DFSMS Using the New Functions
Seventh Edition, September 2009

This edition applies to Version 1 Release 11 of z/OS® (5694-A01) and to all subsequent releases and modifications until otherwise indicated in new editions.

This edition replaces SC26-7473-05.

IBM® welcomes your comments. A form for readers’ comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation
MHVRCFS, Mail Station P181
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
Your International Access Code +1+845+432-9405

IBMLink™ (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrdfs@us.ibm.com

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:
• Title and order number of this document
• Page number or topic related to your comment

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.

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

About This Information ........................ ix
Required product knowledge .................. ix
Referenced documents ......................... ix
Accessing z/OS DFSMS information on the Internet ........ x
The z/OS Basic Skills Information Center .............. x

Summary of Changes .......................... xi
Summary of Changes for SC26-7473-06 z/OS Version 1 Release 11 ........ xi
New Information ............................... xi
Changed Information ........................... xi
Moved Information ................................ xi
Summary of Changes for SC26-7473-05 z/OS Version 1 Release 10 ........... xii
New Information ............................... xii
Changed Information ........................... xii
Summary of Changes for SC26-7473-04 z/OS Version 1 Release 9 ............ xii
New Information ............................... xii
Deleted Information ........................... xii
Summary of Changes for SC26-7473-03 z/OS Version 1 Release 8 ............ xii
New Information ............................... xii
Changed Information ........................... xii
Summary of Changes for SC26-7473-02 z/OS Version 1 Release 7 ............ xiii
New Information ............................... xiii
Changed Information ........................... xiii

Introduction .................................. xv
What do I need to do to use the z/OS DFSMS enhancements? ............... xv
How can z/OS V1R11 DFSMS help me? .................. xvi
How can z/OS V1R10 DFSMS help me? .................. xvii
How can z/OS V1R9 DFSMS help me? .................. xviii
How can z/OS V1R8 DFSMS help me? .................. xix
How can z/OS V1R7 DFSMS help me? .................. xx

Part 1. Using New DFSMS Functions in z/OS V1R11 ................ 1

Chapter 1. Using extended address volumes enhancements ............. 3
Planning and installation ................................ 3
Planning to use extended address volumes enhancements ............... 3
Setting up support for extended address volumes enhancements ....... 4
Administering ................................... 5
DFSMSShfp administration ................................ 5
Application programming ................................ 5
Modifying applications to use extended address volumes enhancements .... 5
Diagnosing problems with extended address volumes enhancements .... 6
Chapter 2. Using the access method services DELETE MASK option .... 7
Chapter 3. Using the SMS enhancements .......................... 9
Activating data set separation by volume ..................... 9
Using the striping volume selection enhancements .................. 10
Chapter 4. Using the catalog enhancements ......................... 11
Administering ................................... 11
Using the catalog check ................................ 11
Redefining catalogs to remove IMBED and REPLICATE attributes ..... 12
Chapter 5. Using the Open/Close/EOV enhancements ................ 13
Planning and Installation ................................ 13
Planning and implementing the OCE enhancements .................. 13
Chapter 6. Using the object access method enhancements ............ 15
System Programming ................................ 16
Specifying object sizes up to 2000 MB for tape .................. 16
Enabling and specifying the level of deletion-protection ............ 16
Administering ................................... 17
Enabling larger object sizes for tape ........................ 17
Defining deletion- and retention-protection for object storage groups .... 17
Using the CHGCOL utility to modify OAM collection defaults ....... 18
New return codes to be processed by the CBRUXSAE security authorization exit ... 18
Application Programming ................................ 19
Specifying storage for large objects in parts .................. 19
Specifying deletion-hold status and extending expiration dates for objects .... 19
Using expanded wildcards in OSREQ QUERY requests ............ 20
Operating ..................................... 20
Displaying deletion- and retention-protection status ............... 20
Displaying IEFSSNxx parameter settings for OAM1 .............. 21
Diagnosing .................................... 21
Diagnosing problems related to expanded object sizes ............. 21
Chapter 7. Using the DFSMSShsm enhancements .................... 23
Planning and Installation ................................ 25
About This Information

This information helps you to understand the z/OS V1R11, V1R10, V1R9, V1R8, and V1R7 enhancements for DFSMSdfp and the DFSMS features. Appropriate cross-references are provided throughout the document if you need more background information about a specific concept or task.

If you are an experienced system programmer, application programmer, storage administrator, or member of the information technology team responsible for understanding and using the DFSMS enhancements, this information provides you with enough information to understand and apply the DFSMS enhancements at your site. This information is also helpful for anyone who wants a quick explanation of the tasks associated with these enhancements.

For information about accessibility features of z/OS, for users who have a physical disability, please see “Accessibility,” on page 189.

Required product knowledge

To use this book effectively, you should have an in-depth knowledge of your current DFSMSdfp™ installation, including the programming, configuration, and procedures used at your site. Use this document in combination with z/OS Migration, which covers the required tasks for migrating to each enhancement.

Referenced documents

The following publication has an overview of all enhancements for z/OS® DFSMS:

<table>
<thead>
<tr>
<th>Publication Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS Introduction and Release Guide</td>
<td>GA22-7502</td>
</tr>
</tbody>
</table>

The following publications are referenced in this information:

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS Migration</td>
<td>GA22-7499</td>
</tr>
<tr>
<td>z/OS DFSMS OAM Application Programmer’s Reference</td>
<td>SC35-0425</td>
</tr>
<tr>
<td>z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support</td>
<td>SC35-0426</td>
</tr>
<tr>
<td>z/OS DFSMSrmm Managing and Using Removable Media</td>
<td>SC26-7404</td>
</tr>
<tr>
<td>z/OS DFSMSrmm Reporting</td>
<td>SC26-7406</td>
</tr>
<tr>
<td>z/OS DFSMSrmm Implementation and Customization Guide</td>
<td>SC26-7405</td>
</tr>
<tr>
<td>z/OS DFSMSrmm Diagnosis Guide</td>
<td>GY27-7619</td>
</tr>
<tr>
<td>z/OS DFSMSdfp Storage Administration</td>
<td>SC26-7402</td>
</tr>
<tr>
<td>z/OS DFSMS Using Data Sets</td>
<td>SC26-7410</td>
</tr>
<tr>
<td>z/OS DFSMS Access Method Services for Catalogs</td>
<td>SC26-7394</td>
</tr>
<tr>
<td>z/OS DFSMS Managing Catalogs</td>
<td>SC26-7409</td>
</tr>
<tr>
<td>z/OS Security Server RACF Security Administrator’s Guide</td>
<td>SA22-7683</td>
</tr>
</tbody>
</table>
### Accessing z/OS DFSMS information on the Internet

In addition to making softcopy information available on CD-ROM, IBM provides access to z/OS softcopy information on the Internet. To view, search, and print z/OS information, go to the z/OS Internet Library:

http://www.ibm.com/systems/z/os/zos/bkserv/

### The z/OS Basic Skills Information Center

The z/OS Basic Skills Information Center is a Web-based information resource intended to help users learn the basic concepts of z/OS, the operating system that runs most of the IBM mainframe computers in use today. The Information Center is designed to introduce a new generation of I/T professionals to basic concepts and help them prepare for a career as a z/OS professional, such as a z/OS systems programmer.

Specifically, the z/OS Basic Skills Information Center is intended to:

- Provide basic education and information about z/OS without charge
- Shorten the time it takes for people to become productive on the mainframe
- Make it easier for new people to learn z/OS.

To access the z/OS Basic Skills Information Center, open your Web browser to the following Web site, which is available to all users (no login required):

http://publib.boulder.ibm.com/infocenter/zos/basics/index.jsp

---

<table>
<thead>
<tr>
<th>Title</th>
<th>Order Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>z/OS DFSMS Installation Exits</td>
<td>SC26-7396</td>
</tr>
<tr>
<td>z/OS DFSMS Using the Interactive Storage Facility</td>
<td>SC26-7411</td>
</tr>
<tr>
<td>z/OS DFSMS Implementing System-Managed Storage</td>
<td>SC26-7407</td>
</tr>
<tr>
<td>z/OS MVS JCL User's Guide</td>
<td>SA22-7598</td>
</tr>
<tr>
<td>z/OS UNIX System Services User's Guide</td>
<td>SA22-7801</td>
</tr>
<tr>
<td>z/OS MVS Initialization and Tuning Reference</td>
<td>SA22-7592</td>
</tr>
<tr>
<td>z/OS DFSMS Using Magnetic Tapes</td>
<td>SC26-7412</td>
</tr>
<tr>
<td>z/OS MVS Data Areas, Vol 2 (DCCB-ITZYRETC)</td>
<td>GA22-7582</td>
</tr>
<tr>
<td>z/OS Planning for Multilevel Security and the Common Criteria</td>
<td>GA22-7509</td>
</tr>
</tbody>
</table>
Summary of Changes

This document contains terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations are indicated by a vertical line to the left of the change.

You might notice changes in the style and structure of some content in this document—for example, headings that use uppercase for the first letter of initial words only, and procedures that have a different look and format. The changes are ongoing improvements to the consistency and retrievability of information in DFSMS documents.

Summary of Changes for SC26-7473-06 z/OS Version 1 Release 11

This document contains information that was previously presented in DFSMS: Using the New Functions, SC26-7473-05.

The following sections summarize the changes to that information.

New Information

This edition includes the following new chapters for z/OS V1R11:

- Chapter 1, “Using extended address volumes enhancements,” on page 3
- Chapter 2, “Using the access method services DELETE MASK option,” on page 7
- Chapter 3, “Using the SMS enhancements,” on page 9
- Chapter 4, “Using the catalog enhancements,” on page 11
- Chapter 5, “Using the Open/Close/EOV enhancements,” on page 13
- Chapter 6, “Using the object access method enhancements,” on page 15
- Chapter 7, “Using the DFSMShsm enhancements,” on page 23
- Chapter 8, “Using the DFSMSrmm enhancements,” on page 29

Changed Information

The following information changed in this edition:

- Text on lock structures was updated in Chapter 10, “Using VSAM RLS secondary lock structures,” on page 49.

Moved Information

Information about the z/OS V1R6 enhancements has been moved to other books:

- Information about the SMS Parallel Access Volume option has been moved to z/OS DFSMSdfp Storage Administration.
- Information about using security labels in ACS routines has been moved to z/OS DFSMSdfp Storage Administration.
- Information about the restartable PDSE address space, SMSPDSE1, has been moved to z/OS DFSMS Using Data Sets and z/OS DFSMSdfp Diagnosis.
Summary of Changes for SC26-7473-05 z/OS Version 1 Release 10

This document contains information that was previously presented in DFSMS:
Using the New Functions, SC26-7473-04.

The following sections summarize the changes to that information.

New Information
This edition includes the following new enhancements:
- Extended address volumes
- VSAM RLS secondary lock structures
- Dynamic display of PDSE caching statistics
- SMS enhancements
- OAM enhancements
- DFSMShsm and DFSMSdss enhancements
- DFSMSrmm enhancements.

Changed Information
The following information changed in this edition:
- Steps for implementing extended TTOCs were updated in "Specifying the use of extended TTOCs" on page 185.

Summary of Changes for SC26-7473-04 z/OS Version 1 Release 9

This document contains information that was previously presented in DFSMS:
Using the New Functions, SC26-7473-03.

The following sections summarize the changes to that information.

New Information
This edition includes the following new enhancements:
- BSAM and QSAM enhancements
- VSAM system managed buffering enhancements
- SMS volume selection enhancement to support data set fast replication
- OAM enhancements
- DFSMSrmm enhancements
- DFSMShsm enhancements.

Deleted Information
The following information was deleted:
- z/OS V1R5 enhancements have been removed from this book.

Summary of Changes for SC26-7473-03 z/OS Version 1 Release 8

This document contains information that was previously presented in DFSMS:
Using the New Functions, SC26-7473-02.

