncommitted RLS updates. Its functions result in the special handling of transaction backouts. Use the facility class STGADMIN.IGWSHCDS.REPAIR to protect these SHCDS functions. Refer to *DFSMS/MVS Access Method Services for ICF* for further details.

Protecting DFSMSrmm Functions

DFSMSrmm uses the System Authorization Facility (SAF) to perform authorization checking for DFSMSrmm resources defined in the RACF FACILITY class. DFSMSrmm optionally uses RACF to maintain TAPEVOL access lists.

DFSMSrmm provides automatic security classification through installation-specified criteria based on data set names. See “Planning DFSMSrmm Authorization and Security” on page 251 for more information.

Protecting DFSMSdss Functions

You can use RACF to protect resources that DFSMSdss operates on, such as DASD volumes, tape volumes, or data sets, against unauthorized access. You can also use RACF to limit the use of certain DFSMSdss functions to privileged users.

Data and Volume Access

DFSMSdss first invokes RACF to check the user's access authority to a volume. DFSMSdss uses RACF to check at the data set level only if:

- The user does not have the required volume level authority.
- The volume is not protected.
- DFSMSdss cannot determine the protection status of the volume.

When a data set is protected by RACF, DFSMSdss ignores the data set password.

When copying or restoring data, DFSMSdss ensures that users have sufficient authority to create or overlay the target data sets. In addition, DFSMSdss tries to protect the target data sets with the same RACF protection as the source data sets.

DFSMSdss also supports the DASD erase-on-scratch attributes defined in the RACF profile of the data set.

DFSMSdss Commands and Keywords

By defining the appropriate RACF FACILITY class profile, you can protect certain DFSMSdss commands and keywords against unauthorized use. DFSMSdss invokes RACF checking to ensure that users have sufficient authority to perform the function. For more information on the FACILITY class profiles used by DFSMSdss, see *DFSMS/MVS DFSMSdss Storage Administration Guide*.

Protecting DFSMShsm Functions

Securing DFSMShsm-owned and managed data sets and volumes ensures that unauthorized users cannot access these resources while they are under DFSMShsm's control.

Data Set Access

DFSMShsm calls the system authorization facility (SAF) to check for authorization when a data set is accessed. With a RACF profile defined, the data is protected from access under the data set name and also under the name that DFSMShsm assigns to the data set when it is placed in DFSMShsm-owned storage. DFSMShsm also honors password protection; however, SMS does not accept passwords for system-managed data sets.

Tape Access

RACF can also protect your DFSMShsm-owned tapes. If you are using a tape management product, you might need to use expiration date protection for your tapes.

Before you can reuse a tape that is returning to the global scratch pool, you must remove its RACF protection. DFSMShsm does this for tapes it has added to the RACF tape volume resource, but it cannot remove the protection for tapes that have been added to the RACF tape volume resource by the RACF administrator.

DFSMShsm cannot remove password protection; only a user with storage administrator authority can do this after the password-protected tapes are returned to the global scratch pool.

DFSMShsm Commands

Protecting the DFSMShsm processing environment includes limiting the number of users who can issue DFSMShsm commands. DFSMShsm commands are divided into *authorized commands*, which control operations against any data set accessible to DFSMShsm; and *unauthorized commands*, which control operations against only those data sets for which an issuer has the appropriate security authority.

One system user is defined as the control storage administrator. The control storage administrator can issue authorized commands and authorize other storage administrators for these commands.

DFSMShsm also allows you to control how it detects whether a user is authorized to submit DFSMShsm commands in a batch environment. If you have RACF, DFSMShsm can check the user ID in the RACF access control environment element (ACEE). If you do not have RACF, DFSMShsm can check the user ID in the TSO-protected step control block (PSCB).

Appendixes

© Copyright IBM Corp. 1979, 1999

Appendix A. DFSMS/MVS Library and Related Publications

The tables included here describe information sources that can help you understand and use the functions of DFSMS/MVS. The tables are organized as follows:

DFSMS/MVS Library provides comprehensive information about DFSMS/MVS, including guidance in establishing storage management practices and implementing system-managed storage.

Storage Subsystem Library (SSL) provides comprehensive information about direct access storage devices (DASD) and storage control units (with and without cache).

Related Product Publications provide helpful information for tasks associated with DFSMS/MVS. Included are publications for the OS/390 operating system.

Information about electronic access and other ways you can acquire DFSMS/MVS books and related publications is also included here.

DFSMS/MVS Library

The DFSMS/MVS library contains information that helps you evaluate, install, customize, use, and diagnose the DFSMS/MVS product.

Because DFSMS/MVS includes the functions formerly provided in MVS/DFP Version 3, DFDSS Version 2, and DFHSM Version 2, the DFSMS/MVS library contains information from the libraries of these earlier products as well as information on newer functions exclusive to DFSMS/MVS.

DFSMS/MVS publications are available to you through the following channels:

- As a DFSMS/MVS license holder, you receive DFSMS/MVS publications as part of the distribution package containing the DFSMS/MVS program tapes and program directory. The IBM Distribution Center includes those publications required for you to plan, install, and customize functions of the current release of DFSMS/MVS (classified as basic materials).

  DFSMS/MVS Online Product Library is also shipped with your order. This CD-ROM includes unencrypted licensed materials and is available only to licensees of DFSMS/MVS.

- You can order additional copies of all DFSMS/MVS printed books or the CD-ROM from the IBM Distribution Center or your local branch office.
- DFSMS/MVS publications are available with these kits: *IBM Online Library Omnibus Edition MVS Collection* and *IBM Online Library Omnibus Edition OS/390 Collection*.
- If you subscribe to the S/390 Technical Library you can access selected DFSMS/MVS publications electronically through IBMLINK or on the Internet. See “Electronic Access to Online IBM Books” on page 303 for more information.

Publication	Order No.	Description of Contents	Audience
General Product Information			

© Copyright IBM Corp. 1979, 1999

Publication	Order No.	Description of Contents	Audience
DFSMS/MVS General Information	GC26-4900	Overview of DFSMS/MVS.	all users
DFSMS/MVS Library Guide	GC26-4902	Highlights of library changes for the current release; Cross-reference to DFSMS/MVS books and related product publications.	all users

Installation and Migration Planning			
DFSMS/MVS Licensed Program Specifications	GC26-4903	Warranty information for DFSMS/MVS.	varies
DFSMS/MVS Planning for Installation	SC26-4919	Guidance for planning the installation of DFSMS/MVS and migration to new functions of DFSMSdfp, DFSMShsm, DFSMSdss, and DFSMSrmm.	system programmer, system analyst, storage administrator

Customization			
DFSMS/MVS DFSMShsm Implementation and Customization Guide	SH21-1078	Guidance for creating and maintaining DFSMShsm-owned data sets, including migration and coexistence considerations.	system programmer, storage administrator
DFSMS/MVS DFSMSrmm Implementation and Customization Guide	SC26-4932	Information about implementing and customizing the functions of DFSMSrmm.	system programmer, storage administrator
DFSMS/MVS Installation Exits	SC26-4908	Instructions on exit routines and modules that extend or replace IBM-supplied functions.	system programmer, storage administrator

Storage Management			
MVS/ESA SML: Leading a Storage Administration Group	SC26-3126	Guidance for setting up a storage administration group and planning for system-managed storage.	data processing manager, storage administration group manager
DFSMS/MVS Implementing System-Managed Storage	SC26-3123	Tasks for migrating to system-managed storage and optimizing use of the storage management subsystem.	storage administrator, system programmer
MVS/ESA SML: Managing Data	SC26-3124	Techniques for establishing and enforcing data set policies, managing active and inactive data, and providing data set security.	storage administrator, system programmer
MVS/ESA SML: Managing Storage Groups	SC26-3125	Techniques for designing storage groups, making the transition to storage groups, and maintaining and monitoring the effectiveness of storage groups.	storage administrator, system programmer

DFSMS/MVS DFSMSdfp Storage Administration Reference	SC26-4920	Instructions for initializing and maintaining the storage management subsystem and for performing storage management tasks using ISMF panels.	storage administrator, system programmer
DFSMS/MVS DFSMShsm Storage Administration Guide	SH21-1076	DFSMShsm tasks for managing storage, including examples of storage administrator commands and results of DFSMShsm processing.	storage administrator, system programmer, operator
DFSMS/MVS DFSMShsm Storage Administration Reference	SH21-1075	DFSMShsm storage management commands, including syntax and coding techniques.	storage administrator, system programmer, operator
DFSMS/MVS DFSMShsm Storage Administration Reference Summary	SX26-3808	Summary of DFSMShsm storage management command syntax.	storage administrator, system programmer, operator
DFSMS/MVS Managing Data Availability	SC26-4928	Information about disaster prevention and the recovery of DFSMShsm data based on real experiences.	storage administrator, system programmer, application programmer
DFSMS/MVS DFSMSdss Storage Administration Guide	SC26-4930	DFSMSdss tasks for copying, dumping, and restoring data sets and DASD volumes.	storage administrator, system programmer, operator
DFSMS/MVS DFSMSdss Storage Administration Reference	SC26-4929	DFSMSdss commands, including syntax and coding examples.	storage administrator, system programmer, operator
DFSMS/MVS DFSMSrmm Command Reference Summary	SX26-6016	Summary of DFSMSrmm command syntax.	storage administrator, system programmer, tape librarian, operator, and DFSMSrmm end user
DFSMS/MVS DFSMSrmm Guide and Reference	SC26-4931	Instructions for using DFSMSrmm to manage removable media.	storage administrator, system programmer, tape librarian, operator
DFSMS/MVS DFSMSrmm Application Programming Interface	SC26-7272	storage administrator, system programmer, tape librarian, operator	
DFSMS/MVS Using the Volume Mount Analyzer	SC26-4925	Instructions for analyzing tape mounts with the volume mount analyzer.	storage administrator, system programmer
DFSMS Optimizer User's Guide and Reference	SC26-7047	Instructions for installing and using the DFSMS Optimizer for your storage system analysis.	performance analyst, storage administrator, capacity planner, financial analyst
DFSMS/MVS NaviQuest User's Guide	SC26-7194	Instructions for using the NaviQuest tool to migrate to SMS and maintain an SMS configuration.	storage administrator, system programmer

Catalog Management			
DFSMS/MVS Managing Catalogs	SC26-4914	Techniques for managing integrated catalog facility catalogs, VSAM catalogs, and OS CVOLs.	system programmer, storage administrator
DFSMS/MVS Access Method Services for ICF	SC26-4906	IDCAMS commands for using integrated catalog facility catalogs.	system programmer, storage administrator, operator
DFSMS/MVS Summary of Access Method Services for ICF	SX26-3807	Summary of IDCAMS commands for integrated catalog facility catalogs.	system programmer, storage administrator, operator
DFSMS/MVS Access Method Services for VSAM	SC26-4905	IDCAMS commands for using VSAM catalogs.	system programmer, operator

Data Management			
DFSMS/MVS Using ISMF	SC26-4911	Commands and techniques for using ISMF to analyze and manage data resources.	ISMF end user
DFSMS/MVS DFSMShsm Managing Your Own Data	SH21-1077	DFSMShsm user command syntax, conventions, and coding examples.	DFSMShsm end user
DFSMS/MVS DFSMShsm User Commands Reference Summary	SX26-3806	Summary of DFSMShsm user command syntax.	DFSMShsm end user
DFSMS/MVS Macro Instructions for Data Sets	SC26-4913	Access method macro instructions for processing user data sets.	application programmer, system programmer
DFSMS/MVS DFSMSdfp Advanced Services	SC26-4921	Techniques for extending DFSMS/MVS data management capabilities and other system data operations.	system programmer
DFSMS/MVS Using Data Sets	SC26-4922	Techniques for processing, protecting, and maintaining user data sets.	application programmer, system programmer
DFSMS/MVS Utilities	SC26-4926	Instructions for using DFSMSdfp utility programs for data, program, and device management.	application programmer, system programmer, storage administrator
DFSMS/MVS Checkpoint/Restart	SC26-4907	Information on establishing checkpoints during a program and restarting a job.	application programmer, system programmer
DFSMS/MVS Remote Copy Administrator's Guide and Reference	SC35-0169	Instructions for implementing remote copy as a disaster recovery and workload migration tool.	