The following sections summarize the changes to that information.
New Information

This edition includes the following new enhancements:
- DFSMSHsm fast replication, tape support
- DFSMSHsm fast replication, data set recovery
- PDSE 64-bit buffering, and caching data beyond closure of a PDSE
- Catalog enhancements
- New diagnostic tools
- A command to communicate with the device manager address space
- Creation of an ACDS from an SCDS
- Enhanced volume selection performance
- OAM enhancements
- SMS enhancements
- DFSMSrmm enhancements
- Tape data set authorization
- DFSMSHsm enhancements.

Changed Information

The following information changed in this edition:
- In z/OS V1R8, the default value for the BLOCKTOKENSIZE parameter in the IGDSMSxx member of SYS1.PARMLIB has changed from REQUIRE to NOREQUIRE, allowing more applications to use large format data sets without coding BLOCKTOKENSIZE=LARGE on the DCBE macro. For more information, see Chapter 34, “Using large format data sets,” on page 153.
- Restrictions on the name-hiding function have been added.

Summary of Changes for SC26-7473-02 z/OS Version 1 Release 7

This document contains information that was previously presented in DFSMS: Using DFSMSdfp in the z/OS V1R6 Environment, SC26-7473-01.

The following sections summarize the changes to that information.

New Information

This edition includes the following new enhancements:
- Large format data sets
- Device support address space
- DFSMS subchannel set support
- REPRO MERGECAT FromKey/ToKey function
- Catalog enhancements
- VSAM extent constraint removal
- VSAM RLS 64-bit virtual storage for data buffers
- SMS volume and ACS allocation test enhancements
- OAM enhancements
- DFSMSrmm enterprise enablement
- DFSMSHsm enhancements

Changed Information

The following information changed in this edition:
- Starting with the z/OS V1R7 functions, this document includes new functions for DFSMSrmm, DFSMSHsm, and DFSMSdss as well as DFSMSdfp.
- Steps for using the data set name-hiding function have been updated.
- Added a considerations table to information on re-indexing of online volumes.
• JOBCAT and STEPCAT DD statement enablement has been deleted from z/OS V1R7. The information about this function has been updated with notes indicating that it only applies to z/OS V1R5 and V1R6.
Introduction

If you are the person responsible for implementing the enhancements for z/OS DFSMS V1R11, V1R10, V1R9, V1R8, or V1R7 at your site, this document is written for you. It uses a "cookbook" approach to using these enhancements. First, you will find out what is included in each enhancement — the "ingredients." Then, you will find out how to implement these enhancements — the step-by-step procedures or "directions" for creating the "recipe."

This chapter includes information about the z/OS DFSMS enhancements and how they are presented in this document. It also includes a description of the tasks you need to perform to take advantage of these enhancements.

This document tells how to use the new DFSMS enhancements. Read the DFSMS chapter in the z/OS Introduction and Release Guide to learn about all of the new enhancements for z/OS V1R7, V1R8, V1R9, and V1R10 DFSMS. Read z/OS Migration to find out how to migrate to the new DFSMS releases. Only a subset of the enhancements have required migration actions.

What do I need to do to use the z/OS DFSMS enhancements?

Before you can implement the z/OS DFSMS enhancements at your site, you need to understand the enhancement and how it will fit with your current configuration. Then you might need to perform some tasks in order to start using the enhancement.

This book is divided into parts, each one for a separate release of z/OS:

- Part 1, "Using New DFSMS Functions in z/OS V1R11," on page 1 describes enhancements in z/OS V1R11, for all of the DFSMS elements and features including DFSMSdfp, DFSMSdss™, DFSMShsm™, and DFSMSrmm™.
- Part 2, "Using New DFSMS Functions in z/OS V1R10," on page 33 describes enhancements in z/OS V1R10, for all of the DFSMS elements and features.
- Part 3, "Using New DFSMS Functions in z/OS V1R9," on page 81 describes enhancements in z/OS V1R9, for all of the DFSMS elements and features.
- Part 4, "Using New DFSMS Functions in z/OS V1R8," on page 101 describes enhancements in z/OS V1R8, for all of the DFSMS elements and features.
- Part 5, "Using New DFSMS Functions in z/OS V1R7," on page 151 describes enhancements in z/OS V1R7, for all of the DFSMS elements and features.

Within the parts, each chapter gives an overview about one enhancement, and information about how the enhancement works. There is also a roadmap of the tasks associated with the enhancement. Each chapter includes the tasks that you must perform to use this enhancement and the associated procedures. Not every enhancement requires that you perform every type of task for full use of the enhancement at your site.

These tasks are organized based on the usual tasks performed by DFSMSdfp users like you. The types of tasks include the following:

**Evaluating**

Judging the applicability of an IBM® program for your installation and deciding whether or not to install the program at your site. It includes
deciding which program options are useful for your site, what data processing resources are needed to support the program, and what skills need to be taught to users.

Planning
Making fundamental decisions about the options a program offers. These decisions guide, set limits for, and identify requirements for installation, customization, operation, administration, application programming, and diagnosis tasks. Planning is an ongoing task; decisions are made before installation, evaluated after installation, and revised as appropriate.

Installing
Making a program ready to do useful work. This task includes generating a program, initializing a program, and applying program temporary fixes (PTFs) to a program.

Administering
Managing the data processing resources used with an IBM program to meet the planned processing goals of an enterprise. Resources include such items as processor cycles, real and virtual storage, networks, nodes, communication paths, programs, data terminals, and queues.

Operating
Starting and stopping programs, checking and controlling programs, recording the status of programs and data, and reacting to abnormal events.

Customizing
Tailoring a program to suit the needs of your installation. Enhancing or extending an IBM program by using services and built-in facilities provided by IBM.

Application programming
Designing, coding, compiling, executing, debugging, and testing application programs. Application programs put your computing system to work to meet the specific needs of your business. All other programming supports the tasks of installing, administering, operating, customizing, or diagnosing.

Diagnosing
Identifying the IBM program that is the source of a programming problem. Describing the problem, comparing it to similar known problems, reporting a new problem, and correcting the problem. It does not include diagnosing hardware problems or user errors (debugging).

Each chapter that follows provides you with the information that you need to understand the new enhancement and begin analyzing how you can use it most effectively on your system. If you need to brush up on your basic knowledge about a specific DFSMSdfp function, you can read the related background information suggested in the chapter. After you have a good understanding of the enhancement, look at the specific tasks you need to complete in order to use this enhancement effectively.

How can z/OS V1R11 DFSMS help me?
The z/OS V1R11 DFSMS enhancements are intended to help you use DFSMS more effectively and efficiently. Each of the following chapters describes one of the z/OS V1R11 DFSMS enhancements:
• **Extended Address Volumes:** removes restrictions on the types of data sets supported for EAVs. See [Chapter 1, “Using extended address volumes enhancements,”](#) on page 3 for more information.

• **Access Method Services enhancement:** adds a new MASK option to the DELETE command, providing a way to specify many variations of a data set name on a single deletion. See [Chapter 2, “Using the access method services DELETE MASK option,”](#) on page 7 for more information.

• **SMS enhancements:** provide data set separation by volume, to prevent DASD hot spots and reduce I/O contention for critical data sets, and remove restrictions that apply to SMS striping volume selection. See [Chapter 3, “Using the SMS enhancements,”](#) on page 9 for more information.

• **Catalog enhancement:** provides a new health check to identify catalogs that have the obsolete attributes of IMBED or REPLICATE defined. Catalogs that have the attributes defined should be redefined without them. See [Chapter 4, “Using the catalog enhancements,”](#) on page 11 for more information.

• **OPEN/CLOSE/EOV enhancements:** provide improved mapping for SMF type 14 and 15 records, an option to allow abending of jobs with certain multi-volume tape conditions, and other enhancements to reliability, availability, and serviceability. See [Chapter 5, “Using the Open/Close/EOV enhancements,”](#) on page 13 for more information.

• **OAM enhancements:** increase the maximum size for objects that OAM can store on tape media, add functions to prevent inadvertent or premature object deletion, enhanced wildcard support for OSREQ QUERY requests, and other enhancements. See [Chapter 6, “Using the object access method enhancements,”](#) on page 15 for more information.

• **DFSMShsm enhancements:** provide improvements to fast replication (recovery) of data sets, a way to specify a retention period on data set backup requests, and the ability to use ML1 volumes for larger data sets and for migration processing. See [Chapter 7, “Using the DFSMShsm enhancements,”](#) on page 23 for more information.

• **DFSMSrmm enhancements:** provide a number of improvements to DFSMSrmm, including improvements to the DFSMS report generator, the DFMSrmm CIM agent, and ease of use enhancements. See [Chapter 8, “Using the DFSMSrmm enhancements,”](#) on page 29 for more information.

---

**How can z/OS V1R10 DFSMS help me?**

The z/OS V1R10 DFSMS enhancements are intended to help you use DFSMS more effectively and efficiently. Each of the following chapters describes one of the z/OS V1R10 DFSMS enhancements:

• **Extended Address Volumes:** Provide a large extension to the previous size limit of 65,520 cylinders per volume. See [Chapter 9, “Using extended address volumes,”](#) on page 35 for more information.

• **VSAM RLS secondary lock structures:** Allow you to define multiple, secondary lock structures for VSAM RLS workloads, to avoid previous locking constraints. See [Chapter 10, “Using VSAM RLS secondary lock structures,”](#) on page 49 for more information.

• **PDSE caching enhancements:** Allow you to dynamically display PDSE caching statistics, using the DISPLAY SMS,PDSE and DISPLAY SMS,PDSE1 commands with the new HPSTATS parameter. See [Chapter 11, “Using the PDSE enhancements,”](#) on page 55 for more information.
• **SMS data class override and other enhancements**: Provide the ability for space attributes set in the data class to override corresponding attributes set in JCL; allow enforcement of system-determined block size when a user-determined block size exists; provide new read-only variables for ACS routines; and write certain key message to the job log and hardcopy log. See Chapter 12, “Using the SMS enhancements,” on page 59 for more information.

• **OAM enhancements**: Extend the maximum object size that can be accepted and managed by OAM from 256 megabytes to practically 2 gigabytes for the DASD level of the OAM hierarchy. Also provide a number of other key OAM improvements. See Chapter 13, “Using the object access method enhancements,” on page 63 for more information.

• **DFSMShsm and DFSMSdss enhancements**: Expand data set backup processing, allowing you to specify a new data set name to assign to the newly created backup version, as well as other enhancements including improvements to CDS backup processing. DFSMShsm also supports the DFSMSrmm report generator as an alternative method of gathering reports about DFSMShsm activity from SMF records. See Chapter 14, “Using the DFSMShsm and DFSMSdss enhancements,” on page 69 for more information.

• **DFSMSrmm enhancements**: Provide a number of improvements to DFSMSrmm enterprise enablement and ease of use, to the DFSMSrmm report generator, and to optimization and management capabilities. See Chapter 15, “Using the DFSMSrmm enhancements,” on page 75 for more information.

**How can z/OS V1R9 DFSMS help me?**

The z/OS V1R9 DFSMS enhancements are intended to help you use DFSMS more effectively and efficiently. Each of the following chapters describes one of the z/OS V1R9 DFSMS enhancements:

• **BSAM and QSAM enhancements**: Allow you to specify long-term page fixing for BSAM data buffers, and benefit from QSAM support for the DCBE macro's MULTSDN parameter and BSAM and QSAM support for the MULTACC parameter. See Chapter 16, “Using BSAM and QSAM Enhancements,” on page 83 for more information.

• **VSAM system-managed buffering enhancements**: Allow you to modify the SMBVSP value more readily for large numbers of data sets, by specifying it for a data class using ISMF, instead of in individual JCL statements. See Chapter 17, “Using VSAM System-Managed Buffering enhancements,” on page 87 for more information.

• **SMS volume selection enhancements**: Support data set fast replication by giving preference to volumes that are in the same SFI, when allocating or extending an SMS-managed multi-volume data set. See Chapter 18, “Using SMS volume selection preference for the same storage facility image,” on page 89 for more information.

• **OAM enhancements**: Provide two new tape sublevels, giving the ability to transition object data within the tape family, for example, from VTS to native tape. See Chapter 19, “Using the Object Access Method enhancements,” on page 91 for more information.

• **DFSMSrmm enhancements**: Extend the CIM subclasses to differentiate the resources/objects that DFSMSrmm manages, and the CIM Provider to support the out-of-process mode functionality added to OpenPegasus CIM Server with 2.5.1. In addition, a large number of enhancements to DFSMSrmm's performance, RAS, operability, and usability are included. See Chapter 20, “Using the DFSMSrmm enhancements,” on page 95 for more information.
• **DFSMShsm enhancements**: Provide new information in the FSR to improve analysis of DFSMShsm processing and performance. See Chapter 21, “Using the DFSMShsm enhancements,” on page 99 for more information.

**How can z/OS V1R8 DFSMS help me?**

The z/OS V1R8 DFSMS enhancements are intended to help you use DFSMS more effectively and efficiently. Each of the following chapters describes one of the z/OS V1R8 DFSMS enhancements:

• **Fast replication enhancements**: Allow you to dump fast replication backup copies to tape (through operator commands or automatic dump processing), and allow you to restore individual data sets from copy pool backup copies. See Chapter 22, “Using DFSMShsm fast replication, tape support,” on page 103 and Chapter 23, “Using DFSMShsm fast replication, data set recovery,” on page 109 for more information.

• **PDSE enhancements**: Allow you to specify an amount of 64-bit virtual storage to be used for caching PDSE directory buffers, and to specify that PDSE data be cached in memory after PDSEs are closed. These enhancements can increase the number of PDSE members that can be open concurrently, and improve performance for data sets that repeatedly opened for update. See Chapter 24, “Using PDSE enhancements,” on page 117 for more information.

• **Catalog enhancements**: Allow you to create page spaces that are larger than four gigabytes in size, and allow you to specify a maximum number of catalog service requests of up to 999 (the previous limit was 180). The higher limit for catalog service requests can improve system performance, making more service tasks available to handle requests without waiting, especially with faster and more numerous processors. In addition, the IDCAMS LISTCAT command is enhanced to display catalog entries sorted in order within the catalogs they belong to. See Chapter 25, “Using catalog enhancements,” on page 121 for more information.

• **New diagnostic tools** New dumps, trace events, and a new DISPLAY command provide improved diagnostic capabilities. See Chapter 26, “Using the new diagnostic tools,” on page 125 for more information.

• **MODIFY DEVMAN command**: Allows you to display information about, or to request a specified service from, the device manager address space (DEVMAN). See Chapter 27, “Communicating with the device manager address space (MODIFY DEVMAN command),” on page 127 for more information.

• **Creating an ACDS directly from a SCDS** Allows you to use an operator command to create an ACDS from any valid SCDS. See Chapter 28, “Creating an ACDS directly from a SCDS,” on page 129 for more information.


• **OAM enhancements**: Allows you to store objects larger than 32 KB using DB2®’s large object (LOB) support and the binary large object (BLOB) data type, allows automatic selection of tape volumes for MOVEVOL with RECYCLE processing, and adds support for an immediate backup copy of an object. See Chapter 30, “Using the Object Access Method enhancements,” on page 133 for more information.

• **DFSMShsm enhancements**: Provide improvements in the areas of enterprise level interface, UTC implementation, tape data set authorization, vital record specification policy management simplification, and usability items. See Chapter 31, “Using the DFSMShsm enhancements,” on page 137 for more information.
How can z/OS V1R7 DFSMS help me?

The z/OS V1R7 DFSMS enhancements are intended to help you use DFSMS more effectively and efficiently at your site. Each of the following chapters describes one of the z/OS V1R7 DFSMS enhancements:

- **Large Format Data Sets enhancement**: Allows you to create and use physical sequential data sets that can grow beyond the previous size limit of 65,535 tracks per volume. See Chapter 34, “Using large format data sets,” on page 153 for more information.

- **Device Support Address Space enhancement**: Provides a new address space for gathering first data capture trace information for improved serviceability. See Chapter 35, “Using the device support address space,” on page 159 for more information.

- **Additional Subchannel Sets enhancement**: Allows SMS parallel access volume (PAV) users to create and use an additional subchannel to increase the number of available devices. See Chapter 36, “Using the DFSMS subchannel set support,” on page 161 for more information.

- **REPRO MERGECAT FromKey/ToKey Enhancement**: Allows you to repair damaged catalogs by targeting a range of catalog keys in a REPRO MERGECAT command. See Chapter 37, “Using the REPRO MERGECAT FromKey/ToKey Enhancement,” on page 163 for more information.

- **Catalog enhancements**: Allow you to specify the amount of space to be used for implicit allocation of a VSAM volume data set (VVDS), and automatically tune default parameter values for buffers and strings when catalogs are created. See Chapter 38, “Using the catalog enhancements,” on page 165 for more information.

- **VSAM extent constraint removal enhancement**: For non SMS-managed volumes, VSAM data sets can expand up to 255 extents per component and striped VSAM data sets can expand up to 255 extents per stripe. For SMS-managed volumes, these extent limits are removed, if the extent constraint removal parameter in the data class is set to Y (yes). See Chapter 39, “Using VSAM extent constraint removal,” on page 167 for more information.

- **VSAM RLS 64-bit Data Buffers enhancement**: Allows VSAM RLS processing to use 64-bit addressable virtual storage for data buffers, to potentially improve performance for your high-transaction applications. See Chapter 40, “Using VSAM RLS 64-bit data buffers,” on page 169 for more information.

- **SMS Volume and ACS Allocation Test enhancements**: These enhancements, described in Chapter 41, “Using the SMS volume and ACS allocation test enhancements,” on page 173, allow you to:
  - Use the V SMS,VOLUME command to alter the status of the volume from NOTCON to another status
  - Request summarized and detailed volume selection messages including DADSM failure reason codes
- Receive volume selection analysis selection data in SMS trace including additional trace data for SMS and non-SMS VSAM allocations
- Save temporary changes made to the active configuration data set (ACDS)

• **Object Access Method Enhancements**: Provide a number of improvements in tape and optical object availability, including immediate recall to DB2 DASD. See Chapter 42, “Using the Object Access Method Enhancements,” on page 175 for more information.

• **DFSMSrmm enterprise enablement enhancements**: Allows you to use the high-level language application programming interface as a web service. This enables the high-level language application programming interface to be used from any system or platform that can run Java™, C++, or any language that supports the web services standards. In addition, a plug-in adapter uses the CIM provider interface to provide real-time information about storage resources. See Chapter 43, “Using DFSMSrmm enterprise enablement,” on page 181 for more information.

• **DFSMShsm enhancements**: Provide improvements in the areas of constraint relief, usability, serviceability, and tape management. See Chapter 44, “Using the DFSMShsm enhancements,” on page 183 for more information.
# Part 1. Using New DFSMS Functions in z/OS V1R11

## Chapter 1. Using extended address volumes enhancements
- Planning and installation
- Planning to use extended address volumes enhancements
- Setting up support for extended address volumes enhancements
- Administering
- DFSMSdfp administration
- Application programming
- Modifying applications to use extended address volumes enhancements
- Diagnosing problems with extended address volumes enhancements

## Chapter 2. Using the access method services
### DELETE MASK option

## Chapter 3. Using the SMS enhancements
- Activating data set separation by volume
- Using the striping volume selection enhancements

## Chapter 4. Using the catalog enhancements
- Administering
- Using the catalog check
- Redefining catalogs to remove IMBED and REPPLICATE attributes

## Chapter 5. Using the Open/Close/EOV enhancements
- Planning and implementing the OCE enhancements

## Chapter 6. Using the object access method enhancements
- System Programming
- Specifying object sizes up to 2000 MB for tape
- Specifying deletion-protection
- Administering
- Enabling larger object sizes for tape
- Defining deletion- and retention-protection for object storage groups
- Using the CHGCOL utility to modify OAM collection defaults
- NEW return codes to be processed by the CBRUXSAE security authorization exit
- Application Programming
- Specifying storage for large objects in parts
- Specifying deletion-hold status and extending expiration dates for objects
- Using expanded wildcards in OSREQ QUERY requests
- Operating
- Displaying deletion- and retention-protection status
- Displaying IEFSSNxx parameter settings for OAM
- Displaying OAM parameter settings
- Specifying a retention period on DFSMSrmm commands

## Chapter 7. Using the DFSMSHsm enhancements
- Planning and Installation
- Planning to use the DFSMS enhancements
- Administering
- Specifying a retention period for a data set backup copy
- Providing access to the RETAINDAYS keyword
- Capturing catalog information for data set recovery
- Specifying the use of Metro Mirror volumes for backup and recovery using the copy pool definition
- Enabling ML1 overflow volume selection for migration or backup of large data sets
- Operating
- Specifying a retention period on DFSMSHsm commands

## Chapter 8. Using the DFSMSrmm enhancements
- Planning and Installation
- Planning for migration
- Planning to use DFSMSrmm enhancements
- Exploiting report generator enhancements
- Exploiting DFSMSrmm operational enhancements
- Exploiting DFSMSrmm API enhancements
Chapter 1. Using extended address volumes enhancements

z/OS V1R10 introduced extended address volume (EAV), which allowed DASD storage volumes to be larger than 65,520 cylinders. The space above the first 65,520 cylinders is referred to as cylinder-managed space. Tracks in cylinder-managed space use extended addressing space (EAS) techniques to access these tracks. Data sets that are able to use cylinder-managed space are referred to as being EAS-eligible. The following table shows enhancements to the EAV function for z/OS R11:

<table>
<thead>
<tr>
<th>z/OS V1R10 EAV</th>
<th>z/OS V1R11 EAV enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only VSAM data sets are EAS-eligible.</td>
<td>Extended format sequential data sets are now EAS-eligible.</td>
</tr>
<tr>
<td>Control whether VSAM data sets can reside in cylinder-managed space by including or excluding EAVs in particular storage groups. For non-SMS managed data sets, control the allocation to a volume by specifying a specific VOLSER or esoteric name.</td>
<td>Control whether the allocation of EAS-eligible data sets can reside in cylinder-managed space using both the methods supported in z/OS V1R10 and by using the new EATTR data set attribute keyword.</td>
</tr>
</tbody>
</table>

For information about planning, administering, modifying applications and diagnosing problems for EAV and EAS in general, see Chapter 9, “Using extended address volumes,” on page 35. For z/OS V1R11 enhancement details, read this section.

The following table lists the types of tasks and associated procedures that you must complete to use extended address volumes.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
</table>
| “Planning and installation” | • “Planning to use extended address volumes enhancements”  
| | • “Setting up support for extended address volumes enhancements” on page 4 |
| “Administering” on page 5 | |
| “Application programming” on page 5 | “Modifying applications to use extended address volumes enhancements” on page 5 |
| “Diagnosing problems with extended address volumes” on page 6 | |

Planning and installation

This section describes the planning and installation tasks that you must perform to use extended address volumes.

Planning to use extended address volumes enhancements

Before you begin: Ensure that you understand and have set up extended address volumes, see Chapter 9, “Using extended address volumes,” on page 35.
Software requirements: Using the enhancements to extended address volumes requires z/OS V1R11. EAV requires z/OS V1R10 or higher.

Hardware requirements: DS8000 Licensed Internal Code 4.0 or higher is required to support EAV.

Coexistence requirements:
- Systems prior to z/OS V1R11 will not be able to use EAV enhancements.
- Release z/OS V1R10 provides toleration of EAV enhancements.
- Releases prior to z/OS V1R10 also provide toleration of EAVs.