Program Management			
DFSMS/MVS Program Management	SC26-4916	Instructions for using the binder, linkage editor, and loader to create, store, load, access, and update load modules and program objects.	application programmer, system programmer

Device Management			
DFSMS/MVS OAM Application Programmer's Reference	SC26-4917	Information about the OAM application interface used to manipulate objects.	application programmer
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Object Support	SC26-4918	OAM planning, installation, customizing, and storage management information for object support using DASD, and optical and tape libraries.	storage administrator, system programmer
DFSMS/MVS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries	SC26-3051	OAM planning, installation, customizing, and storage management information for tape libraries.	storage administrator, system programmer
DFSMS/MVS Using Magnetic Tapes	SC26-4923	Techniques for using magnetic tape volumes.	application programmer, storage administrator, system programmer

Distributed Data Access			
OS/390 Network File System User's Guide	SC26-7254	Instructions for accessing and creating MVS/ESA data sets from an AIX, UNIX, OS/2, or DOS environment using NFS protocol.	application programmer, system programmer
OS/390 Network File System Customization and Operation	SC26-7253	Techniques for customizing and operating the OS/390 Network File System server.	system programmer
DFSMS 1.3 Network File System Customization and Operations Guide	SC26-7029	Techniques for customizing and operating the DFSMS/MVS Network File System server.	system programmer
DFSMS/MVS DFM/MVS Guide and Reference	SC26-4915	Information about accessing and creating MVS/ESA data sets from remote systems.	application programmer, system programmer

Diagnosis (Restricted Materials)			
DFSMS/MVS DFSMSdfp Diagnosis Guide	SY27-9605	Information for diagnosing DFSMSdfp-related errors, including instructions for building keyword strings to search for known component failures.	system programmer, operator
DFSMS/MVS DFSMSdfp Diagnosis Reference	LY27-9606	Guidance for using DFSMSdfp diagnostic tools and aids to gather failure-related information.	system programmer, operator
DFSMS/MVS DFSMShsm Diagnosis Guide	LY27-9607	Diagnosis information for DFSMShsm, including instructions for building keyword search strings and other diagnostic tools.	system programmer, operator
DFSMS/MVS DFSMShsm Diagnosis Reference	LY27-9608	Information about the DFSMShsm control blocks and data areas used during diagnostic and maintenance procedures.	system programmer, operator
DFSMS/MVS DFSMSdss Diagnosis Guide	LY27-9609	Diagnosis information for DFSMSdss, including keyword search strings and other diagnostic tools.	system programmer, operator
DFSMS/MVS DFSMSrmm Diagnosis Guide	SY27-9615	Information for diagnosing DFSMSrmm-related errors, including how to use DFSMSrmm diagnostic tools.	system programmer, operator

Storage Subsystem Library

The Storage Subsystem Library (SSL) provides comprehensive information about direct access storage devices (DASD) and storage control units. In addition to describing characteristics, capabilities, and configurations for specific devices, the SSL specifies software requirements and options for using the devices in various operating environments. These publications complement books in the DFSMS/MVS library by providing device-specific techniques for effective storage management.

Publications from the SSL are shipped to you when you order these devices, and can be ordered separately as well. You can also receive SSL publications along with Device Support Facilities publications in BookManager readable format by ordering the *IBM Storage Subsystem Library and Device Support Facilities Release 14 Collection Kit* (SK2T-1209). The table here lists many of the books in the SSL. See your IBM representative for other device-related publications.

SSL Publication	Order No.	Description of Contents
Storage Subsystem Library Master Bibliography, Index, and Glossary	GC26-4496	Locating storage subsystem information.
IBM 3380 Direct Access Storage Introduction	GC26-4491	Overview of all IBM 3380 Models.
Using the IBM 3380 Direct Access Storage in an MVS Environment	GC26-4492	Planning, installing, and operating 3380s under MVS.
IBM 3390 Direct Access Storage Introduction	GC26-4573	Overview of all IBM 3390 Models.
Using IBM 3390 Direct Access Storage in an MVS Environment	GC26-4574	Planning, installing, and operating 3390s under MVS.
IBM 3990 Storage Control Introduction	GA32-0098	Introduction to the 3990.
IBM 3990 Storage Control Planning, Installation, and Storage Administration Guide	GA32-0100	Planning, installing, operating, and using the IDCAMS cache commands with the 3990.
IBM 9340 Direct Access Storage Subsystems Introduction	GC26-4694	Introduction to the 9340.
Using IBM 9340 Direct Access Storage Subsystems in an MVS Environment	GC26-4646	Planning, installing, and operating 9340s under MVS.
IBM RAMAC Array DASD Introduction	GC26-7012	Introduction to RAMAC array DASD
Using IBM RAMAC Array DASD in MVS, VM, or VSE Environment	GC26-7013	Planning, installing, and operating RAMAC array DASD under MVS, VM, or VSE
IBM RAMAC Array Subsystem Introduction	GC26-7004	Introduction to the RAMAC array subsystem
Using IBM RAMAC Array Subsystem in MVS, VM, or VSE Environment	GC26-7005	Planning, installing, and operating the RAMAC array subsystem under MVS, VM, or VSE
IBM RAMAC Array Subsystem Reference	GC26-7006	Reference information on channel commands, sense bytes and error recovery procedures

Related Product Publications

The following publications are helpful for performing tasks related to, but not specifically part of, DFSMS/MVS. If a publication is not available through the IBM Distribution Center, contact your local IBM representative for assistance.

Publication	Order No.	Description of Contents
Online Library Collection Kits (CD-ROM)		
IBM Online Library Omnibus Edition MVS Collection	SK2T-0710	MVS-based publication libraries (including DFSMS/MVS library) in BookManager readable format.

System/390 Sysplex Publications		
OS/390 Parallel Sysplex Overview	GC28-1860	Introduction to the sysplex environment.
OS/390 Parallel Sysplex Hardware and Software Migration	GC28-1862	Planning for system-wide sysplex implementation.
OS/390 Parallel Sysplex Systems Management	GC28-1861	Planning for business, configuration, operations, and change management in a sysplex environment.
OS/390 Parallel Sysplex Application Migration	GC28-1863	Planning for migrating applications to an MVS/ESA sysplex environment.

MVS/ESA Publications		
OS/390 Introduction and Release Guide	GC28-1725	Overview of MVS/ESA SP and its requirements.
OS/390 Planning for Installation	GC28-1726	Requirements and planning information for an OS/390 installation.
OS/390 MVS Conversion Notebook	GC28-1747	Technical descriptions and migration actions for converting to the latest release of MVS/ESA.
OS/390 MVS Initialization and Tuning Guide	SC28-1751	Guidance for initializing MVS and improving system performance.
OS/390 MVS Initialization and Tuning Reference	SC28-1752	Reference to SYS1.PARMLIB members and selected system commands.
OS/390 JES2 Initialization and Tuning Reference	SC28-1792	Planning for installing, initializing, customizing and tuning JES2.
OS/390 JES3 Introduction	GC28-1808	Overview of JES3.
OS/390 JES3 Initialization and Tuning Reference	SC28-1803	Planning for installing, initializing, customizing and tuning JES3.
OS/390 HCD User's Guide	SC28-1848	Procedures for using HCD dialogs.
OS/390 MVS Planning: APPC/MVS Management	GC28-1807	Planning information for configuring, activating, customizing, and maintaining APPC/MVS.
OS/390 MVS Planning: Global Resource Serialization	GC28-1759	Planning information for using GRS to serialize MVS resources.
OS/390 MVS Setting Up a Sysplex	GC28-1779	Overview of setting up an MVS system to run in a sysplex.
OS/390 MVS System Management Facilities (SMF)	GC28-1783	Planning information for implementing SMF.
OS/390 MVS System Data Set Definition	GC28-1782	Instructions for defining system data sets, including those required for DFSMS/MVS.
OS/390 MVS JCL User's Guide	GC28-1758	Instructions for coding job control language.
OS/390 MVS JCL Reference	GC28-1757	Job control language reference
OS/390 MVS System Messages, Vol 1 (ABA-ASA)	GC28-1784	Message library for MVS/ESA products, including DFSMS/MVS.
OS/390 MVS System Messages, Vol 2 (ASB-EWX)	GC28-1785	
OS/390 MVS System Messages, Vol 3 (GDE-IEB)	GC28-1786	
OS/390 MVS System Messages, Vol 4 (IEC-IFD)	GC28-1787	
OS/390 MVS System Messages, Vol 5 (IGD-IZP)	GC28-1788	

Licensed Program Publications		
Assembler H Version 2 General Information	GC26-4035	Overview of Assembler H and its requirements.
IBM High Level Assembler/MVS & VM & VSE General Information	GC26-4943	Overview of High Level Assembler and its requirements.
Character Data Representation Architecture Overview	GC09-2207	Overview of CDRA and its applications.
Character Data Representation Architecture Reference and Registry	SC09-2190	Architecture and programming reference for CDRA.
CICS Transaction Server for OS/390 Migration Guide	GC34-5353	Migrating CICS applications.
CICS Recovery and Restart Guide	SC33–1698	Recovering and restarting CICS.
DB2 for MVS/ESA Version 4 Installation Guide	SC26-3456	Installing DB2 for MVS/ESA
Distributed Data Management Architecture: General Information	GC21-9527	Overview of DDM architecture.
DFSORT Getting Started R14	SC26-4109	Overview of DFSORT and its requirements.
DFSORT Application Programming Guide R14	SC33-4035	Instructions for using DFSORT to sort, merge, and copy data sets.
Device Support Facilities User's Guide and Reference	GC35-0033	Initializing DASD volumes, recovering from track errors.
Environmental Record Editing and Printing Program User's Guide and Reference	GC28-1378	Instructions for using EREP to record, edit, and print error records.
Integrated Cryptographic Service Facility/MVS General Information	GC23-0093	Overview of ICSF/MVS and its requirements.
OS/390 UNIX System Services Planning	SC28-1890	Planning information for setting up and managing OS/390 UNIX System Services.
OS/390 Security Server (RACF) Introduction	GC28-1912	Overview of RACF and its requirements.
OS/390 Security Server (RACF) General User's Guide	SC28-1917	Instructions for using RACF to perform basic security tasks.

Hardware Product Publications		
Advanced Function Printing: Printer Summary	G544-3135	Listing of AFP printers and their characteristics.
IBM 3494 Tape Library Dataserver Introduction and Planning Guide	GA32-0279	Planning for installation of the IBM 3494 Tape Library Dataserver including migration from non-automated tape to a fully automated tape library environment.
IBM 3495 Tape Library Dataserver Installation Planning and Migration Guide	GC35-0135	Planning for installation of the IBM 3495 Tape Library Dataserver including migration from non-automated tape to a fully automated tape library environment.
IBM 3995 Optical Library Dataserver Products: Introduction and Planning Guide	GA32-0121	Introduction to the IBM 3995 family of optical libraries and planning information for installing the machines.

ITSO Redbooks		
Get DFSMS FIT: Fast Implementation Techniques	SG24-2568	Information on using the DFSMS Fast Implementation Technique (FIT) for system-managed storage.
IBM Magstar Virtual Tape Server and Enhancements to Magstar: A New Era in Tape	SG24-4917	Description of the IBM Magstar Virtual Tape Server

Electronic Access to Online IBM Books

The S/390 Technical Library provides electronic access to over 5000 online IBM books in BookManager format.

The S/390 Technical Library is part of the System/390 Support Family of Services, a portfolio of services designed to complement IBM's software, hardware, and technical support services. Users can select S/390 Technical Library as one of the individual services to meet their specific requirements.