Supported data sets: EAS-eligible data sets are those that can be allocated anywhere on an extended address volume. z/OS V1R11 supports the following types of data sets as EAS-eligible data sets:
- SMS managed VSAM (all types)
- Non-SMS VSAM (all types)
- VSAM data sets inherited from prior physical migrations or copies
- VSAM temporary data sets
- zFS data sets (they are VSAM).
- Extended format sequential data sets

Non-EAS eligible data sets are those that can be allocated only in the track-managed space of an EAV volume. For z/OS V1R11, non-EAS eligible data sets include:
- Page data sets
- Catalog data sets (BCS and VVDS)
- VSAM data sets with imbed or keyrange attributes that may have been inherited from prior physical migrations or copies
- VSAM data sets with incompatible control area (CA) sizes. (Compatible CA sizes are 1, 3, 5, 7, 9 and 15 tracks.)
- Non-VSAM data sets, with the exception of extended format sequential data sets.

Planning steps:
- To help determine which applications might need to be modified to support EAVs, see "Application programming" on page 39 and information about the EAV migration assistance tracker in z/OS DFSMSdfp Advanced Services.

Setting up support for extended address volumes enhancements

Related reading: For information on initial set up for extended address volumes, see "Setting up support for extended address volumes" on page 37. This section covers only set up for extended address volumes enhancements.

On the ALLOCATE or DEFINE CLUSTER commands, allocate new data sets with parameter EATTR(OPT) to specify that the data set can have extended attribute DSCBs (format 8 and 9) and can optionally reside in EAS. This is the default behavior for VSAM data sets if EATTR(OPT) is not specified. For non-VSAM data sets the default is that the data set can not have extended attribute DSCBs and
optionally reside in EAS. This is equivalent to specifying EATTR(NO) on the
command. See ALLOCATE and DEFINE CLUSTER in z/OS DFSMS Access Method
Services for Catalogs.

Use the EATTR JCL keyword, EATTR=OPT, to specify that the data set can have
extended attribute DSCBs and can optionally reside in EAS. This is the default
behavior for VSAM data sets if EATTR(OPT) is not specified. For non-VSAM data
sets the default is that the data set can not have extended attribute DSCBs and
optionally reside in EAS. This is equivalent to specifying EATTR=NO on the JCL
and is applicable to all data set types. See EATTR Parameter in z/OS MVS JCL
Reference.

Use the EATTR attribute on the ISMF Data Class Define panel, to define a data set
that can have extended attribute DSCBs and optionally reside in EAS. See Defining
Volume and Data Set Attributes for Data Class in z/OS DFSMSdfp Storage
Administration.

Administering

This section describes administration tasks for extended address volumes.

**DFSMSdfp administration**

Administration tasks for DFSMSdfp include:

- Specifying EATTR in the CSIFLDNM field name list for the Catalog Search
  Interface to request information about whether a data set supports extended
  attributes (format 8 and 9) and can reside in the EAS of an EAV. See Field Name
  Directory in z/OS DFSMS Managing Catalogs.

- Specifying REVFTOC on the MODIFY DEVMAN command to use ICKDSF to
  automatically REFORMAT/REFVTOC a volume when it expands. See z/OS MVS
  System Commands.

- Specifying REVFTOC=ENABLE | DISABLE in parmlib member DEVSUPxx to
  specify whether you want to enable or disable use of ICKDSF to automatically
  REFORMAT/REFVTOC a volume when it expands. See z/OS MVS Initialization
  and Tuning Reference.

**Application programming**

### Modifying applications to use extended address volumes enhancements

The following types of applications might need modifications for extended address
volumes.

1. Applications using interfaces that deal with DASD volume capacity:
   - Output of IEHLIST LISTVTOC – the output will now display the job name,
     step name, and create time for a data set with a format 8 or format 9 DSCB.
   - Output of IDCAMS DCOLLECT – the active data set information (record
type “D”) and data class construct information (record type ‘DC’) have new
   and changed fields. See Interpreting DCOLLECT Output in z/OS DFSMS
   Access Method Services for Catalogs.
   - The parameter list built by the DSECT form of the REALLOC macro includes
     information about the extended attribute (EATTR) for a data set, indicating
whether it supports extended attributes (format 8 and 9) and can reside in the EAS of an EAV. See REALLOC Parameter List in z/OS DFSMSdfp Advanced Services.

**Diagnosing problems with extended address volumes**

z/OS R11 includes new and changed messages for extended address volumes and changes to reason code 01 for wait state code 00E.

**Related reading:** For information about wait state code 00E, see z/OS MVS System Codes.
Chapter 2. Using the access method services DELETE MASK option

In z/OS V1R11, DFSMS access method services (IDCAMS) adds a new MASK option to the DELETE command. This option lets you specify many variations of a data set name on a single deletion, using new wild card characters and rules to give more flexibility in selecting the data sets to be deleted.

Previously, only one data set qualifier could be replaced by a wild card on a DELETE command. For example: DELETE A.*.C would delete data sets with A as the first qualifier, any second qualifier, and C as the third qualifier, such as A.B.C or A.BB.C. With the new MASK keyword, you can replace multiple qualifiers with wild cards, and can replace specific characters in a qualifier name with wild cards as well.

The MASK keyword treats a single asterisk in the same way as the generic DELETE command. For example DELETE A.*.C MASK specifies a data set with A and C as the first and third qualifiers, and a second qualifier with any set of characters. In addition, the MASK keyword allows two consecutive asterisks to replace multiple qualifier names. For example, the entry name DELETE A.**.B MASK means all data set names of two or more levels with A as the first qualifier, and B as the last qualifier. A.B.** means all data set names of two or more levels where A is the first qualifier and B is the second (or last) qualifier. Double asterisks cannot be part of a qualifier name; they must be preceded or followed by either a period or a blank character.

The MASK keyword also accepts percent signs (%) as wild cards to replace individual characters in the same position. You can specify from one to eight percent signs in each qualifier. For example, DELETE A%DE MASK would match a data set named ABDE, while DELETE A%09/04/29 MASK would match one named ABCDE.

The DELETE MASK command allows only one data set entry-name to be specified. If multiple entry-names are specified, the DELETE request will fail with error messages. If more than 100 data set names are filtered from the wild card notation, AMS only deletes the first 100 data sets identified by the filtering process.

If you do not specify the MASK keyword on the DELETE command, or explicitly specify NOMASK, the previous wild card rules remain in effect (one asterisk can replace one qualifier in a data set name).

For more information about using the new MASK keyword on the IDCAMS DELETE command, see z/OS DFSMS Access Method Services for Catalogs.
Chapter 3. Using the SMS enhancements

In z/OS V1R11, SMS introduces the following enhancements:

1. **Data set separation by volume**: starting in z/OS V1R11, you can specify that critical data sets be allocated onto different volumes, to prevent DASD hot spots and reduce I/O contention for them. The existing data set separation function is extended from the PCU level to the volume level, to separate critical SMS-managed data sets onto different extent pools and volumes from other data sets in the separation group.

2. **Striping enhancement**: This enhancement removes a number of restrictions to make SMS striping volume selection as close as possible to conventional volume selection.

### Activating data set separation by volume

To separate data sets by volume, specify the groups of data sets that need to be kept separate from each other, and whether the separation is mandatory or preferred. This specification is done in the separation profile. The specified separation groups include a list of data set names to be separated from each other during allocation.

To specify data set separation by volume, use the new `VOLUME` value on the `SEPARATIONGROUP` keyword of the data set separation profile. You can specify the separation level to be by PCU as previously, or by volume. A new `TYPE` keyword specifies whether the separation is required or preferred. For example, the following syntax specifies that separation is required to be by volume:

```
SEPARATIONGROUP(VOLUME)
TYPE(REQUIRED)
DSNLIST(dsn,...)
```

When you specify that separation by volume is required, SMS allocates the data set onto the volumes that are not used by any of the data sets in the same separation group. When separation by volume is preferred, the candidate volumes that are already occupied by other data sets in the separation group will be ranked lower and less preferred for selection.

For both `VOLUME` and PCU, the data set names can contain the wild card characters *, **, and %, starting at the third level qualifier; for example:

```
SEPARATIONGROUP(VOLUME)
TYPE(REQUIRED)
DSNLIST(A.B.*,E.F.%%,...)
```

For details on using the wild card characters, see the Catalog Search Interface User’s Guide section of [z/OS DFSMS Managing Catalogs](https://www.ibm.com/products/zos-managing-catalogs).

Toleration PTFs are available that provide support for the z/OS V1R11 syntax on the lower level systems. For more details on specifying the data set separation by volume, see [Using Data Set Separation](#) in [z/OS DFSMSdfp Storage Administration](https://www.ibm.com/products/zos-dfsmsdfp-storage-administration).
Using the striping volume selection enhancements

z/OS R11 removes a number of restrictions that applied to striping volume selection, to make that function as close as possible to conventional volume selection. The following new functions are added:

- Enabled volumes are preferred over quiesced volumes.
- Volumes that do not have sufficient space below the "high threshold" to contain the stripe are eligible for selection, and not rejected outright.
- Normal storage groups are preferred over overflow storage groups.
- The storage group sequence order as specified in the ACS storage group selection routines is supported when multi-tiered storage group is requested in the storage class.
- Data set separation is supported.
- Volumes preference attributes, such as availability, accessibility, and PAV capability are supported.
- Fast Volume Selection is supported, regardless of the current specification of the FAST_VOLSEL parameter. SMS will reject the candidate volumes that do not have sufficient free space for the stripe when 100 volumes have already been rejected by DADSM for insufficient space. This is to prevent the striping allocation from over-using the system resources, because an iteration of volume re-selection may consume a lot of overhead when there are a large number of candidate volumes.

For more details on the striping volume selection enhancements, see [Striping Volume Selection](#) in [z/OS DFSMSdfp Storage Administration](#).
Chapter 4. Using the catalog enhancements

z/OS no longer supports the IMBED and REPLICATE attributes for user catalogs and master catalogs. These attributes have become obsolete with the introduction of newer cached DASD devices, and they can have adverse effects on catalogs that were originally defined with them, including degraded performance and in some cases unplanned outages. In z/OS R11, z/OS provides a new IBM Health Checker for z/OS check that you can use to identify catalogs which have these attributes defined. It is recommended that you re-define all catalogs that currently have the IMBED or REPLICATE attributes, to remove them.

See Detecting Obsolete Catalog Attributes with IBM Health Checker for z/OS in z/OS DFSMS Managing Catalogs for complete details on using the check to identify user and master catalogs that have the IMBED or REPLICATE attributes.

Before you begin: For more information on setting up and using IBM Health Checker for z/OS, see Setting up IBM Health Checker for z/OS in IBM Health Checker for z/OS: User's Guide.

The following table lists the overall tasks to perform, to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedures you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administering</td>
<td>• Using the new check to identify catalogs with IMBED or REPLICATE attributes</td>
</tr>
<tr>
<td></td>
<td>• Redefining catalogs to remove IMBED and REPLICATE attributes.</td>
</tr>
</tbody>
</table>

Administering

You can implement the functions associated with the catalog enhancements by completing the following tasks:

• Using the check
• Removing IMBED or REPLICATE attributes

Using the catalog check

The check issues an informational message for successful completion when no catalogs are defined with IMBED or REPLICATE attributes, and an exception message for catalogs identified with IMBED or REPLICATE attributes. If an exception is found, the system issues message IGGHC104E and generates a report in the message buffer to describe the check’s findings in message IGGHC106I.

Sample report:

<table>
<thead>
<tr>
<th>CATALOG NAME</th>
<th>ATTRIBUTE</th>
<th>COMPONENT</th>
<th>AVAILABILITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>catname</td>
<td>attribute</td>
<td>component</td>
<td>availability</td>
</tr>
<tr>
<td>catname</td>
<td>attribute</td>
<td>component</td>
<td>availability</td>
</tr>
</tbody>
</table>

where

catalog name

is the name of the affected catalog
attribute
   is the obsolete attribute (IMBED or REPLICATE) with which the catalog is
defined

component
   identifies the particular component of the catalog that has the
corresponding attribute

availability
   indicates (Yes or No) whether or not the catalog is available for processing.