The S/390 Technical Library uses the BOOKS application on ServiceLink. The BOOKS application provides users with access to IBM product books available online. This access is provided electronically, through IBMLINK, from the ServiceLink main menu. The categories of available books are:

AIX	AIX/ESA
AS/400	AS/400
BULLETIN	US Technical Bulletins
CROSS PLATFORM	Cross Platform Applications (AD/Cycle, GDDM, DCF, etc.)
MVS/ESA	MVS/ESA (JES3, DFSMS, TSO/E, ISPF, SMP/E, DFSORT, etc.)
NETWORKING	Networking Systems (NCP, Netview, VTAM, TCP/IP, etc.)
PWS	Programmable Workstations (OS/2, LAN, MMPM/2, etc.)
REDBOOKS	ITSO Technical Bulletins
S/390 HARDWARE	Large Systems Hardware (ES/9000, ESCON, etc.)
TRANSACTION	Transaction Processing (CICS, IMS, DB2, QMF, OVVM, etc.)
VM, VSE	VM/ESA, VSE/ESA

Some of the functions available in BOOKS are the ability to select a category of books for reading, the ability to select a specific book to read, and the ability to do keyword searching in a single book or across a bookshelf of books. Form number and title searches are also allowed.

Internet Access

The S/390 Technical Library is now available on the Internet, off the IBM Home Page. Users who have subscribed to the S/390 Technical Library can access the service on the Internet by specifying the URL:

`http://www1.ibmlink.ibm.com/cgi-bin/master`

IBM Public Books

A selection of IBM technical books is available in online format on the Internet for access by all users. These books provide marketing support, planning and general information about products. You can access these books by specifying the URL given below and then selecting “Public Entry.”

`http://www1.ibmlink.ibm.com/cgi-bin/master`

How To Subscribe

For more information about the S/390 Technical Library service and how to subscribe to it, contact your IBM representative or call the Support Family Information Center at 1-800-742-9235.

Appendix B. Abbreviations

The following abbreviations are defined as they are used in the DFSMS/MVS library. If you do not find the term you are looking for, see the *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994. This list may include some terms from the *IBM Dictionary of Computing*.

This list may also include acronyms and abbreviations from:

> The *American National Standard Dictionary for Information Systems*,ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 1430 Broadway, New York, New York 10018.
> The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1).

ACB. Access method control block.

ACDS. Active control data set.

ACS. Automatic class selection.

AMODE. Addressing mode.

API. Application program interface.

APPC. Advanced Program-to-Program Communication.

| **ANSI**. American National Standards Institute.

| **ASCII**. American Standard Code for Information
| Interchange.

BCS. Basic catalog structure.

BCDS. Backup control data set.

BPAM. Basic partitioned access method.

BSAM. Basic sequential access method.

CAS. Catalog address space.

| **CCSID**. Coded character set identifier.

CDRA. Character Data Representation Architecture.

CDS. Control data set.

CF. Coupling facility.

CI. Control interval.

CICS. Customer Information Control System.

COMMDS. Communications data set.

CSA. Common service area.

CVAF. Common VTOC access facility.

DADSM. Direct access device space management.

DASD. Direct access storage device.

DDM. Distributed Data Management.

DFM. Distributed FileManager.

DFSORT. Data Facility Sort.

DSORG. Data set organization.

EA. Extended addressability.

EF. Extended format.

ELPA. Extended link pack area.

EPLPA. Extended pageable link pack area.

ESA. Enterprise Systems Architecture.

ESCON. Enterprise System Connection.

ESD. External symbol dictionary.

ESDS. Entry-sequenced data set.

EXCP. Execute channel program.

FLPA. Fixed link pack area.

© Copyright IBM Corp. 1979, 1999

GB. Gigabyte

GRS. Global resource serialization.

GSR. Global shared resources.

GUI. Graphical user interface.

HCD. Hardware Configuration Definition.

HFS. Hierarchical file system.

IART. Initial access response time.

ICF. Integrated catalog facility.

ICKDSF. Device Support Facilities.

IMS. Information Management System.

IPL. Initial program load.

ISMF. Interactive Storage Management Facility.

ISO. International Organization for Standardization.

JCL. Job control language.

KB. Kilobyte.

KSDS. Key-sequenced data set.

LDS. Linear data set.

LE. Language Environment.

LPA. Link pack area.

LSR. Local shared resources.

MB. Megabyte.

MCDS. Migration control data set.

ML1. Migration level 1.

ML2. Migration level 2.

MLPA. Modified link pack area.

MSR. Millisecond response time.

MVS. Multiple virtual storage.

NFS. Network File System.

NSR. Non-shared resources.

NVR. Non-VSAM volume record.

OAM. Object access method.

OCDS. Offline control data set.

OCO. Object code only.

OLTP. Online transaction processing.

PDS. Partitioned data set.

PDSE. Partitioned data set extended.

PLPA. Pageable link pack area.

PSF. Print Services Facility

PSP. Preventive Service Planning.

PTF. Program Temporary Fix.

QSAM. Queued sequential access method.

RACF. Resource Access Control Facility.

RBA. Relative byte address.

RETAIN. Remote Technical Assistance and Information Network.

RLS. Record-level sharing.

RMF. Resource Measurement Facility.

RMODE. Residence mode.

RPL. Request parameter list.

RRDS. Relative-record data set.

RSECT. Read-only control section.

SAA. Systems Application Architecture.

SAF. System Authorization Facility.

SCDS. Source control data set.

SDSP. Small data set packing.

SHCDS. Sharing control data set.

SMF. System management facility.

SMS. Storage Management Subsystem.

SNA. Systems Network Architecture.

TB. Terabyte.

TCDB. Tape configuration database.

TCP/IP. Transmission Control Protocol/Internet Protocol.

TMM. Tape mount management.

UCB. Unit control block.

UIM. Unit information module.

VIO. Virtual I/O.

VRRDS. Variable-length relative-record data set.

VSAM. Virtual Storage Access Method.

VTOC. Volume table of contents.

VTS. Virtual tape server.

VVDS. VSAM volume data set.

VVR. VSAM volume record.

WORM. Write once read many.

XCF. Cross-system coupling facility.

XRC. Extended remote copy.

Glossary

The following terms are defined as they are used in the DFSMS/MVS Library. If you do not find the term you are looking for, see the IBM Software Glossary:

`http://www.networking.ibm.com/nsg/nsgmain.htm`

This glossary is an ever-evolving document that defines technical terms used in the documentation for many IBM software products.

Numerics

8–name mode. See *compatibility mode*.

32-name mode. For DFSMS/MVS, it is the mode of running SMS on a DFSMS/MVS system in which up to 32 names—representing systems, system groups, or both—are supported in the SMS configuration. When running in this mode, the DFSMS/MVS system can only share SCDSs, ACDSs, and COMMDSs with other DFSMS/MVS systems running in 32-name mode.

A

access method control block (ACB). A control block that links an application program to VSAM or VTAM programs.

access method services. A multifunction service program that manages VSAM and non-VSAM data sets, as well as integrated catalog facility (ICF) and VSAM catalogs. Access method services provides the following functions:

- defines and allocates space for VSAM data sets, VSAM catalogs, and ICF catalogs
- converts indexed-sequential data sets to key-sequenced data sets
- modifies data set attributes in the catalog
- reorganizes data sets
- facilitates data portability among operating systems
- creates backup copies of data sets
- assists in making inaccessible data sets accessible
- lists the records of data sets and catalogs
- defines and builds alternate indexes
- converts CVOLS and VSAM catalogs to ICF catalogs

ACS installation exit. User-written code, run after an ACS routine, that provides capabilities beyond the scope of the ACS routine.

ACS interface routine. This calls an ACS routine from an ACS installation-exit routine.

activate. To load the contents of a source control data set (SCDS) into Storage Management Subsystem address space storage and into an active control data set (ACDS), or to load the contents of an existing ACDS into subsystem address space storage. This establishes a new storage management policy for the subsystem complex.

active configuration. The most recently activated SCDS, which now controls storage management for the Storage Management Subsystem complex.

active control data set (ACDS). A VSAM linear data set that contains an SCDS that has been activated to control the storage management policy for the installation. When activating an SCDS, you determine which ACDS will hold the active configuration (if you have defined more than one ACDS). The ACDS is shared by each system that is using the same SMS configuration to manage storage. See also *source control data set* and *communications data set*.

active data. (1) Data that can be accessed without any special action by the user, such as data on primary storage or migrated data. Active data also can be stored on tape volumes. (2) For tape mount management, application data that is frequently referenced, small in size, and managed better on DASD than on tape. Contrast with *inactive data*.

addressing mode (AMODE). An attribute of an entry point in a load module that identifies the addressing range in virtual storage which the module is capable of addressing. Below the 16-megabyte line, only 24-bit addresses can be used.

aggregate backup. The process of copying an aggregate group and recovery instructions so that a collection of data sets can be recovered later as a group.

aggregate group. A collection of related data sets and control information that have been pooled to meet a defined backup or recovery strategy.

alternate index. In systems with VSAM, a collection of index entries related to a given base cluster and organized by an alternate key, that is, a key other than the prime key of the associated base cluster data records; it gives an alternate directory for finding records in the data component of a base cluster.

always call. See *RACF always call*.

© Copyright IBM Corp. 1979, 1999

AMASPZAP. A service program used to used to dynamically update or dump programs and data sets.

automated tape library. A device consisting of robotic components, cartridge storage areas, tape subsystems, and controlling hardware and software, together with the set of tape volumes that reside in the library and can be mounted on the library tape drives. See also *tape library*. Contrast with *manual tape library*.

automatic backup. (1) In DFSMShsm, the process of automatically copying data sets from primary storage volumes or migration volumes to backup volumes. (2) In OAM, the process of automatically copying objects from DASD, optical, or tape volumes contained in an object storage group, to backup volumes contained in an object backup storage group.

automatic class selection (ACS) routine. A procedural set of ACS language statements. Based on a set of input variables, the ACS language statements generate the name of a predefined SMS class, or a list of names of predefined storage groups, for a data set.

automatic dump. In DFSMShsm, the process of using DFSMSdss automatically to do a full-volume dump of all allocated space on a primary storage volume to designated tape dump volumes.

automatic primary space management insert. In DFSMShsm, the process of deleting expired data sets, deleting temporary data sets, releasing unused space, and migrating data sets from primary storage volumes automatically.

automatic secondary space management. In DFSMShsm, the process of automatically deleting expired migrated data sets, deleting expired records from the migration control data sets, and migrating eligible data sets from migration level 1 volumes to migration level 2 volumes.

automatic volume space management. In DFSMShsm, the process that includes automatic primary space management and interval migration.

availability. For a storage subsystem, the degree to which a data set or object can be accessed when requested by a user.

B

backup. The process of creating a copy of a data set or object to be used in case of accidental loss.

backup control data set (BCDS). In DFSMShsm, a VSAM key-sequenced data set that contains information about backup versions of data sets, backup volumes, dump volumes, and volumes under control of the backup and dump functions of DFSMShsm.

backup-while-open (BWO). This makes a backup copy of a data set while the data set is open for update. The backup copy can contain partial updates.

base configuration. The part of an SMS configuration that contains general storage management attributes, such as the default management class, default unit, and default device geometry. It also identifies the systems or system groups that an SMS configuration manages.

basic catalog structure (BCS). The name of the catalog structure in the integrated catalog facility environment. See also *integrated catalog facility catalog*.

binder. The DFSMS/MVS program that processes the output of language translators and compilers into an executable program (load module or program object). It replaces the linkage editor and batch loader in MVS

buffer. A routine or storage used to compensate for a difference in rate of flow of data, or time of occurrence of events, when transferring data from one device to another.

C

cache fast write. A storage control capability in which the data is written directly to cache without using nonvolatile storage. Cache fast write is useful for temporary data or data that is readily recreated, such as the sort work files created by DFSORT. Contrast with *DASD fast write*.

cache set. A parameter on storage class and defined in the base configuration information that maps a logical name to a set of CF cache structure names.

capacity planning. The process of forecasting and calculating the appropriate amount of physical computing resources required to accommodate an expected workload.

Cartridge System Tape. The base tape cartridge media used with 3480 or 3490 Magnetic Tape Subsystems. Contrast with *Enhanced Capacity Cartridge System Tape*.

Character Data Representation Architecture (CDRA) API. A set of identifiers, services, supporting resources, and conventions for consistent representation, processing, and interchange of character data.

class transition. An event that brings about change to an object's service-level criteria, causing OAM to invoke ACS routines to assign a new storage class or management class to the object.

client. (1) A function that requests services from a server, and makes them available to the user. (2) An

address space in MVS that is using TCP/IP services. (3) A term used in an environment to identify a machine that uses the resources of the network. See also *source*.

client-server relationship. Any process that provides resources to other processes on a network is a *server*. Any process that employs these resources is a *client*. A machine can run client and server processes at the same time.

cluster. In VSAM, a named structure consisting of a group of related components. For example, when the data is key-sequenced, the cluster contains both the data and the index components.

Coded Character Set Identifier (CCSID). A 16-bit number that identifies a specific encoding scheme identifier, character set identifiers, code page identifiers, and additional coding required information. The CCSID uniquely identifies the coded graphic character representation used.

coexistence. The ability of two or more systems to share resources at the same time. For example, two systems of different release levels of DFSMS/MVS might share data sets, catalogs, or volumes.

communications data set (COMMDS). The primary means of communication among systems governed by a single SMS configuration. The COMMDS is a VSAM linear data set that contains the name of the ACDS and current utilization statistics for each system-managed volume, which helps balance space among systems running SMS. See also *active control data set* and *source control data set*.

compatibility mode. For devices, it is a mode of operation that allows you to simulate the function of another device or model. This causes the device to function like a different device of the same type, ignoring some or all of the additional features the device might have. This allows you to migrate between devices with minimal impact on programs that have device dependencies.

compatibility mode. For DFSMS/MVS, it is the mode of running SMS in which no more than eight names—representing systems, system groups, or both—are supported in the SMS configuration. When running in this mode, the DFSMS/MVS system can share SCDSs, ACDSs and COMMDSs with other systems running MVS/DFP or DFSMS/MVS releases prior to DFSMS/MVS 1.3, and with other DFSMS/MVS systems running in compatibility mode.

compress. (1) To reduce the amount of storage required for a given data set by having the system replace identical words or phrases with a shorter token associated with the word or phrase. (2) To reclaim the unused and unavailable space in a partitioned data set that results from deleting or modifying members by moving all unused space to the end of the data set.

compressed format. A particular type of extended-format data set specified with the (COMPACTION) parameter of data class. VSAM can compress individual records in a compressed-format data set. SAM can compress individual blocks in a compressed-format data set. See *compress*.

concurrent copy. A function to increase the accessibility of data by enabling you to make a consistent backup or copy of data concurrent with the usual application program processing.

configuration (Storage Management Subsystem). See *SMS configuration*.

connectivity. (1) The considerations regarding how storage controls are joined to DASD and processors to achieve adequate data paths (and alternative data paths) to meet data availability needs. (2) In a DFSMS environment, the system status of volumes and storage groups.

construct. One of the following: data class, storage class, management class, storage group, aggregate group, base configuration.

control interval (CI). A fixed-length area of auxiliary storage space in which VSAM stores records. It is the unit of information (an integer multiple of block size) transmitted to or from auxiliary storage by VSAM.

convert in place. See *in-place conversion*.

coupling facility (CF). The hardware that provides high-speed caching, list processing, and locking functions in a Parallel Sysplex.

coupling facility (CF) cache structure. The CF hardware that provides a data cache.

coupling facility (CF) lock structure. The CF hardware that supports sysplex-wide locking.

D

DASD fast write. An extended function of some models of the IBM 3990 Storage Control in which data is written concurrently to cache and nonvolatile storage and automatically scheduled for destaging to DASD. Both copies are retained in the storage control until the data is completely written to the DASD, providing data integrity equivalent to writing directly to the DASD. Use of DASD fast write for system-managed data sets is controlled by storage class attributes to improve performance. See also *dynamic cache management*. Contrast with *cache fast write*.

DASD volume. A DASD space identified by a common label and accessed by a set of related addresses. See also *volume*, *primary storage*, *migration level 1*, *migration level 2*.

data class. A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

Data Facility Sort (DFSORT). An IBM licensed program that is a high-speed data processing utility. DFSORT provides an efficient and flexible way to handle sorting, merging, and copying operations, as well as providing versatile data manipulation at the record, field, and bit level.

data set. In DFSMS/MVS, the major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access. In OS/390 non-UNIX environments, the terms *data set* and *file* are generally equivalent and sometimes are used interchangeably. See also *file*. In OS/390 UNIX environments, the terms *data set* and *file* have quite distinct meanings.

data set collection. A group of data sets which are intended to be allocated on the same tape volume or set of tape volumes as a result of data set stacking.

data set stacking. The function used to place several data sets on the same tape volume or set of tape volumes. It increases the efficiency of tape media usage and reduces the overall number of tape volumes needed by allocation. It also allows an installation to group related data sets together on a minimum number of tape volumes, which is useful when sending data offsite.

default device geometry. Part of the SMS base configuration, it identifies the number of bytes per track and the number of tracks per cylinder for converting space requests made in tracks or cylinders into bytes, when no unit name has been specified.

default management class. Part of the SMS base configuration, it identifies the management class that should be used for system-managed data sets that do not have a management class assigned.

default unit. Part of the SMS base configuration, it identifies an esoteric (such as SYSDA) or generic (such as 3390) device name. If a user omits the UNIT parameter on the JCL or the dynamic allocation equivalent, SMS applies the default unit if the data set has a disposition of MOD or NEW and is *not* system-managed.

device category. A storage device classification used by SMS. The device categories are as follows SMS-managed DASD, SMS-managed tape, non-SMS-managed DASD non-SMS-managed tape.

device management. The task of defining input and output devices to the operating system, and then controlling the operation of these devices.

Device Support Facilities (ICKDSF). A program used for initialization of DASD volumes and track recovery.

DFSMS environment. An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. In the DFSMS environment for MVS, the function is provided by DFSORT, RACF, and the combination of DFSMS/MVS and MVS.

DFSMShsm control data set. In DFSMShsm, one of three VSAM key-sequenced data sets that contain records used in DFSMShsm processing. See also *backup control data set*, *migration control data set*, and *offline control data set*.

DFSMS/MVS. An IBM System/390 licensed program that provides storage, data, and device management functions. When combined with MVS/ESA SP Version 5 it composes the base MVS/ESA operating environment. DFSMS/MVS consists of DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm.

DFSMSdfp. A DFSMS/MVS functional component or base element of OS/390, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS/MVS functional component or base element of OS/390, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS/MVS functional component or base element of OS/390, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMShsm-managed volume. (1) A primary storage volume, which is defined to DFSMShsm but which does not belong to a storage group. (2) A volume in a storage group, which is using DFSMShsm automatic dump, migration, or backup services. Contrast with *system-managed volume* and *DFSMSrmm-managed volume*.

DFSMShsm-owned volume. A storage volume on which DFSMShsm stores backup versions, dump copies, or migrated data sets.

DFSMS/MVS Network File System. See *OS/390 Network File System*.

DFSMS/MVS Optimizer Feature. A DFSMS/MVS feature that provides an analysis and reporting capability for SMS and non-SMS environments.

DFSMSrmm. A DFSMS/MVS functional component or base element of OS/390, that manages removable media.

DFSMSrmm-managed volume. A tape volume that is defined to DFSMSrmm. Contrast with *system-managed volume* and *DFSMShsm-managed volume*.

dictionary. A table that associates words, phrases, or data patterns to shorter tokens. The tokens replace the associated words, phrases, or data patterns when a data set is compressed.

direct access device space management (DADSM). A DFSMSdfp component used to control space allocation and deallocation on DASD.

disaster recovery. A procedure for copying and storing an installation's essential business data in a secure location, and for recovering that data in the event of a catastrophic problem. Compare with *vital records*.

Distributed Data Management (DDM). A data protocol architecture for data management services across distributed systems in an SNA environment. DDM provides a common data management language for data interchange among different IBM system platforms.

Distributed FileManager/MVS. (1) The term used to describe the SAA architectures and programming support that provide distributed file access capabilities between SAA systems. (2) The DFSMS/MVS component that implements the DDM target server.

drive definition. A set of attributes used to define an optical disk drive as a member of a real optical library or pseudo optical library.

dual copy. A high availability function made possible by nonvolatile storage in some models of the IBM 3990 Storage Control. Dual copy maintains two functionally identical copies of designated DASD volumes in the logical 3990 subsystem, and automatically updates both copies every time a write operation is issued to the dual copy logical volume.

dummy storage group. A type of storage group that contains the serial numbers of volumes no longer connected to a system. Dummy storage groups allow existing JCL to function without having to be changed. See also *storage group*.

dump class. A set of characteristics that describes how volume dumps are managed by DFSMShsm.

duplexing. The process of writing two sets of identical records in order to create a second copy of data.

dynamic cache management. A function that automatically determines which data sets will be cached based on the 3990 subsystem load, the characteristics of the data set, and the performance requirements defined by the storage administrator.

E

eject. The process used to remove a volume from a system-managed library. For an automated tape library dataserver, the volume is removed from its cell location and moved to the output station. For a manual tape library dataserver, the volume is not moved, but the tape configuration database is updated to show the volume no longer resides in the manual tape library dataserver.

Enhanced Capacity Cartridge System Tape. Cartridge system tape with increased capacity that can only be used with 3490E Magnetic Tape Subsystems. Contrast with *Cartridge System Tape*.

erase-on-scratch. The physical erasure of data on a DASD data set when the data set is deleted (scratched).

esoteric unit name. A name used to define a group of devices having similar hardware characteristics, such as TAPE or SYSDA. Contrast with *generic unit name*.

expiration. The process by which data sets or objects are identified for deletion because their expiration date or retention period has passed. On DASD, data sets and objects are deleted. On tape, when all data sets have reached their expiration date, the tape volume is available for reuse.

export. The operation to remove one or more logical volumes from a virtual tape server library. First, the list of logical volumes to export must be written on an export list volume and then, the export operation itself must be initiated.

extended addressability. The ability to create and access a VSAM data set that is greater than 4 GB in size. Extended addressability data sets must be allocated with DSNTYPE=EXT and EXTENDED ADDRESSABILITY=Y.

extended format. The format of a data set that has a data set name type (DSNTYPE) of EXTENDED. The data set is structured logically the same as a data set that is not in extended format but the physical format is different. See also *striped data set* and *compressed format*.

extended link pack area (ELPA). The extension of the link pack area that resides above 16 megabytes in virtual storage. See also *link pack area*.

extended pageable link pack area (EPLPA). The extension of the pageable link pack area that resides

above 16 megabytes in virtual storage. See also *pageable link pack area*.

extended format 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 format data sets consist of one or more stripes and can take advantage of the sequential data striping access technique. See also *striping* and *stripe*.

extended remote copy. Extended Remote Copy (XRC) is a technique involving both the DFSMS/MVS host and the I/O Subsystem that keeps a "real time" copy of designated data at another location. Updates to the primary center are replicated at the secondary center asynchronously.

F

file. A collection of information treated as a unit. In non-OS/390 UNIX environments, the terms *data set* and *file* are generally equivalent and are sometimes used interchangeably. See also *data set*.

file system. In the OS/390 UNIX HFS environment, the collection of files and file management structures on a physical or logical mass storage device, such as a diskette or minidisk. See also *HFS data set*.

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.

G

generic unit name. A name assigned to a class of devices with the same geometry (such as 3390). Contrast with *esoteric unit name*.

global access checking. In RACF, the ability to establish an in-storage table of default values containing authorization levels for selected resources. RACF refers to this table prior to performing its usual RACHECK processing, and grants the request without performing a RACHECK if the requested access authority does not exceed the global value. Global access checking can grant a user access to the resource, but it cannot deny access.

global resource serialization (GRS). A component of MVS used for serializing use of system resources and for converting hardware reserves on DASD volumes to data set enqueues.