Related reading: For more information, see Detecting Obsolete Catalog Attributes
               with IBM Health Checker for z/OS in z/OS DFSMS Managing Catalogs.

Redefining catalogs to remove IMBED and REPLICATE attributes

If the health check reports any catalogs defined with IMBED or REPLICATE
attributes, a system programmer needs to perform an EXPORT/IMPORT operation
on the affected catalogs to remove the attributes. Use the EXPORT command to
create a back up and later to recover, and use the IMPORT command for the
exported copies. This should ideally be done during system down time, when the
catalogs cannot be accessed by any users.
Chapter 5. Using the Open/Close/EOV enhancements

Open/Close/EOV provides the following enhancements in z/OS V1R11:

- Displaying the fact that a data set has an expiration date set to never expire, in message IEC507D
- The system no longer reprocesses the same volume when duplicate volume serial numbers are detected during multi&hyphen;volume tape processing (automatic recovery for message IEC708I)
- Providing improved mapping for SMF type 14 and 15 records, with the option to provide DSECTs for the individual segments of the records.
- Providing an installation-wide abend option for multi-volume tape conditions that generate messages IEC709I, IEC710I, IEC711I, and IEC712I. New parameter bits and a new return code are added for the label anomaly exit, to allow abending of jobs for these conditions.

If your installation currently has automation to handle IEC507D messages, consider updating it to handle the new NEVEREXPIRE value that appears in the message text. If you currently have automation to handle IEC708I messages, it might need changes to accommodate that Open/Close/EOV will no longer reread duplicated volumes identified in the message.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and Installation</td>
<td>Planning and implementing the OCE enhancements</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and installation tasks that you would perform to use the OCE enhancements.

Planning and implementing the OCE enhancements

Perform the following steps to plan for using the new functions:

1. Decide whether you want to use the new installation-wide abend option for multi-volume tape conditions that generate the following messages:
   - IEC709I ddd,volser,jn,sn,ddn-nu EXPECTED VOLSEQ: nnnn FOUND: nnnn
   - IEC710I ddd,volser,jn,sn,ddn-nu ANOTHER VOLUME EXPECTED
   - IEC711I ddd,volser,jn,sn,ddn-nu RDBACK - NOT LAST VOLUME OF DATA SET I
   - IEC712I ddd,volser,jn,sn,ddn-nu READ - NOT FIRST VOLUME OF DATA SET

   **Note:** Prior to Release 11 and in Release 11 you can use a system automation product to cancel jobs that receive these messages. That process has several disadvantages:
   - The cancellation is asynchronous. The job can damage other data sets before the cancellation.
It might be preferable to cancel only the subtask and not the whole job. The label anomaly exit fails only the current task and the task can recover from it with a DCB ABEND exit or ESTAE exit.

If you decide to use the new installation-wide abend option, check for the new Label Anomaly indicator flags for these conditions then use return code C as documented in the label anomaly exit information in z/OS DFSMS Installation.

Note: there are two unrelated messages numbered IEC709I.

2. If you have automation handling messages IEC507D or IEC708I, update it as appropriate.
Chapter 6. Using the object access method enhancements

z/OS DFSMS V1R11 provides the following enhancements to the object access method (OAM):

- Increases the maximum size for objects that OAM can store on tape media. The previous maximum object size is increased to 2000 MB (2,097,152,000 bytes). This new maximum size matches the maximum size set in z/OS R10 for objects stored on DASD devices. The increase enables OAM to store objects up to 2000 MB in size on disk or tape, and create backup copies of them on tape. Note: the maximum size for objects residing on optical media remains at 256 MB.

- Adds the following new retention-related functions to OAM:
  - Deletion-hold: prevents object deletion while the object is in deletion-hold mode
  - Retention-protection: prevents object deletion prior to the object’s expiration date, and does not allow the expiration date to be changed to an earlier date.
  - Deletion-protection: prevents object deletion prior to the object’s expiration date.
  - Event-based retention: provides an object expiration date that depends on external event notification. When an object is in event-based retention mode, its expiration date is not calculated until OAM has received notification that the event has occurred.

- Additional new functions include enhanced wildcard support for OSREQ QUERY requests, a new utility to modify the storage class and management class defaults associated with an OAM collection, and more granular return codes to be processed by the CBRUXSAE security authorization user exit.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“System Programming” on page 16</td>
<td>• “Specifying object sizes up to 2000 MB for tape” on page 16</td>
</tr>
<tr>
<td></td>
<td>• “Enabling and specifying the level of deletion-protection” on page 16</td>
</tr>
<tr>
<td></td>
<td>• “New return codes to be processed by the CBRUXSAE security authorization exit” on page 18</td>
</tr>
<tr>
<td>“Administering” on page 17</td>
<td>• “Enabling larger object sizes for tape” on page 17</td>
</tr>
<tr>
<td></td>
<td>• “Defining deletion- and retention-protection for object storage groups” on page 17</td>
</tr>
<tr>
<td></td>
<td>• “Using the CHGCOL utility to modify OAM collection defaults” on page 18</td>
</tr>
</tbody>
</table>
System Programming

After installing z/OS DFSMS V1R11, and completing any migration actions, you need to complete the following system programming tasks to take advantage of the new functions.

For information about required migration actions, see z/OS Migration.

### Specifying object sizes up to 2000 MB for tape

To implement the increased OAM object size in the tape level (same as in the DASD level), a system programmer must take the following steps:

In the IEFSSNxx PARMLIB member, update the MOS= keyword on the OAM1 entry to specify the maximum object size that can be stored through the OSREQ programming interface. The MOS= keyword now allows you to specify object sizes greater than 256 MB and up to 2000 MB (2,097,152,000 bytes).

Update the LOB= keyword on the OAM1 entry in the IEFSSNxx to specify partial (P) if only some storage groups have the required DB2 configuration for objects greater than 256 MB in size, or to specify all (A) if all storage groups have the required DB2 configuration for these objects.

For details, see the updated OAM syntax and parameters for IEFSSNxx in z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

If either of the above updates are made to IEFSSNxx, it will not take effect until the next IPL.

### Enabling and specifying the level of deletion-protection

Use the new DP=x keyword on the OAM1 entry in IEFSSNxx to enable deletion-protection of data and specify the level at which it applies, either globally or at the storage-group level. Possible DP values are: A, to enable deletion-protection for all object storage groups, P for partial protection, to be specified at the object storage group level (by the OAM Deletion Protection value specified in the SMS storage group construct for a given object storage group), or
Enabling larger object sizes for tape

To enable the larger OAM object sizes in the tape level of the OAM hierarchy, do the following:

- Configure your systems to make sufficient 64-bit addressable virtual memory available to the OAM address space, by specifying an appropriate MEMLIMIT value or REGION value, in addition to specifying the maximum object size for your installation. For any storage group to support objects greater than 256M and up to 2000M in the tape level of the OAM storage hierarchy, OAM uses 64-bit addressable virtual memory to contain these objects as they are in the process of being stored or transitioned or otherwise moved by OAM management activities.
- Ensure that storage groups are properly configured so that the scratch tape media has a capacity greater than the maximum object size, because OAM objects do not span physical media.
- The MOS= external on the OAM1 entry in the IEFSSNxx SYS1.PARMLIB must specify the maximum object size value in megabytes that OAM can store for your installation. Valid values for MOS= are 50-2000. If MOS= is not specified, then the maximum object size defaults to 50M. Note: Regardless of the MOS= value specified, the maximum object size supported on optical media is 256M.
- For additional steps that are required to support the larger object sizes in the DASD level of OAM, see "DB2 administration" on page 66.

Note that the z/OS DFSMS 1.8 DB2 infrastructure is available in sample job CBRIOB. CBRIOB will create the LOB storage structures required for OAM to exploit OAM Large Object Phase 2 support. This job can be used to delete existing LOB storage structures as well as create LOB storage structures, which include:

- LOB tablespaces
- Base tables
- Base table views
- Auxiliary tables
- LOB indexes

For more information about these administration tasks, see the KB Tracking section in z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

Defining deletion- and retention-protection for object storage groups

New SMS and ISMF parameters let you specify deletion-protection and retention-protection for an object storage group. On the Object Storage Group Define/Alter panel, new fields are added: OAM Deletion Protection and OAM Retention Protection. Specify Y or N to enable or disable the protection mode for all objects in the object storage group.
OAM Deletion Protection = Y | N:
- Y (Enable) – objects in this object storage group cannot be deleted prior to their expiration date.
- N (Disable) – objects can be deleted before their expiration date (the default).

The OAM Deletion Protection parameter is ignored when DP=A or DP=N in the IEFSSNxx PARMLIB member. Deletion-protection does not restrict any changes to an object’s expiration date.

OAM Retention Protection = Y | N:
- Y (Enable) – objects in this object storage group cannot be deleted prior to their expiration date, and their expiration date can never move to an earlier date.
- N (Disable) – objects can be deleted before their expiration date (the default).

New objects stored into an object storage group with retention-protection enabled are flagged as retention-protected for the entire life of the object. Note: enabling retention-protection could result in objects that will never expire and therefore cannot be deleted.

Using the CHGCOL utility to modify OAM collection defaults
Before z/OS R11, modifying storage class and management class defaults for a given OAM collection was a five-step manual process, documented in the z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support. Now, you can use the OAM CHGCOL utility to perform these tasks for an OAM collection:
- Change the default storage class name only
- Change the default management class name only
- Change both default storage class name and default management class name.

To invoke CHGCOL, modify and run the CBRSAMUT SAMPLIB job or issue a TSO/E OAMUTIL command to start the utility. For more information, see the z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

New return codes to be processed by the CBRUXSAE security authorization exit
The CBRUXSAE installation exit provides security authorization checking against users performing OSREQ transactions on object data. This exit is used at the application programming interface (OSREQ macro) level. The sample CBRUXSAE exit in SAMPLIB, defaults to returning a return code 16 indicating “Bypassed”, meaning that the current and all future user IDs are authorized to perform all OSREQ functions and that the exit need not be called again. Installations must substitute this code with a validation routine to determine authority for a specific user ID in order for authorization checking to be performed at the application interface level.

In z/OS V1R11, OAM provides more granular return codes to be processed by the CBRUXSAE security authorization user exit. The CBRUXSAE sample includes the new return code values. The additional return codes enable an installation to code up their CBRUXSAE user exit to:
- Bypass the exit for any combination of functions. For example, the exit can be bypassed for OSREQ QUERY and RETRIEVE requests but active for OSREQ STORE, CHANGE and DELETE requests.
Application Programming

After activating the increased maximum object size as described in “Specifying object sizes up to 2000 MB for tape” on page 16, you can use new functions in the OSREQ API to facilitate the storing of objects greater than 256 MB and up to 2000 MB in size, by allowing for the object to be stored in individual parts. You can use other new OSREQ functions to specify deletion-hold status and event expiration dates for objects.

Specifying storage for large objects in parts

You can use the following new OSREQ API functions to store large objects in separate parts.
- STOREBEG – begin a store operation for an object whose total size is greater than 256 MB
- STOREPRT – store the next sequential contiguous part of an object whose total size is greater than 256 MB
- STOREEND – end the storage of an object whose total size is greater than 256 MB either to complete the storage of the object or to effectively cancel the storage of the object.

Use these new functions in a sequence, starting with the OSREQ STOREBEG to initiate the store operation and provide much of the information that would be provided on an OSREQ STORE. Then code one or more OSREQ STOREPRT function invocations to provide each individual part of the object from the beginning of the object to the end contiguously and in sequence. Finally, code an OSREQ STOREEND function invocation to complete the storage of the object after all of the individual parts have been provided to OAM.

For more information about how to code and use these new functions, see z/OS DFSMS OAM Application Programmer’s Reference.

Specifying deletion-hold status and extending expiration dates for objects

You can use the following new keywords for the OSREQ API and OSREQ TSO command for deletion-protection enhancements.
- DELHOLD(HOLD | NOHOLD), added to the OSREQ CHANGE, STORE, and STOREBEG functions. HOLD indicates that an object is to be put into a deletion-hold status. An object in deletion-hold status cannot be deleted by OSREQ DELETE or by OSMC expiration processing. NOHOLD indicates that an object is not to be placed in a deletion-hold status. This object can be deleted by OSREQ DELETE or by OSMC expiration processing. NOHOLD is the default value if DELHOLD is not specified on a STORE or STOREBEG request.
EVENTEXP(nn), added to the OSREQ CHANGE function. EVENTEXP lets you set the expiration criteria for an object that is currently in event-based-retention mode. The nnn value indicates that the object’s expiration date should be set to the current date + nnn days. If the specified object is not currently waiting for an event, this change request fails.

STIMEOUT, added to the OSREQ STOREBEG function. STIMEOUT specifies an interval after which OAM can free resources being held for the store sequence on behalf of the application. Use this keyword if you anticipate a long delay between individual store sequence requests, as a way to “keep alive” the resources between requests. This keyword does not apply to the DASD level of the OAM storage hierarchy, and is ignored if the store request is directed to DASD.

RETPD accepts a new value of -2 to indicate that event-based-retention is enabled for this object, for OSREQ CHANGE, STORE, and STOREBEG functions.

For more information about how to code and use these new functions, see z/OS DFSMS OAM Application Programmer's Reference.

Using expanded wildcards in OSREQ QUERY requests
z/OS R11 expands wildcard support for OSREQ QUERY searches. Previously, a generic search could be requested by putting an asterisk (*) in the right-most qualifier of the name. With R11, you can use one or more percent signs (%) and/or underscores (_) as wildcards anywhere in the object name.

Use the underscore to represent a single character in the object name. Use the percent sign to represent a string of characters (zero or more) in the object name. For instance, MIKES.MAIL.-IN is a fully qualified name and results in a single QE when a match is found. The names MIKES.MAIL.% and MIKES.M%.P_.L% are generic forms and can return multiple QEs when multiple objects exist that match the parts of the names specified. When multiple objects are returned, no ordering can be assumed.

For more information, see the OSREQ QUERY information in z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

Note: The percent sign and underscore characters are mutually exclusive with the asterisk; you cannot mix asterisk wildcards with either percent sign or underscore wildcards in a single QUERY request. Also, these wildcards are only supported for OSREQ QUERY requests; the new OSREQ wildcards do not work with the OSREQ TSO/E command processor.

Operating

You can use operator commands to display status for new functions that OAM provides in z/OS V1R11.

Before you begin: For more information about using the system commands, see z/OS MVS System Commands.

Displaying deletion- and retention-protection status
You can use the D SMS,STORGRP command to display the current retention-protection and deletion-protection status for a storage group.
1. Specify the D SMS,STORGRP command with a level of "DETAIL".
Example:

D SMS,STORGRP(sgname|ALL),DETAIL

The output from this example includes new RET PRO (retention-protection) and DEL PRO (deletion-protection) indicators, with the value Y indicating that the protection is in effect, and N indicating it is not in effect for the storage group or all storage groups, as specified.

Displaying IEFSSN.xx parameter settings for OAM1

You can use the D SMS,OAM command to show the settings that resulted from the parameters specified for the OAM1 entry in the IEFSSN.xx PARMLIB member, when the OAM1 subsystem was initialized at IPL time.

1. Specify the D SMS,OAM command.

Example:

D SMS,OAM

The output from this command includes settings for the OAM1 parameters:

CBR1100I OAM STATUS:

Explanation: The OAM status is:
...
...
OAM1 Parms: TIME=xxx MSG=xx UPD=x QB=x MOS=xxxx OTIS=x LOB=x DP=x

For more information about these OAM1 parameters, see z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

Diagnosing

This section describes new OAM diagnosis tasks and information.

Diagnosing problems related to expanded object sizes

The QSREQ API provides new return and reason code combinations to identify errors such as programming or environmental conditions that may occur as in relation to the expanded maximum object size. The new reason codes, for return codes 8 and 16, are documented in z/OS DFSMS OAM Application Programmer’s Reference and in z/OS DFSMSdfp Diagnosis.
Chapter 7. Using the DFSMShsm enhancements

In z/OS V1R11, DFSMShsm is enhanced with a new data set backup retention period, fast replication improvements, and enhanced support for migration level 1 (ML1) volumes.

**Data set backup retention period**

The DFSMShsm (H)BACKDS command allows users to create a backup version of a specified data set. This data set backup function is enhanced in z/OS V1R11 to allow users to specify the minimum number of days to retain a specific backup copy of a data set, using the RETAINDAYS keyword. The specified time period can be shorter or longer than the norm, and can be set to keep the copy for an indefinite amount of time. The retention period is applied when a backup copy is rolled-off, and during EXPIREBV processing.

The RETAINDAYS keyword is also added to the ARCHBACK macro and the ARCNBAK program. The (H)RECOVER command and ARCHRCOV macro have new TIME keywords to allow recovery by time as well as by date (retained copies can only be recovered by specifying the date, and optionally the time that the copy was created). The (H)BDELETE command and ARCHBDEL macro have a new ALL parameter which must be specified to delete all backup versions of the data set. If VERSIONS, DATE, or ALL are not specified, the command will fail. Note: prior to R11, the ALL keyword did not need to be specified when deleting all backup versions.

The ARCXTRCT command’s DATA=BUVERS option has new optional TIME and DATE parameters, to allow the display of additional backup versions, when the initial output exceeds 100 backup versions. ARCXTRCT DATA=BUVER also returns the RETAINDAYS value specified on the data set backup command.

The output from the LIST LEVEL(qualifier) and LIST DSNAMES(dsname) BCDS/BOTH commands also provides the RETAINDAYS value that is in effect for each backup version of a data set. A new SELECT keyword on these commands lets you select to display only active copies, or the backup copies that had RETAINDAYS specified at the time of data set backup.

**Fast replication enhancements**

In z/OS R11, DFSMShsm removes a restriction on the recovery of pre-allocated cataloged data sets that has been in effect since fast replication data set recovery was introduced in z/OS R8. Previously, those data sets could only be recovered if they resided on the same volume or volumes where they resided at the time of the backup. Deleted or moved data sets could not be recovered, because a catalog entry representing the state of the data set at the time of backup did not exist. DFSMShsm removes these restrictions by enabling the capture of the catalog information for a data set at the time of backup, and using that information to recover the data set if it has been moved or deleted since the last backup.

To use this enhancement, specify in a new SMS copy pool definition field that you want to capture catalog information for data set recovery, and use other new panel fields to specify up to ten catalog names to be associated with each copy pool version. You can specify that catalog capture is required or preferred. DFSMShsm captures data set catalog information based on the fields specified in the SMS copy pool definition and saves it to a DFSMShsm-managed data set, called the catalog information data set. DFSMShsm creates one catalog information data set for each copy pool version and places them on ML1
volumes. If you use this enhancement, you must ensure that you have sufficient space available on your ML1 volumes to accommodate catalog information data sets.

This enhancement also automatically unallocates each catalog in the copy pool before recovering a copy pool. This alleviates the need for the user to manually unallocate each catalog prior to copy pool recovers to avoid allocation failures. Normal catalog processing then reallocates the catalogs when they are accessed.

In related command updates, DFSMShsm adds a new DATASETS keyword to the LIST COPYPOOL command, allowing you to list all data sets that are backed up with a specified version, including the data sets that were backed up but have been deleted. The list output also indicates whether catalog information was captured for a specific copy pool version. The AUDIT COPYPOOLCONTROLS and AUDIT DIRECTORYCONTROLS commands are enhanced to verify if the catalog information data sets exist on ML1 volumes and if they correspond to valid copy pool versions.

Other new fields in the copy pool definition let you specify whether DFSMShsm should allow Metro Mirror primary volumes as the target of FRBACKUP and FRRECOV processing. You can also temporarily override the copy pool definition values for FRRECOV using the ALLOWPPRCP keyword on the FRRECOV command.

In a further enhancement, DFSMShsm now automatically re-initializes VTOCs that might become inaccessible because they were in a FlashCopy relationship when the relationship was withdrawn. Users previously had to re-initialize VTOCs in that situation. A new message, ARC1838I is added to report the automatic re-initializations.

**ML1 enhancements**

Beginning in V1R11, DFSMShsm enables ML1 overflow volumes to be selected for migration processing, in addition to their current use for data set backup processing. DFSMShsm enables these ML1 overflow volumes to be selected for migration or backup of large data sets, with the determining size values specified by a new parameter of the SETSYS command. Use the new ML1OVERFLOW parameter with the subparameter of DATASETSIZE(dssize) to specify the minimum size that a data set must be in order for DFSMShsm to prefer ML1 overflow volume selection for migration or backup copies.

In addition, DFSMShsm removes the previous ML1 volume restriction against migrating or backing up a data set whose expected size after compaction (if active and used) is greater than 65 536 tracks. The new limit for backed up or migrated copies is equal to the maximum size limit for the largest volume available.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation” on page 25</td>
<td>“Planning to use the DFSMS enhancements” on page 25</td>
</tr>
</tbody>
</table>
### Planning and Installation

This section describes the planning and installation tasks that you would perform for the DFSMS/hsm enhancements.

#### Planning to use the DFSMS enhancements

**Before you begin:** Be familiar with the following topics:
- For information about the DFSMS/hsm (H)BACKDS, (H)RECOVER, (H)BDELETE, and EXPIREBV commands, see [z/OS DFSMS/hsm Administration](#).

**Software requirements:** The backup enhancements require z/OS V1R11.

### Administering

This section describes the administration tasks that you would perform for the DFSMS/hsm enhancements.

#### Specifying a retention period for a data set backup copy

To specify a retention period for a copy of a backup data set, using the new RETAINDAYS keyword, you can use one of the following methods:
- (H)BACKDS command
- ARCHBACK macro
- ARCINBAK program.

The RETAINDAYS value must be an integer in the range of 0 to 50000, or 99999 (the “never expire” value).

**Related reading:**
- For information about using the (H)BACKDS command, see [z/OS DFSMS/hsm Storage Administration](#).
- For information about using the ARCHBACK macro, see [z/OS DFSMS/hsm Managing Your Own Data](#).
Providing access to the RETAINDAYS keyword

The new RETAINDAYS keyword on the BACKDS and HBACKDS commands is security protected, so a security profile must be defined and users must be given access to it before the keyword can be issued. For information about defining the FACILITY class profiles for this keyword, see z/OS DFSMShsm Implementation and Customization Guide.

Capturing catalog information for data set recovery

To allow for the recovery of data sets which no longer reside on the volumes where they were backed up or data sets that have been deleted, you can specify that DFSMShsm capture catalog information for backed up data sets. To do so, on the ISMF Copy Pool Define/Alter panel, specify R (for required) or P (for preferred) as the value for the new field “Capture catalog information for data set recovery.” On that same panel, specify up to 10 catalog names for copy pool processing, in the new fields provided. DFSMShsm will use these catalogs to capture all information about the data sets within a copy pool at backup time. The catalogs specified in the copy pool definition must reside on a volume within the copy pool, represent all data sets in the copy pool, and must not contain any data sets outside of the copy pool that it is defined in.

Related reading:

For more information about managing backups for fast replication, and filling in the new ISMF panel fields, see z/OS DFSMShsm Storage Administration including the following sections:
- “Saving Catalog Information at the Time of Backup”
- “Recovering a Data Set from a Fast Replication Backup”
- “FRVOLS | NOVOLS | DUMPVOLS | ALLVOLS | DATASETS: Listing a Specific Type of Copy Pool Information.”