GRS complex (GRSplex). One or more MVS images that share a common global resource serialization policy in either a ring or star configuration.

group. (1) With respect to partitioned data sets, a member and the member's aliases that exist in a PDS or PDSE, or in an unloaded PDSE. (2) A collection of users who can share access authorities for protected resources.

guaranteed space. A storage class attribute indicating that DFSMS is to honor user-specified VOL=SER= parameters, and that the Storage Management Subsystem is to fail the request if space is not available on the requested volumes. You can use this attribute to preallocate space on all volumes that you specify. If you do not supply volume serials, the Storage Management Subsystem selects volumes and preallocates space on as many of them as you requested, either explicitly by being specified on the JCL or implicitly from the “volume count” field of the data class.

H

hardware configuration definition (HCD). An interactive interface in MVS that enables an installation to define hardware configurations from a single point of control.

hierarchical file system (HFS) data set. A data set that contains a POSIX-compliant file system, which is a collection of files and directories organized in a hierarchical structure, that can be accessed using OS/390 UNIX System Services. See also*file system.*

Hiperspace. A high-performance space backed by either expanded storage or auxiliary storage, which provides high performance storage and retrieval of data.

HSM complex (HSMplex). One or more MVS images running DFSMShsm that share a common set of control data sets (MCDS, BCDS, OCDS, and Journal).

I

implementation by milestone. A conversion approach that allows for a staged conversion of your installation’s data to system-managed storage on DASD, tape, or optical devices.

import. The operation to enter previously exported logical volumes residing on a stacked volume into a virtual tape server library. First, the list of logical volumes to import must be written on an import list volume and the stacked volumes must be entered, and then, the import operation itself must be initiated.

improved data recording capability (IDRC). A recording mode that can increase the effective cartridge data capacity and the effective data rate when enabled and used. IDRC is always enabled on the 3490E Magnetic Tape Subsystem.

inactive data. (1) A copy of active data, such as vital records or a backup copy of a data set. Inactive data is never changed, but can be deleted or superseded by another copy. (2) In tape mount management, data that is written once and never used again. The majority of this data is point-in-time backups. (3) Objects infrequently accessed by users and eligible to be moved to the optical library or shelf. Contrast with *active data*.

indexed VTOC. A volume table of contents with an index that contains a list of data set names and free space information, which allows data sets to be located more efficiently.

in-place conversion. The process of bringing a volume and the data sets it contains under the control of SMS without data movement, using DFSMSdss.

integrated catalog facility catalog. A catalog that is composed of a basic catalog structure (BCS) and its related volume tables of contents (VTOCs) and VSAM volume data sets (VVDSs). See also *basic catalog structure* and *VSAM volume data set*.

Interactive Storage Management Facility (ISMF). The interactive interface of DFSMS/MVS that allows users and storage administrators access to the storage management functions.

interval migration. In DFSMShsm, automatic migration that occurs when a threshold level of occupancy is reached or exceeded on a DFSMShsm-managed volume, during a specified time interval. Data sets are moved from the volume, largest eligible data set first, until the low threshold of occupancy is reached.

K

key-sequenced data set (KSDS). A VSAM data set whose records are loaded in ascending key sequence and controlled by an index.

L

linear data set. A VSAM data set that contains data but contains no control information. A linear data set can be accessed as a byte-addressable string in virtual storage.

link pack area (LPA). In MVS, an area of virtual storage that contains reenterable routines that are loaded at IPL time and can be used concurrently by all tasks in the system.

load module. An executable program stored in a partitioned data set program library. See also *program object*.

M

management class. A collection of management attributes, defined by the storage administrator, used to control the release of allocated but unused space; to control the retention, migration, and backup of data sets; to control the retention and backup of aggregate groups, and to control the retention, backup, and class transition of objects.

manual tape library. A set of tape drives defined as a logical unit by the installation together with the set of system-managed volumes which can be mounted on those drives. See also *tape library*. Contrast with *automated tape library*.

migration. The process of moving unused data to lower cost storage in order to make space for high-availability data. If you wish to use the data set, it must be recalled. See also *migration level 1* and migration level 2.

migration control data set (MCDS). In DFSMShsm, a VSAM key-sequenced data set that contains statistics records, control records, user records, records for data sets that have migrated, and records for volumes under migration control of DFSMShsm.

migration level 1. DFSMShsm-owned DASD volumes that contain data sets migrated from primary storage volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 2*.

migration level 2. DFSMShsm-owned tape or DASD volumes that contain data sets migrated from primary storage volumes or from migration level 1 volumes. The data can be compressed. See also *storage hierarchy*. Contrast with *primary storage* and *migration level 1*.

modified link pack area (MLPA). An area of virtual storage containing reenterable routines from the SYS1.LINKLIB, SYS1.SVCLIB, or SYS1.LPALIB system data sets that are to be part of the pageable extension of the link pack area during the current IPL. See also *link pack area*.

MVS/ESA. An MVS operating system environment that supports ESA/390.

MVS/ESA SP. An IBM licensed program used to control the MVS operating system. MVS/ESA SP together with DFSMS/MVS compose the base MVS/ESA operating environment. See also *OS/390*.

N

NaviQuest. A component of DFSMSdfp for implementing, verifying, and maintaining your DFSMS SMS environment in batch mode. It provides batch testing and reporting capabilities that can be used to automatically create test cases in bulk, run many other storage management tasks in batch mode, and use supplied ACS code fragments as models when creating your own ACS routines.

nondisruptive installation. A capability that allows users to access data in an existing storage subsystem while installing additional devices in the subsystem.

nonvolatile storage (NVS). Additional random access electronic storage with a backup battery power source, available with an IBM Cache Storage Control, used to retain data during a power outage. Nonvolatile storage, accessible from all storage directors, stores data during DASD fast write and dual copy operations.

O

OAM-managed volumes. Optical or tape volumes controlled by the object access method (OAM).

OAM complex (OAMplex). One or more instances of OAM running on systems that are part of a parallel sysplex. The OAM systems that are part of an OAMplex share a common OAM database in a DB2 data-sharing group.

object. A named byte stream having no specific format or record orientation.

object access method (OAM). An access method that provides storage, retrieval, and storage hierarchy management for objects and provides storage and retrieval management for tape volumes contained in system-managed libraries.

object backup storage group. A type of storage group that contains optical or tape volumes used for backup copies of objects. See also *storage group*.

object storage group. A type of storage group that contains objects on DASD, tape, or optical volumes. See also *storage group*.

object storage hierarchy. A hierarchy consisting of objects stored in DB2 table spaces on DASD, on optical or tape volumes that reside in a library, and on optical or tape volumes that reside on a shelf. See also *storage hierarchy*.

offline control data set (OCDS). In DFSMShsm, a VSAM key-sequenced set that contains information about tape backup volumes and tape migration level 2 volumes.

OpenEdition MVS. See *OS/390 UNIX System Services*

OpenEdition MVS file system. See *OS/390 UNIX file system*.

optical disk drive. The mechanism used to seek, read, and write data on an optical disk. An optical disk drive can be operator-accessible, such as the 3995 Optical Library Dataserver, or stand-alone, such as the 9346 or 9347 optical disk drives.

optical library. A storage device that houses optical drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives.

optical volume. Storage space on an optical disk, identified by a volume label. See also *volume*.

OS/390 Network File System. A base element of OS/390, that allows remote access to MVS host processor data from workstations, personal computers, or any other system on a TCP/IP network that is using client software for the Network File System protocol.

OS/390 UNIX System Services (OS/390 UNIX). The set of functions provided by the SHELL and UTILITIES, kernel, debugger, file system, C/C++ Run-Time Library, Language Environment, and other elements of the OS/390 operating system that allow users to write and run application programs that conform to UNIX standards.

P

pageable link pack area (PLPA). An area of virtual storage containing SVC routines, access methods, and other read-only system and user programs that can be shared among users of the system. See also *link pack area*.

parallel sysplex. A systems complex that consists of multiple systems coupled together by hardware elements and system services. The systems must be defined by the COUPLExx members of SYS1.PARMLIB as being a parallel sysplex.

partitioned data set (PDS). A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE). A system-managed data set that contains an indexed directory and members that are similar to the directory and

members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

performance. (1) A measurement of the amount of work a product can produce with a given amount of resources. (2) In a DFSMS environment, a measurement of effective data processing speed with respect to objectives set by the storage administrator. Performance is largely determined by throughput, response time, and system availability.

permanent data set. A user-named data set that is normally retained for longer than the duration of a job or interactive session. Contrast with *temporary data set*.

pool. See *storage pool*.

pool storage group. A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also *storage group*.

primary data set. When referring to an entire data set collection, the primary data set is the first data set allocated. For individual data sets being stacked, the primary data set is the one in the data set collection that precedes the data set being stacked and is allocated closest to it.

primary space allocation. Amount of space requested by a user for a data set when it is created. Contrast with *secondary space allocation*.

primary storage. A DASD volume available to users for data allocation. The volumes in primary storage are called primary volumes. See also *storage hierarchy*. Contrast with *migration level 1* and *migration level 2*.

program management. The task of preparing programs for execution, storing the programs, load modules, or program objects in program libraries, and executing them on the operating system.

program object. All or part of a computer program in a form suitable for loading into virtual storage for execution. Program objects are stored in PDSE program libraries and have fewer restrictions than load modules. Program objects are produced by the binder.

pseudo optical library. A set of shelf-resident optical volumes associated with either a stand-alone or an operator-accessible optical disk drive; see also *real optical library*.

R

RACF always call. A term for the procedure used by DFSMSdfp to check all data sets automatically for discrete or generic RACF profiles to verify access authority.

real optical library. Physical storage device that houses optical disk drives and optical cartridges, and contains a mechanism for moving optical disks between a cartridge storage area and optical disk drives. Contrast with *pseudo optical library*.

record-level sharing. See *VSAM record-level sharing*.

recovery. The process of rebuilding data after it has been damaged or destroyed, often by using a backup copy of the data or by reapplying transactions recorded in a log.

relative byte address (RBA). In VSAM, the displacement 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.

removable media library. The volumes that are available for immediate use, and the shelves where they could reside.

residence mode (RMODE). The attribute of a load module or program object that identifies where in virtual storage the program is to reside (above or below 16 megabytes).

Resource Access Control Facility (RACF). An IBM-licensed program or a base element of OS/390, that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, logging the detected unauthorized attempts to enter the system, and logging the detected accesses to protected resources.

Resource Measurement Facility (RMF). An IBM licensed program or optional element of OS/390, that measures selected areas of system activity and presents the data collected in the format of printed reports, system management facilities (SMF) records, or display reports. Use RMF to evaluate system performance and identify reasons for performance problems.

retained lock. A lock protecting transaction updates when a problem delays transaction recovery of the updates. The retained status is cleared when transaction recovery completes.

S

secondary space allocation. Amount of additional space requested by the user for a data set when primary space is full. Contrast with *primary space allocation*.

sequential data striping. A software implementation of a disk array that distributes data sets across multiple volumes to improve performance.

service level (Storage Management Subsystem). A set of logical characteristics of storage required by a Storage Management Subsystem-managed data set (for example, performance, security, availability).

service-level agreement. (1) An agreement between the storage administration group and a user group defining what service-levels the former will provide to ensure that users receive the space, availability, performance, and security they need. (2) An agreement between the storage administration group and operations defining what service-level operations will provide to ensure that storage management jobs required by the storage administration group are completed.

Service Level Reporter (SLR). An IBM licensed program that provides the user with a coordinated set of tools and techniques and consistent information to help manage the data processing installation. For example, SLR extracts information from SMF, IMS, and CICS logs, formats selected information into tabular or graphic reports, and gives assistance in maintaining database tables.

sharing control data set. A VSAM linear data set that contains information DFSMSdfp needs to ensure the integrity of the data sharing environment.

shelf. A place for storing removable media, such as tape and optical volumes, when they are not being written to or read.

shelf location. A single space on a shelf for storage of removable media.

small-data-set packing (SDSP). In DFSMShsm, the process used to migrate data sets that contain equal to or less than a specified amount of actual data. The data sets are written as one or more records into a VSAM data set on a migration level 1 volume.

small-data-set-packing data set. In DFSMShsm, a VSAM key-sequenced data set allocated on a migration level 1 volume and containing small data sets that have been migrated.