Specifying the use of Metro Mirror volumes for backup and recovery using the copy pool definition

Before z/OS R11, the FRBACKUP and FRRECOV commands had the ALLOWPPRCP keyword added, to specify that Metro Mirror (PPRC) primary volumes should be targeted for backups and recovery, if they are available. Starting in z/OS R11, you need to make this specification using new fields in the SMS copy pool definition. The keyword will not be recognized on the FRBACKUP command. On the Copy Pool Define/Alter panel, the new fields are:
- FRBACKUP to Metro Mirror Primary Vols allowed . . ___ (NO, PN=PMNO, PP=PMPREF, PR=PMREQ or blank)
- FRRECOV to Metro Mirror Primary Vols allowed . . ___ (NO, PN, PP, PR or blank)

For each of these fields, valid values are:
- NO – no, do not use target Metro Mirror Primary volumes
- PN – PMNO, do not target Metro mirror Primary volumes
- PP – PMPREF, target Metro Mirror Primary volumes preferred
- PR – PMREQ, target Metro Mirror Primary volumes required
- Blank – none specified

If no value is specified, the default value is ”NO,” to not use Metro Mirror primary volumes for backup or recovery. Any value that you specify in these fields
becomes the new default. However, you can use the ALLOWPPRCP parameter on
FRRECOV command to override the value specified in the FRRECOV to Metro
Mirror Primary Vols field.

Related reading:
- For more information about this interface change, see the DFSMS section of
  z/OS
  Migration and see the new ISMF panel fields in
  z/OS DFSMShsm Storage Administration

**Enabling ML1 overflow volume selection for migration or
backup of large data sets**

To enable ML1 overflow volumes to be selected for migration or backup of large
data sets, specify the ML1OVERFLOW parameter on the SETSYS command. This
parameter lets you specify the minimum size that a data set must be for ML1
overflow volumes to be preferred, and the threshold amount for the ML1 overflow
volume pool to be filled. The subparameters are:
  - DATASETSIZE(dssize) – the minimum size in kilobytes of data, for which an
    ML1 overflow volume is preferred to migrate or backup the data set (using
    inline backup, HBACKDS or BACKDS commands, or the ARCHBACK macro).
    The default value is 200000.
  - THRESHOLD(threshold) – the limit for the percentage of occupied space in the
    ML1 overflow volume pool. When threshold is reached or exceeded, migration
    of data sets from level 1 volumes to level 2 volumes will start during secondary
    space management. The default value is 80%.

For example, you could use the following command to specify that ML1 overflow
volumes are preferred for data sets of 360 000 kilobytes or greater size, and that the
threshold for the ML1 overflow pool be set at 85% full.

**Example:**

```
SETSYS ML1OVERFLOW(DATASETSIZE(360000) THRESHOLD(85))
```

Related reading:
- For complete information about using the SETSYS command, see z/OS
  DFSMShsm Storage Administration

**Operating**

**Specifying a retention period on DFSMShsm commands**

You can specify a retention period for a copy of a back-up data set using the
(H)BACKDS operator command:

**Example:**

```
BACKDS dsname RETAINDAYS(number of days)
```

The number of retain days must be an integer in the range of 0 to 50000, or 99999
(the “never expire” value).
A new TIME keyword is added to the (H)RECOVER command. You can use this keyword to recover a backup version of a data set by time as well as by date. Specify the time when the backup version to be recovered was created, by hour, minute and second (hhmmss).

**Example:**

```
RECOVER dsname DATE(yyyy/mm/dd) TIME(hhmmss)
```

To determine the TIME value to specify in a RECOVER command, use the LIST LEVEL(QUALIFIER) and LIST DSNAME(dsname) BCDS/BOTH commands. Those commands display the RETAINDAYS value for each backup version of a data set, and the date and time when each was created.
Chapter 8. Using the DFSMSrmm enhancements

The functional enhancements available with z/OS V1R11 DFSMSrmm provide you with these benefits:

- **DFSMSrmm report generator**

  The DFSMSrmm Report Generator is updated. Using DFSMSrmm V1R11, you can now:
  
  - Override a data type within the dialog. The updated data type is saved in the report type and report definition. The report generator remembers the original data type and your override. You can use report type inheritance to benefit from any new data type values in report types.
  
  - Inherit changes in report types into existing report definitions. This enables changes in data types, comments, and other criteria to be merged into pre-existing report definitions. This enables your existing reports to benefit from improved report types shipped with DFSMSrmm.
  
  - Select which fields are to be excluded from total and break totals. When you specify that a field used for a report column is not to be subject to totaling, the report generator uses the NOST option with the ICETOOL reporting tool. When used with ICETOOL, all numeric fields are automatically totaled, unless you request that they be excluded.
  
  - Specify a list of field values and the text to be used for them in the report. The report generator uses existing field equate values to construct an initial list of possible values. Before you can use any of these, you must provide a new value to which the field will be changed for the report. Only those that have a change value are used for report generation.
  
  - Use substring for record selection criteria, regardless of the data type of the field.
  
  - Specify that the reports created from the DFSMSrmm report extract include the date and time when the extract was created. The report extract type and samples are updated to exploit this. You can do this for any type of report where the input records include one or more values that you want to include in the report title. This exploits new DFSORT ICETOOL capability to specify multiple report TITLE strings.
  
  - Use the equated assembler symbols, instead of the absolute value, when specifying field compare values used for record selection. An option in the dialog displays the available equates ready for you to use. When equates are available, the report generator uses these as a basic set to allow the use of an alternate value (called a ‘change value’) in the report. For example, ‘I’ can be changed to ‘INFO’.
  
  - Override the default ICETOOL processing for the generated ICETOOL statements when you specify the column width to be used. If no override is provided, the width is set to the larger of the column header text or the data size.
  
  - Use guidance information that is now provided within report types and report definitions. This information is presented to the user on request and when generating JCL. You can edit and add to this information, which is stored in the definitions. Existing report types and definitions are updated with help and guidance information about (for example) what variables to set in JCL or how to run the HSM preprocessor to convert data. The help
information is split into three parts; Type, Report, and JCL help. It is browsed or edited as a single set of help information.

All report types are updated to include the relevant data types and help information. In addition, all report samples are updated to inherit all new information from the report type, including the data type and help information.

- Manipulate records with DFSORT by using a new DFSMSrmm reporting tool. The output of this tool is not a report, but rather reformatted records. The JCL generated by the reporting tool includes comments that contain DFSORT symbol definitions, so that you can easily process the record further using DFSORT or ICETOOL.

**DFSMSrmm operational enhancements**

- **SEARCHVOLUME subcommand:** The DFSMSrmm TSO SEARCHVOLUME subcommand has additional operands to provide more flexibility and scope for querying a wider range of volume attributes, such as dates, actions, options, and flag settings.
- DFSMSrmm TSO command line parsing is improved by ensuring that similar values support the same syntax, reducing variance and enabling a wider scope of values.
- You can use MVS system commands to display, update, or deactivate any of the exits or exit modules for the new DFSMSrmm dynamic installation exits.
- Using the DFSMSrmm ISPF dialog, you have more flexibility in:
  - Adding new volumes. You can specify the volume type, storage group name, and creation date and time.
  - Navigating from data set information to the policies that are being used for the data set.
- DFSMSrmm vital record specifications (VRS) handling is simplified by allowing you to change location definitions after a VRS has been defined.

**DFSMSrmm programming enhancements**

- **DFSMSrmm EDGXHINT programming interface:** The EDGXHINT programming interface provides a function call based interface, rather than an executable assembler macro. EDGXHINT supports the return of associated error messages, in addition to the error codes.
- **DFSMSrmm API:** You can use a new MULTI=YES keyword on the EDGXCI executable macro to specify that your code can handle more than one entry in the return work area. Depending on the size of your work area DFSMSrmm can now return all the prepared entries in a single call. DFSMSrmm attempts to fit as many entries as it can into the work area, up to its internal limit. The internal limit depends on the resource being searched. To obtain the next/remaining entries you use the REQUEST=CONTINUE keyword as today.
- Multiple installation exit routines are supported for each of the DFSMSrmm installation exits by exploiting the z/OS Dynamic Exit Facility for all exits.

**DFSMSrmm ease of use enhancements**

- You can now use the new EDGUPDT utility to process the CDS transactions from test and recovery systems back into the original CDS. This ensures that, even after testing or recovery exercises you can ensure that the DFSMSrmm CDS reflects the actual content of the tape volumes in your library.
- Based on a new JRNLTRAN parmlib option, DFSMSrmm now additionally writes the previous level of the updated record to the journal along with the updated record and updated, related records. This option results in additional journal records being written each time a CDS record is updated.
- The DFSMSrmm tape utility EDGINERS is updated to support the reading and cross-verification of label information with the records defined in the DFSMSrmm control data set.
- New DFSMSrmm parmlib options provide flexibility in how tape generation data sets are managed for cyclic retention. You no longer need a USERMOD to enable duplicate generation retention.
- The sample "volume not in library" installation exit, CBRUXVNL, is enhanced to enable fewer installations to require customization. Volumes will be requested to be entered into the tape library whenever possible.

- **DFSMSrmm enterprise enablement**
  - *SMI-S Storage Library Profile:* The DFSMSrmm CIM agent has been updated to support CIM level 2.17. CIM agent and providers have been updated to run with OpenPegasus 2.8.1.

---

### Planning and Installation

This section describes the planning and installation tasks that you would perform to use the DFSMSrmm enhancements.

#### Planning for migration

You should plan to perform the following migration tasks, if required:

- Use DFSMSrmm parmlib options instead of USERMOD to specify how VRSEL processing handles duplicate generation data groups.
  
  Before z/OS V1R11, a USERMOD could be applied to the EDGVREC load module to influence how VRSEL processing handles duplicate generation data groups (GDG). The USERMOD may be called RMDUPGD. Starting with z/OS V1R11, this USERMOD is no longer supported. You must now use the GDG operand of the OPTION command in parmlib member EDGRMMxx to specify how VRSEL processing handles duplicate generations.

- Update operator procedures and system automation for dynamic installation exits in DFSMSrmm.
  
  Before z/OS V1R11, installation exits were loaded by DFSMSrmm at initialization time. Starting with z/OS V1R11, z/OS Dynamic Exit Services is used to load and activate the default (EDGUXn00) exit modules at initialization time. This change requires changes to DFSMSrmm operating procedures and system automation (if any).

- Check Rexx execs used with DFSMSrmm subcommands for obsolete stem variables.
  
  Before z/OS V1R11, DFSMSrmm, when creating Rexx variables for SEARCH subcommands, returned these variables as stem variables and created a .0 stem variable for almost all stem variables. Starting with z/OS V1R11, DFSMSrmm will create a .0 stem variable only for the key variable for each SEARCH command.

See *z/OS Migration, GA22-7499* for a complete description of these migration tasks.

#### Planning to use DFSMSrmm enhancements

**Exploiting report generator enhancements**

An action is required only if you want to exploit new capability within the report types for existing report definitions. After the first IPL, you can use the DFSMSrmm report generator dialog ‘Report Migration Tasks’ to cause new information shipped in report types to be inherited into existing report definitions.
Once you have completed this task, do not share the updated report definitions with a lower release of the DFSMSrmm report generator.

For more information on the DFSMSrmm ‘Report Migration Tasks’ dialog; see **z/OS DFSMSrmm Reporting**.

**Exploiting DFSMSrmm operational enhancements**
You should examine any existing operating procedures to see whether any of the enhancements to the TSO commands or ISPF panels can be used to improve the procedures. This should be done in coordination with any changes made in a migration task.

**Exploiting DFSMSrmm API enhancements**
Existing DFSMSrmm application programs should be examined to see whether they can be improved by using one of the DFSMSrmm API enhancements and plan for the needed updates.