SMS complex. A collection of systems or system groups that share a common configuration. All systems in an SMS complex share a common active control data set (ACDS) and a communications data set (COMMDS). The systems or system groups that share the configuration are defined to SMS in the SMS base configuration.

SMS configuration. A configuration base, Storage Management Subsystem class, group, library, and drive definitions, and ACS routines that the Storage Management Subsystem uses to manage storage. See also *base configuration* and *source control data set*.

SMS control data set. A VSAM linear data set containing configurational, operational, or communications information that guides the execution of the Storage Management Subsystem. See also *source control data set*, *active control data set*, and *communications data set*.

source. That portion of the DDM architecture that is necessary for a system to provide source support, allowing a system providing this support to request access to remote data on a remote (target) system. See also *client*.

source control data set (SCDS). A VSAM linear data set containing an SMS configuration. The SMS configuration in an SCDS can be changed and validated using ISMF. See also *active control data set* and *communications data set*.

spill storage group. An SMS storage group used to satisfy allocations which do not fit into the primary storage group.

storage administration group. A centralized group within the data processing center that is responsible for managing the storage resources within an installation.

storage administrator. A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class. A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage control. The component in a storage subsystem that handles interaction between processor channel and storage devices, runs channel commands, and controls storage devices.

storage director. In a 3990 Storage Control, a logical entity consisting of one or more physical storage paths in the same storage cluster. In a 3880, a storage director is equivalent to a storage path.

storage group. A collection of storage volumes and attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD, optical, or tape volumes

treated as a single object storage hierarchy. See also *VIO storage group, pool storage group, tape storage group, object storage group, object backup storage group*, and *dummy storage group*.

storage group category. A grouping of specific storage groups which contain the same type of data. This concept is analogous to storage pools in a non-system-managed environment.

storage hierarchy. An arrangement of storage devices with different speeds and capacities. The levels of the storage hierarchy include main storage (memory, DASD cache), primary storage (DASD containing uncompressed data), migration level 1 (DASD containing data in a space-saving format), and migration level 2 (tape cartridges containing data in a space-saving format). See also *primary storage*, *migration level 1*, *migration level 2*, and *object storage hierarchy*.

storage location. A location physically separate from the removable media library where volumes are stored for disaster recovery, backup, and vital records management.

storage management. The activities of data set allocation, placement, monitoring, migration, backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automates these processes for you, while optimizing storage resources. See also *Storage Management Subsystem*.

Storage Management Subsystem (SMS). A DFSMS/MVS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

storage pool. A predefined set of DASD volumes used to store groups of logically related data according to user requirements for service or according to storage management tools and techniques.

storage subsystem. A storage control and its attached storage devices. See also *tape subsystem*.

stripe. In DFSMS/MVS, the portion of a striped data set that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

striping. A software implementation of a disk array that distributes a data set across multiple volumes to improve performance.

sysplex. A set of MVS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

system data. The data sets required by MVS or its subsystems for initialization and control.

system group. All systems that are part of the same Parallel Sysplex and are running the Storage Management Subsystem with the same configuration, minus any systems in the Parallel Sysplex that are explicitly defined in the SMS configuration.

system-managed buffering for VSAM. A facility available for system-managed extended-format VSAM data sets in which DFSMSdfp determines the type of buffer management technique along with the number of buffers to use, based on data set and application specifications.

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, and space to applications. See also *DFSMS environment*.

system-managed tape library. A collection of tape volumes and tape devices, defined in the tape configuration database. A system-managed tape library can be automated or manual. See also *tape library*.

system-managed volume. A DASD, optical, or tape volume that belongs to a storage group. Contrast with *DFSMShsm-managed volume* and *DFSMSrmm-managed volume*.

system management facilities (SMF). A component of MVS that collects input/output (I/O) statistics, provided at the data set and storage class levels, which helps you monitor the performance of the direct access storage subsystem.

system programmer. A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

T

tape configuration database. One or more volume catalogs used to maintain records of system-managed tape libraries and tape volumes.

tape librarian. The person who manages the tape library.

tape library. A set of equipment and facilities that support an installation's tape environment. This can include tape storage racks, a set of tape drives, and a set of related tape volumes mounted on those drives. See also *system-managed tape library* and *automated tape library*.

Tape Library Dataserver. A hardware device that maintains the tape inventory associated with a set of tape drives. An automated tape library dataserver also manages the mounting, removal, and storage of tapes.

tape mount management. The methodology used to optimize tape subsystem operation and use, consisting of hardware and software facilities used to manage tape data efficiently.

tape storage group. A type of storage group that contains system-managed private tape volumes. The tape storage group definition specifies the system-managed tape libraries that can contain tape volumes. See also *storage group*.

tape subsystem. A magnetic tape subsystem consisting of a controller and devices, which allows for the storage of user data on tape cartridges. Examples of tape subsystems include the IBM 3490 and 3490E Magnetic Tape Subsystems.

tape volume. A tape volume is the recording space on a single tape cartridge or reel. See also *volume*.

target. That portion of the DDM architecture that is necessary for a system to provide target support, allowing a system providing this support to receive and process requests from a remote (source) system. See also *server*.

temporary data set. An uncataloged data set whose name begins with & or &&, that is normally used only for the duration of a job or interactive session. Contrast with *permanent data set*.

threshold. A storage group attribute that controls the space usage on DASD volumes, as a percentage of occupied tracks versus total tracks. The *low migration threshold* is used during primary space management and interval migration to determine when to stop processing data. The *high allocation threshold* is used to determine candidate volumes for new data set allocations. Volumes with occupancy lower than the high threshold are selected over volumes that meet or exceed the high threshold value.

Transmission Control Protocol/Internet Protocol (TCP/IP). A suite of protocols designed to allow communication between networks regardless of the technologies implemented in each network.

U

unit affinity. Requests that the system allocate different data sets residing on different removable volumes to the same device during execution of the step to reduce the total number of tape drives required to execute the step. Explicit unit affinity is specified by coding the UNIT=AFF JCL keyword on a DD statement. Implicit unit affinity exists when a DD statement requests more volumes than devices.

unit control block (UCB). A control block in storage that describes the characteristics of a particular I/O device on the operating system.

use attribute. (1) The attribute assigned to a DASD volume that controls when the volume can be used to allocate new data sets; use attributes are *public*, *private*, and *storage*. (2) For system-managed tape volumes, use attributes are *scratch* and *private*.

user group. A group of users in an installation who represent a single department or function within the organization.

user group representative. A person within a user group who is responsible for representing the user group's interests in negotiations with the storage administration group.

V

validate. To check the completeness and consistency of an individual ACS routine or an entire SMS configuration.

virtual input/output (VIO) storage group. A type of storage group that allocates data sets to paging storage, which simulates a DASD volume. VIO storage groups do not contain any actual DASD volumes. See also *storage group*.

Virtual Tape Server (VTS). This subsystem, integrated into the Magstar 3494 Tape Library, combines the random access and high performance characteristics of DASD with outboard hierarchical storage management and virtual tape devices and tape volumes.

vital records. A data set or volume maintained for meeting an externally-imposed retention requirement, such as a legal requirement. Compare with *disaster recovery*.

vital record specification. Policies defined to manage the retention and movement of data sets and volumes for disaster recovery and vital records purposes.

volume. The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume*, *optical volume*, and *tape volume*.

volume mount analyzer. A program that helps you analyze your current tape environment. With tape mount management, you can identify data sets that can be redirected to the DASD buffer for management using SMS facilities.

volume status. In the Storage Management Subsystem, indicates whether the volume is fully available for system management:

- "Initial" indicates that the volume is not ready for system management because it contains data sets that are ineligible for system management.
- "Converted" indicates that all of the data sets on a volume have an associated storage class and are cataloged in an integrated catalog facility catalog.
- "Non-system-managed" indicates that the volume does not contain any system-managed data sets and has not been initialized as system-managed.

VSAM record-level sharing (VSAM RLS). An extension to VSAM that provides direct record-level sharing of VSAM data sets from multiple address spaces across multiple systems. Record-level sharing uses the System/390 Coupling Facility to provide cross-system locking, local buffer invalidation, and cross-system data caching.

VSAM sphere. The base cluster of a VSAM data set and its associated alternate indexes.

VSAM volume data set (VVDS). A data set that describes the characteristics of VSAM and system-managed data sets residing on a given DASD volume; part of an integrated catalog facility catalog. See also *basic catalog structure* and *integrated catalog facility catalog*.

Index

Numerics

31-bit addressing
  access methods 203
3495 Model M10 Tape Library Dataserver
  migration considerations 185
3590 Tape Subsystem 96
3995-Cxx Optical Library
  description 103
3995-SW4
  8x-optical drive 64

A

ABARS (aggregate backup and recovery support) 265
  64 tasks 85
  ABARSOPTIMIZE parameter 82
  activity log deletion 81
  ARCBEEXT installation exit, enhancements 93
  ARECOVERTGTGDS parameter 82
  changes to ISMF aggregate group definition panels 88
  CPU time in WWFSR and ABR records 87
  file stacking 83
  GDG base name in allocate statement 84
  recovering applications 265
  recovering data sets 265
ABARSOPTIMIZE parameter 82
abend
  jobs using system-managed data sets 220
ABR records, CPU time 87
Access Method Services (IDCAMS)
  ALTER 116
  DCOLLECT 115
  DEFINE 116
    USERCATALOG 116
  DELETE
    GDG FORCE 116
  LISTCAT 116, 159
  REPRO 116
    MERGECAT 116
  space constraint relief 109
account information, data set 122
accounting, ABARS ISMF definition panels 88
ACDS (active control data set)
  description 29
activity log deletion, ABARS 81
adding
  file system 237
aggregate group
  compatibility between DFHSM and DFSMShsm 32
allocation
  distribution library 214
  ISMF load module libraries 213
  PDSE data sets 229
alter without recall 93
alternate offerings, DFSMS/MVS 9
APPC (advanced program-to-program communication)
  Distributed FileManager 180
APPLYINCREMENTAL command 263
ARCBEEXT installation exit, enhancements 93
ARCINEXT installation exit changes 94
ARCRPEXT, return priority exit 130
ARCSDEXT, shutdown exit 130
ARCTVEXT
  planning 253
ARCTVEXT installation exit changes 94
ARECOVERTGTGDS parameter 82
AUDIT command improvements 94
authorization
  define catalog alias 120
  delete catalog alias 120
  planning for DFSMSrmm 252
availability management 261
  description 261
  sample plan 282

B

backing up
  file system 239
backup
  HFS file 239
backup versions, number of 126
backup-while-open
  planning 177
Backup-While-Open of IMS data sets 124
backup, DFSMShsm 262
  cycle 262
  media 263
  parameters 262
  tasks 262
  volume cleanup 263
  window 262
bin number 247
binder
  planning for new functions 137
block
  simulated 227
book
  DFSMS/MVS 293
  MVS/ESA 301
  online library (CD-ROM) 299

© Copyright IBM Corp. 1979, 1999

book *(continued)*
ordering 293
SSL 298
System/390 Sysplex 300
books
DFSMS/MVS 10
BSAM (basic sequential access method)
compression 191
HFS file access 117
BSP macro
PDSE restrictions 233

C

cache structure
VSAM RLS
DCOLLECT output 115
description 161
catalog
account field 122
alias
authorizing to define 120
authorizing to delete 120
symbolic substitution 36
attribute extension cell 122
catalog search interface 110
CVOL restrictions 157, 197, 219
CVOL supported 126
enhanced catalog sharing (ECS) 38
MODIFY 120
performance 36
sharing 30
VSAM catalog restrictions 157, 219
VSAM catalog supported 126
CDS RLS serialization 85
Character Data Representation Architecture (CDRA)
planning 121
checkpoint
data set
partitioned data set or PDSE 233
security 32, 287
shared volumes 32
checkpointed data sets, DFSMShsm support 91
checkpointed data sets, protection of 79
CICS
required level 15
CICSVR MVS/ESA
required level 15
CLIST
example 216
commands
displaying OAM XCF status 64, 65
displaying SETOPT and SETOAM settings 64, 65
new and changed (OAM) 64
reformatting a 3995 optical disk 64
relabeling a 3995 optical disk 64, 65
commands *(continued)*
updating SETOPT and SETOAM settings 64, 66
COMMDS (communications data set)
description 29
compression
SAM tailored 107
sequential data sets 191
VSAM data sets 198
concatenated parmlib support 131
concurrent copy 204, 260
DFSMShsm control data sets 260
concurrent copy, using for volume dumps 130
control data set
DFSMSrmm
definition of 248
global resource serialization 249
GUARANTEED SPACE attribute 249
placement of 249
control data sets 259
about 259
allocating 259
availability 260
backing up 260
DFSMShsm 259
journal 259
log 259
improving performance 249
journal data set, backing up 260
conversion
partitioned data set to PDSE 230
PDSE to partitioned data set 230
corequisite licensed programs 14
coupling facility (CF)
cache structure
VSAM RLS 161
cpio shell command
backing up HFS file 239
CPOOL performance improvement 125
CPU time in WWFSR and ABR records 87
creating
file system 236, 237
root file system 236
CVOL restrictions 157, 197, 219
CVOL supported 126

D

DASD (direct access storage device)
extents 225
initialization 16
data management
description 4
data set 31
account information 122
DFSMShsm control 279
backup 279
maintaining, sample plan 281

data set *(continued)*
DFSMShsm on-site recoverability 261
distribution library 214
empty 205
extended format 205
sequential 234
VSAM 234
HFS
creating 236
hierarchical file system
planning 235
IMBED
not supported 36
non-recoverable VSAM, RLS 161, 167, 169, 172
non-SMS-managed 270
pooling 270
performance statistics 204
recoverable VSAM, RLS 161, 167, 172
recovery 263
sharing 31
SMS-management 219
striping 205
data set dump
backing up file system 239
DB2 (DATABASE 2)
required level 14
DCOLLECT enhancements 95
DEB table expansion 40
defining
root file system 236
deleting
file system 239
DESERV 138
device
combining in storage groups 182
maximum number 196
device management
description 4
modes 17
supported by MVS/ESA 16
device requirements 16
Device Support Facilities
storage requirements 16
DFDSS
coexistence between releases 32
DFHSM
coexistence between releases 32
DFM
DataAgent 78
DFM/MVS
DataAgent 78
DFSMS
catalog management 219
configuration planning 219
environment 3
implementing 220
DFSMS *(continued)*
indexed VTOC 220
integrated catalog facility catalogs 219
Optimizer 6
planning tasks 221
DFSMS/MVS
corequisite licensed programs 14
DFDSS coexistence 32
DFHSM coexistence 32
distribution package 9
functional components 3
licensed program requirements 13
ordering 9
publications 10, 293
requisite programs for full-function support 14
DFSMSdfp
functions of 4
Year 2000 support 25
DFSMSdss
coexistence with DFDSS 32
enhanced protection of checkpointed data sets 79
functions of 4
IMS Backup-While-Open 124
logical processing for RELEASE 41
SELECTMULTI keyword 80
SnapShot support 41
Stand-Alone Services function 80
STORGRP parameter 42
VOLCOUNT Parameter 123
Year 2000 support 26
DFSMSdss multivolume selection 80
DFSMShsm
allocating data sets 280
sample plan 280
APPLYINCREMENTAL command 263
ARCINEXT installation exit changes 94
ARCTVEXT installation exit changes 94
AUDIT command improvements 94
backup 262
cycle 262
media 263
parameters 262
tasks 262
volume cleanup 263
window 262
CDS extended addressability 44
CDS RLS serialization 85
checkpointed data set protection 91
coexistence with DFHSM 32
concatenated parmlib support 131
control data sets 258
allocating 259
backup copy 258
BCDS (backup control data set) 258
description of 258
MCDS (migration control data set) 258
OCDS (offline control data set) 258

DFSMShsm *(continued)*
coordinating processing functions 277
CPOOL performance improvement 125
customizing 276
customizing, sample plan 285
data set 279
control, backup 279
data set recovery 263
deactivating default RLEs 131
device selection 274
DFSMShsm-owned volumes 260
ML1 260
ML2 260
recovery 264
disaster recovery, off-site 265
dump analysis elimination 88
dump class 264
dump cycle 263
dump tasks 264
dump window 264
duplex tape function 89
eliminating unnecessary I/O 131
enhanced HBACKDS command 130
expanded data set name filter 130
EXPIREBV command 263
FSR enhancements 94
functional verification procedure 275
functions of 5
general improvements, 1.5 47
general improvements, V1R4 93
greater than 13 backup versions 126
HRECOVER command 263
improvements, performance 130
improvements, to functions 130
improvements, usability 130
improvements, Version 1 Release 3 130
installation verification procedure 275
interval migration 278
interval migration, checking 131
introduction 5
journal data set 259
journal data sets 258
log data set 259
log data sets 258
migrate prefix HSM and SYS1 131
multi-processor environment 257
multiple buffer usage 130
multiple tape buffers 179
multitasking recovery 128
on-site recoverability 261
on-site volume recovery 263
overflow ML1s 130
overhead, reducing 45
parallel sysplex, enabling 46
planning to run with DFSMSrmm 252
primary processor 257

DFSMShsm *(continued)*
processing functions 277
recycle enhancements 178
return priority exit (ARCRPEXT) 130
scratch pool 273
global 273, 275
specific 273, 275
SDSP data sets 268
backing up 269
description 268
recovering 270
reorganizing 270
secondary host promotion 43
shutdown exit (ARCSDEXT) 130
single-processor environment 257
space checking for TMM 130
space management 266
automatic primary 266
description 266
parameters 266
primary 279
specifying as single host at start up 131
SWAPLOG PDA command 131
tape 273
recycling 273
tape I/O improvements 179
tape library 271
tape management 271
description 271
output processing, selection 273
scratch pool 273
testing 275
tuning options 276
tuning, sample plan 285
virtual storage constraint relief 180
volume dumps using concurrent copy 130
VSAM enhancements 73
Year 2000 support 26, 132
DFSMSPKG, started task 10
DFSMSrmm
130 enhancements 132
140 enhancements 95
150 enhancements 49, 50, 51, 52, 53, 57, 67
application programming interface 49
authorization 252
catalog status tracking 50
code change in ARCTVEXT 94
control data sets 248
DASD space requirements 248
DFSMShsm and DFSMSrmm interaction 252
disposition control 51
EDGTVEXT 95
evaluating removable media management needs 247
functions of 5
ISO/ANSI label support 57

DFSMSrmm *(continued)*
open, close, end-of-volume dependencies 255
planning for operating 252
problem determination log data sets 248
programming interfaces 253
security 252
subsystem enhancements 52
TSO subcommand enhancements 53
user exits 253
utilities 254
virtual tape server support 67
vital record specification enhancements 53
DFSORT (Data Facility Sort)
description 7
required level 14
directory
PDSE
services (DESERV) 138
structure 226
disaster recovery 263
ABARS 265
off-site 265
on-site 263
distributed data access
description 4
Distributed FileManager
migration considerations 180
DSCB (data set control block)
compatibility between MVS/XA and MVS/ESA 33
last-referenced date (LRD) at close 112
dump analysis elimination 88
dump data set
backing up file system 239
duplex tape function 89

E

EDGPDOX PDA trace data set 251
EDGPDOY PDA trace data set 251
ejecting volumes 68
EMODVOL1, using 254
enabling, DFSMS/MVS 10, 137
enhanced protection of checkpointed data sets 79
environment
MVS 3
erase-on-scratch
DFSMSdss 289
RACF security 288
EREP (environmental record editing and printing)
required level 14
expiration date
overriding 133, 190
EXPIREBV command 263
exporting volumes 68
extended addressability
VSAM data sets 157
extended format
sequential compression 191
sequential data sets 234
VSAM data sets 157, 198, 234
VSAM partial release 200
extended remote copy (XRC) 134
extents
DASD 225

F

features, DFSMS/MVS
enabling 10, 137
ordering 9
file stacking, ABARS 83
file system
adding 237
backing up 239
creating additional 237
managing activity 238
managing size 238
migrating 239
mounting 237
moving 238
OS/390 UNIX System Services 134, 189, 235
removing 239
transporting 238
FSR enhancements to DFSMShsm 94
functional components
DFSMS/MVS 3
functional components, DFSMS/MVS
enabling 10, 137
ordering 9
functional verification procedure 275

G

GDG base name in ABARS allocate statement 84
generation data group base name in ABARS allocate statement 84
global resource serialization
control data set 249
greater than 13 versions 126
GRS (global resource serialization)
checkpoint data sets 32
GUARANTEED SPACE attribute
control data set 249
journal 251

H

hardware requirements 13
HBACKDS command enhancements 130
HCD (Hardware Configuration Definition)
device initialization 16

HFS (hierarchical file system) data set
backing up 239
backing up files 239
BSAM access 117
deleting 239
directory
creating additional 237
file system
adding 237
mounting 237
managing activity 238
managing size 238
migrating 239
moving 238
performance changes 55
planning 134, 189, 235
QSAM access 117
recovering 239
standards compliance 134
transporting 238
VSAM access 117
High Level Assembler/MVS
required level 13
HRECOVER command 263

I

I/O devices, supported 16
IBM 3995 Optical Library Dataserver
mixing WORM and rewritable media 186
ICKDSF (Device Support Facilities)
required level 14
IGDDFPKG, updating 10, 137
IGXMSGEX
planning 253
IMBED
data set option
not supported 36
importing volumes 69
improvements, DFSMShsm 130
IMS Backup-While-Open
DFSMSdss support 124
IMS/ESA
required level 15
incremental backup 275
unattended system 275
indexed VTOC
SMS 220
initial access response time (IART)
targeting devices 182
inserting volumes 69
installation 275
verification procedure 275
installation exits
ARCTVEXT 253
EMODVOL1 253
installation exits *(continued)*
IGXMSGEX 253
OMODVOL1 253
installation requirements 15
integrated catalog facility
SMS 219
Integrated Cryptographic Service Facility/MVS (ICSF/MVS)
required level 14
interval migration 267, 278
interval migration, checking 131
inventory management
definition of 254
ISMF (Interactive Storage Management Facility)
description 213
distribution libraries
performance 214
where to store 214
graphical user interface 136
ISMF changes for ABARS accounting 88
ISPF 214
load library placement 213
load module libraries
access control 213
performance 213
placement considerations 213
SYS1.LINKLIB, using 213
where to store 214
Menus to invoke 217
optional products 213
protecting functions 214
reduce out-of-space condition (X37 abend) 109
required products 213
table output library 214
TSO users 215
ISO/ANSI Version 4 Tape 57
ISPF (Interactive System Productivity Facility)
table libraries 214
ISPTABL 214
ISPTLIB 214

J

JES2 270
DFSMS/MVS required level 14
pooling 270
JES3 271
DFSMS/MVS required level 14
pooling 271
SMS 32 name support 141
journal data set 259
journal, DFSMSrmm
calculating space for 250
definition of 248
GUARANTEED SPACE attribute, using 251
placement of 251

L

Language Environment for MVS and VM
  required level 14
library
  tape conversion activities 243
library disabling processing (DISABLE)
  description 100
licensing, DFSMS/MVS 10, 137
log data set 259

M

machine requirements
  enhanced dynamic cache management 205
manual
  DFSMS/MVS 293
  MVS/ESA 301
  online library (CD-ROM) 299
  ordering 293
  SSL 298
  System/390 Sysplex 300
manual tape library
  planning 185
manuals
  DFSMS/MVS 10
migrate prefix HSM and SYS1 131
migrating
  file system 239
Migration Considerations
  access method services 115
  backup-while-open 177
  Character Data Representation Architecture (CDRA) 121
  data set rename changes 148
  DCOLLECT 111
  DFSMShsm improvements 130
  DFSMShsm recycle enhancements 178
  DFSMShsm tape I/O improvements 179
  DFSMShsm virtual storage constraint relief 180
  Distributed FileManager target server 180
  enhanced SMS volume selection 182
  extended remote copy (XRC) 134
  IBM 3495 Model M10 185
  IBM 3590 96
  IBM 3995 Models 133, 113 186
  ISMF GUI 136
  Naviquest 97
  O/C/EOV 105
  OAM objects on tape 187
  OAM RMF transaction reporting 188
  OS/390 UNIX System Services 134, 189
  overriding expiration dates 190
  PDSE 135
  program management 106
  program management extensions 137
Migration Considerations *(continued)*
  QSAM and BSAM compression 191
  SMS 32 name support 141
  SMS construct access checking 148
  SMS system groups 193
  tape mount management 149
  UCB constraint relief 196
  UCB virtual storage constraint relief 154
  VSAM
    compression 198
    extended addressability 157
    partial release 200
    record-level sharing (RLS) 161
millisecond response time (MSR)
  targeting devices 182
ML1 (migration level 1) 268
  threshold value 268
  volumes, description 268
ML1, overflow volumes 130
ML2 (migration level 2) 268
  media 268
MODIFY
  catalog functions 120
MODIFY OAM Command
  description 100
mounting
  file system 237
moving
  file system 238
multiple buffer usage 130
multisystem environment
  catalog sharing 30
  coexistence with MVS/DFP and MVS/XA DFP 29
multitasking recovery 128
MVS/ESA
  I/O devices 16
  VTOC compatibility with MVS/XA 33
MVS/ESA publications 301
MVS/ESA System Product
  compatible DFSMS/MVS version 14
MVS/XA
  VTOC compatibility with MVS/ESA 33

N

national language support
  implementing with CDRA interfaces 121
NaviQuest 60, 97
NetView
  transferring data between systems 31
non-recoverable data set
  VSAM RLS 161, 167, 169, 172
non-scratch tape 247
null record segments 232

O

OAM (object access method)
  objects on tape 187
  parallel sysplex support 61
  RMF transaction reporting 188
OAM DUMP Command
  description 98
OAM Move Volume Utility
  available function 100
  description 100
  scenarios 100, 101
OAM QUERY Command
  description 98, 99
OAM RESTART Command
  description 98
  sample procedure statement 98, 99
OAM System Management Facility (SMF)
  description 102
  optical and tape activities 102
  OSREQ functions 102
objects
  tape, writing to 187
OMODVOL1, using 254
online library (CD-ROM) 299
open, close, end-of-volume
  DFSMSrmm considerations 255
open,close,end-volume
  serviceability 105
operating system
  MVS/DFP compatibility 29
  MVS/XA DFP compatibility 29
optical library dataserver
  mixing WORM and rewritable media 186
optimizer code change in ARCINEXT 94
Optimizer, DFSMS 6
OS/390 Network File System
  description 7
OS/390 UNIX System Services
  DFSMS/MVS planning 134, 189
  file system planning 235
overflow ML1s 130
overriding
  expiration date 133, 190

P

packaging, DFSMS/MVS 9
partial release
  VSAM data sets 200
passwords
  not supported 36
PAX shell command
  backing up HFS file 239
PDSE
  directory
    services (DESERV) 138
PDSE *(continued)*
  directory *(continued)*
    structure 226
PDSE (partitioned data set extended)
  advantages 225
  block size 227
  BSP macro 233
  buffer 227
  defining 227
  directory space 229
  directory structure 226
  extents 225
  hiperspace size 135
  keyed BSAM 233
  migration from partitioned data set 230
  performance 227
  record processing 227
  restrictions 233
  secondary space 229
  sharing access 231
  storage allocation 229
  storage considerations 230
planning
  file system 235
pooling 270
  JES2 270
  JES3 271
  non-SMS-managed volumes 270
product packaging,DFSMS/MVS 9
products, DFSMS/MVS required 13
program management
  binder 106
  compressed data sets 32
  description 4
  loader 106
  planning for new functions 106, 137, 206
program object
  planning for new functions 137
programming interfaces
  planning 253
pseudo library
  concept within OAM 62
PSF for OS/390
  required level 14
publication
  DFSMS/MVS 293
  MVS/ESA 301
  online library (CD-ROM) 299
  ordering 293
  SSL 298
  System/390 Sysplex 300
publications
  DFSMS/MVS 10
  ordering 10

Q

QSAM (queued sequential access method)
  compression 191
  HFS file access 117

R

RACF (Resource Access Control Facility)
  always-call 287
  checkpoint data set 32
  DFSMSdss resources 289
  DFSMShsm commands 290
  DFSMShsm resources 289
  DFSMSrmm resources 288
  discrete profile 287
  erase-on-scratch 288
  generic profile 287
  ISMF program control 214
  program control 214
  required level 14
  security 287
  VSAM data sets 287
  VSAM RLS 288
rack number 247
record-level sharing (RLS), VSAM 161
recoverable data set
  VSAM RLS 161, 167, 172
recovering
  HFS data set 239
recovery, multitasking 128
recycle enhancements, DFSMShsm 178
removable media
  management 5
removing
  file system 239
requirements
  installation 15
  storage 15
RETAIN
  install indexes 16
return priority exit (ARCRPEXT) 130
RMF (resource measurement facility)
  required level 14
  transaction reporting for OAM 188
root file system
  creating 236
  defining 236

S

SAM Tailored Compression 107
sample plan, DFSMShsm 279
  allocating DFSMShsm data sets 280
  availability management 282
  customizing and tuning 285
  maintaining control data sets 281
sample plan, DFSMShsm *(continued)*
  setup 279
  space management 283
SCDS (source control data set)
  description 29
scratch mounts
  definition of 247
  non-scratch tape, definition of 247
scratch pool 273
  global 273, 274
  specific 273, 274
SDSP (small data set packing) 268
  data set 268
    backing up 269
    description 268
    recovering 270
    reorganizing 270
  requirements 268
security
  checkpoint data set 32
  DFSMSdss resources 289
  DFSMShsm commands 290
  DFSMShsm resources 289
  DFSMSrmm 252
  DFSMSrmm resources 288
  erase-on-scratch 288
  RACF 287
  VSAM data sets 287
  VSAM RLS 288
SELECTMULTI keyword 80
sequential data set
  compressing 191
  extended format 191
service agreements 11
setup plan, DFSMShsm 279
  sample 279
shared volumes
  checkpoint data sets on 32
sharing
  data sets 31
  multisystem considerations 29
  transferring between systems 31
shelf location, DFSMSrmm
  definition of 247
shutdown exit (ARCSDEXT) 130
simulated blocks, record processing 227
SMP/E (System Modification Program Extended)
  APPLY, ISMF libraries used 213
  required level 13
  storage requirements 15
SMS (Storage Management Subsystem)
  construct access checking 148
  control data set changes 39
  data class
    REUSE parameter 148
    SPEED parameter 148

SMS (Storage Management Subsystem) *(continued)*
JES3 considerations 141
management class
ACS routine redriven for rename 148
sysplex planning 193
system group considerations 141, 193
system names 141
virtual storage requirements 220
volume selection 182
X37 abend reduction 109
space checking for TMM 130
space management, DFSMShsm 266
automatic primary 266
automatic secondary 267
description 266
interval migration 267
parameters 266
sample plan 283
secondary 278
volume migration 267
SPACE parameter
PDSE 229
Stand-Alone Services function of DFSMSdss
improvements from previous Stand-Alone function 80
introduction 80
purpose 80
standards
OS/390 UNIX System Services 134
storage class
initial access response time
targeting devices 182
millisecond response time
targeting devices 182
storage constraint relief, DFSMShsm 180
storage group
combining devices 182
storage management
description 4
storage requirements, DFSMSrmm
control data set 248
Device Support Facilities 16
for DFSMS/MVS 15
journal 250
SMP/E 15
striping 234
SWAPLOG PDA command 131
sysplex
coexistence considerations 194
program requirements 14, 194
system groups 193
system group
planning 193
system-managed buffering (SMB)
VSAM 113

T

table library
ISPF 214
tape
copy processing 279
DFSMShsm 271
management 271, 273
multiple tape buffers 179
recycle enhancements 178
recycling 273
tape I/O improvements 179
DFSMSrmm–managed 5
environment 274
non-SMS-managed 274
ISO/ANSI Version 4 57
library 274
attachment RPQ 206
catalog sharing 32
defining 17
Hardware Configuration Definition 206
SMS-managed 274
library conversion activities 243
non-SMS-managed 274
partially unattended operation 274
removable media management 5
system-managed 243
Using VMA to analyze 242
tape I/O improvements, DFSMShsm 179
tape mount management (TMM)
description 4
DFSMShsm role in 5
implementation plan 242
planning 149
space checking 130
TAR shell command
backing up HFS file 239
transporting
file system 238
TSO (time sharing option)
execution data sets 215
ISMF access menus 217
ISMF availability 215
modifying logon procedure 215
required level of TSO/E 14
TSO TRANSMIT and RECEIVE commands 31
tuning options 276
DFSMShsm 276

U

UCB (unit control block)
4-digit 196
constraint relief 196
maximum number 196

unit control block (UCB)
  above 16MB 154
  captured 154
  constraint relief 154
  maximum number 154
UNIX standards
  HFS compliance 134
user exits
  planning 253
utilities
  planning 254
UTILMSG keyword, DFSMSdss 124

V

virtual storage
  SMS 220
virtual storage constraint relief 180
virtual storage constraint relief, UCB
  planning 154
virtual storage requirements
  DFSMS/MVS 15
  IBM licensed programs 15
Virtual Tape Server
  OAM enhancements 67
vital record specification enhancements 53
VMA (volume mount analyzer) 242
VOLCOUNT Parameter (DFSMSdss) 123
volume 263
  DFSMSrmm 255
  initialization 16
  on-site recovery, DFSMShsm 263
volume dumps using concurrent copy 130
Volume Mount Analyzer 207
Volume Stacking
  description 104
  tape volume cache 104
  Virtual Tape Server (VTS) 104
VSAM (virtual storage access method)
  candidate volume space 72
  catalog search interface 110
  compression planning 198
  DCOLLECT enhancements 111
  extended addressability 72, 157
  extended format 72, 157, 198
  HFS file access 117
  last-referenced date (LRD) at close 112
  partial release 72, 200
  record-level sharing (RLS) 161
  system-managed buffering 72, 113
VSAM catalog restrictions 157, 219
VTAM (Virtual Telecommunications Access Method)
  Distributed FileManager 180
VTOC (volume table of contents)
  compatibility between MVS/XA and MVS/ESA 33
  index 220

U.S.A.

W

WWFSR records, CPU time 87

X

X37 abend reduction 109
XMIT (TSO TRANSMIT) command 31
XRC (extended remote copy) 55, 134
  PPRC CGROUP command 55
  XRC Planned Outage Support 55

Y

Year 2000 support 132
  DFSMS/MVS readiness 25
  DFSMSdfp changes 25
  DFSMSdss changes 26
  DFSMShsm changes. 26
  migration actions 27
  migration assistance 28
  testing 27

Readers' Comments — We'd Like to Hear from You

DFSMS/MVS Version 1 Release 5
Planning for Installation

Publication No. SC26-4919-06

Overall, how satisfied are you with the information in this book?

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Overall satisfaction	☐	☐	☐	☐	☐

How satisfied are you that the information in this book is:

	Very Satisfied	Satisfied	Neutral	Dissatisfied	Very Dissatisfied
Accurate	☐	☐	☐	☐	☐
Complete	☐	☐	☐	☐	☐
Easy to find	☐	☐	☐	☐	☐
Easy to understand	☐	☐	☐	☐	☐
Well organized	☐	☐	☐	☐	☐
Applicable to your tasks	☐	☐	☐	☐	☐

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. _____

Readers' Comments — We'd Like to Hear from You
SC26-4919-06

Cut or Fold
Along Line

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

IBM Corporation
RCF Processing Department
G26/050
5600 Cottle Road
San Jose, CA 95193-0000

Fold and Tape | **Please do not staple** | Fold and Tape

SC26-4919-06

Cut or Fold
Along Line

File Number: S370/S390-34
Program Number: 5695-DF1
5647-A01

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC26-4919-06

Spine information:

IBM DFSMS/MVS Version 1 Release 5 Planning for Installation