For more information, see **z/OS DFSMSrmm Application Programming Interface**, SC26-7403.
## Part 2. Using New DFSMS Functions in z/OS V1R10

### Chapter 9. Using extended address volumes
- **Planning and installation**
  - Setting up support for extended address volumes
- **Administering**
  - DFSMSshm administration
  - DB2 Administration
- **Application programming**
  - Modifying applications to use extended address volumes
  - Finding and Changing Programs that Issue EXCP, EXCPVR, or XDAP
    - Finding Programs that Do Not Issue OPEN
    - But Issue EXCP, EXCPVR or XDAP
    - Finding Programs that Issue an OPEN and Issue EXCP, EXCPVR or XDAP
    - Changing programs that issue EXCP, EXCPVR or XDAP
  - Finding Programs that may need to be changed to support EAV – DFSMS Migration Assistance
  - General information about the tracker
  - DFSMS Instances to track
- **Operating**
  - Using commands with extended address volumes
  - Diagnosing problems with extended address volumes

### Chapter 10. Using VSAM RLS secondary lock structures
- **Planning and Installation**
- **Setting up VSAM RLS secondary lock structures**
- **Administering**
  - Monitoring the use of secondary lock structures
- **Operating**
  - Activating and deactivating secondary lock structures
  - Displaying secondary lock structures
  - Deleting secondary lock structures

### Chapter 11. Using the PDSE enhancements
- **Operating**
  - Displaying PDSE 64-bit virtual storage buffer usage
  - Displaying caching statistics for PDSE data in hiperspace
  - Command output example

### Chapter 12. Using the SMS enhancements
- **Activating the Data Class Space Override**
- **Specifying a system-determined block size using data class**
- **Using the new read-only variables for ACS routines**
- **Using the message enhancements**

### Chapter 13. Using the object access method enhancements
- **System Programming**
  - Specifying object sizes up to 2000 MB
  - Specifying whether to retrieve backup volume information on the OSREQ QUERY command
  - Configuring automatic access to backup for OAM
  - Limiting CBROAMxx statements to one system, or all systems in an OAMplex
- **Administering**
- **DB2 administration**
- **Application Programming**
  - Specifying storage for large objects in parts
  - Operating
    - Marking tape and optical volumes as permanently full
    - Displaying the new automatic access to backup settings
    - Retrieving backup copies of objects on volumes marked lost or not defined
    - Stopping all OSMC processing

### Chapter 14. Using the DFSMSshm and DFSMSdss enhancements
- **Planning and Installation**
- **Administering**
  - Programming
    - Specifying a new name for DFSMSdss dumped data sets
- **Operating**
  - Specifying a new name on the (H)BACKDS command

### Chapter 15. Using the DFSMSrmm enhancements
- **Planning and Installation**
- **Administering**
  - Programming
    - Specifying a new name for DFSMSrmm dumped data sets
- **Operating**
  - Specifying a new name on the (H)BACKDS command
  - Displaying the new automatic access to backup settings
  - Retrieving backup copies of objects on volumes marked lost or not defined
  - Stopping all OSMC processing
Chapter 9. Using extended address volumes

Before z/OS V1R10, DASD storage was limited to 65,520 cylinders per volume. To satisfy growing DASD storage requirements, z/OS V1R10 supports extended address volumes. An extended address volume (EAV) is by definition 65,521 cylinders or larger. In this release the maximum size is 262,668 cylinders.

The extra space on an extended address volume (the cylinders whose addresses are equal to or greater than 65,536) is referred to as the extended addressing space (EAS). These cylinder addresses are represented by 28-bit cylinder numbers. On an extended address volume, the cylinders whose addresses are below 65,536 are referred to as the base addressing space. These cylinder addresses are represented by 16-bit cylinder numbers or by 28-bit cylinder numbers whose high order 12 bits are zero.

The track address with 28-bit cylinder numbers has the following format: track address is a 32-bit number that identifies each track within a volume. It is in the format hexadecimal CCCCcccH, where CCCC is the low order 16 bits of the cylinder number, ccc is the high order 12 bits of the cylinder number, and H is the four-bit track number. For compatibility with older programs, the ccc portion is hexadecimal 000 for tracks in the base addressing space. Note: 28-bit cylinder arithmetic and compares with these nonlinear cylinder address (CCCCccc) could be error-prone. A new macro, TRKADDR, is provided for general use to do conversions and compares. See "Modifying applications to use extended address volumes" on page 39 for details.

Extended address volumes are divided into track-managed space and cylinder-managed space.

Track-managed space is the space on a volume that is managed in tracks and cylinders. Track-managed space ends at cylinder address 65,519. Each data set occupies an integral multiple of tracks. Track-managed space also exists on all non-EAV volumes.

Cylinder-managed space is the space on the volume that is managed only in multicylinder units. Cylinder-managed space begins at cylinder address 65,520. Each data set occupies an integral multiple of multicylinder units. Space requests targeted for the cylinder-managed space will be rounded up to the next multicylinder unit. The cylinder-managed space only exists on EAV volumes. An EAS-eligible data set is one that can be allocated on cylinder numbers 65,536 or greater.

EAS-eligible data sets allocated on an EAV are created with format 8 and format 9 DSCBs in the VTOC. A format 8 DSCB is equivalent to a format 1 DSCB and the format 9 DSCB provides for additional attribute data and a set of pointers to possible format 3 DSCBs. The purpose of these DSCBs is to indicate that the extent descriptors contained in a format 8 or format 3 DSCB may contain 28-bit cylinder numbers. Programs may need to be modified to support extended attribute DSCBs. For more information, see "Modifying applications to use extended address volumes" on page 39.

The following table lists the types of tasks and associated procedures that you must complete to use extended address volumes.
Planning and installation

This section describes the planning and installation tasks that you must perform to use extended address volumes.

Planning to use extended address volumes

Before you begin: To understand the details of extended address volumes, see z/OS DFSMSdfp Advanced Services. To understand the concepts of DASD volumes in general, see z/OS DFSMS Using Data Sets. To understand how the storage administrator uses IGDSMSxx parameters, see z/OS MVS Initialization and Tuning Reference. For the related DFSMSdss updates, see z/OS DFSMSdss Storage Administration and the DFSMSdss section of the z/OS DFSMSdfp Storage Administration.

Software requirements: Extended address volumes require z/OS V1R10.

Hardware requirements: A Licensed internal code (LIC) upgrade to an IBM DS8000 is needed so that an EAV can be configured in the storage subsystem.

Coexistence requirements:

- Systems prior to z/OS V1R10 will not be able to bring an extended address volume online. Verify that applications which use an extended address volume (EAV) have the proper software levels installed.

- Previous releases of z/OS provide toleration of EAVs – extended address volumes can be defined in the IODF on lower level systems, but are not allowed to come online.

- The DEVSEVR PATHS operator command is updated to report a new field for the number of cylinders. The DEVSEVR QDASD operator command is changed to recognize and display the larger device size for an EAV. This larger size is reflected in the CYL column of the DEVSEVR QDASD display.

- LSPACE macro processing on pre-V1R10 systems is changed to accept the extended parameter list and not fail it with return code 12. LSPACE macro processing is also changed to treat EXPDATA= or XEXPMGS= requests as DATA= or EXPMGS= respectively. This change allows LSPACE programs that are assembled using the new z/OS V1R10 keywords to run on pre-R10 systems.

- DFSMSdss allows restore of a data set using dump from an EAV on V1R10 to a non-EAV on V1R9 or lower. If you attempt to restore a data set using dump
from an EAV on R10 to a non-EAV on R9 (or lower) without this support (or lower), you may have unpredictable results.

- DFSMSdss full volume and tracks dumps of EAVs are not compatible with dumps of volumes that are 64K cylinders or fewer, due to changes required to format the extended-address space in the dump.
- DFSMSdss provides limited restore capability on supported lower levels of z/OS for data dumped from an EAV on z/OS V1R10 or higher levels.
- The following restore functions and limitations are supported via a coexistence PTF on pre-z/OS V1R10 systems
  - Versions of DFSMSdss prior to V1R10 will not perform a full volume restore of a full volume dump from an EAV. In addition, tracks restore where track 0 is included will fail and tracks restore, not including track 0, will restore only the track-managed space from an EAV. Logical data set restore and physical data set restore will convert format 8 and 9 DSCBs to format 1 DSCBs.
  - DFSMSdss Stand-Alone Restore will not perform a full volume or tracks restore of an EAV to a non-EAV.

- DFSMSHsm adds support for Recall/Recover on R9 or lower of an EAS data set with a format 8 DSCB that was migrated or backed up on z/OS V1R10.

**Supported data sets**: EAS-eligible data sets are those that can be allocated anywhere on an extended address volume. z/OS V1R10 supports the following types of data sets as EAS-eligible data sets:
- SMS managed VSAM (all types)
- Non-SMS VSAM (all types)
- VSAM data sets inherited from prior physical migrations or copies
- VSAM temporary data sets
- zFS data sets (they are VSAM).

Non-EAS eligible data sets are those that can be allocated only in the track-managed space of an EAV volume. Non-EAS eligible data sets include:
- Page data sets
- Catalog data sets (BCS and VVDS)
- VSAM data sets with imbed or keyrange attributes that may have been inherited from prior physical migrations or copies
- Non-VSAM data sets.

**Planning steps:**
- To help determine which applications might need to be modified to support EAVs, see “Application programming” on page 39 and information about the EAV migration assistance tracker in z/OS DFSMSdfp Advanced Services.

**Setting up support for extended address volumes**

**Related reading**: For more information, see “Initializing SMS through the IGDSMSxx member” in the z/OS DFSMSdfp Storage Administration, and z/OS MVS Initialization and Tuning Reference.

Perform the following steps to set up support for extended address volumes:

1. Allow SMS to select extended address volumes during its volume selection processing. Set the USEEAV parameter in the IGDSMSxx PARMLIB member to YES: USEEAV(YES). This specifies at the system level that SMS may select an
EAV during volume selection processing. SMS checks this value for new allocations and when extending data sets to a new volume.

You can also use the SETSMS operator command to change the value of this keyword without having to re-IPL. The syntax of the operator command is: SETSMS USEEAV(YES\|NO). The modified setting remains in effect until the next IPL, when it reverts to the value specified in the active IGDSMSxx member of PARMLIB.

2. In the IGDSMSxx PARMLIB member, specify an optional break point value for SMS to use when making volume selection decisions for VSAM data sets. On the BreakPointValue parameter, specify a number of cylinders (0-65520). If the BreakPointValue (BPV) keyword is not specified in IGDSMSxx, SMS uses a default value of 10 cylinders. When a primary or secondary disk space request is made for a data set that is eligible for cylinder-managed space and the request is for this size or more, the system will prefer to use the cylinder-managed space for that extent. Otherwise the system will prefer to use track-managed space for that request. This applies to each request for primary or secondary space for data sets that are eligible for the cylinder-managed space.

SMS uses the break-point value when making volume selection decisions for VSAM data sets only. You can also specify a break-point value for each storage group separately, by specifying it on a Define Storage Group panel. For more information, see z/OS DFSMS Implementing System-Managed Storage. A break-point value at the storage group level overrides any value in the IGDSMSxx PARMLIB member.

In addition to the break-point value setting, the storage group also lets you specify a setting for track-managed free space thresholds (high and low). These thresholds are factored into the SMS volume selection algorithms and DFSMShsm space management.

**Administering**

This section describes administration tasks for extended address volumes.

**DFSMShsm administration**

Administration tasks for DFSMShsm include:

- Performing space management using both the volume threshold (an existing process) and the track-managed threshold (new).
- Setting a track-managed threshold, in one of two ways. For SMS extended address volumes the track-managed threshold is acquired from the storage group. For non-SMS extended address volumes you can specify a parameter on the ADDVOL command (see the z/OS DFSMShsm Storage Administration for more information). The purpose of the track-managed threshold is to ensure that the track-managed space gets managed separately, even if the overall volume has not exceeded the extended address volume threshold value.
  - Storage group (SMS-managed data) – DFSMShsm will use the newly defined track-managed low and high threshold values associated with each storage group.
  - DFSMShsm ADDVOL command – a new TRACKSMANAGEDTHRESHOLD (TMT) keyword is added. Use this keyword to specify high and low thresholds for the track-managed space of a non-SMS EAV Level-0 volume. If TMT is not specified, then the corresponding values for THRESHOLD will be used.
IBM recommends that in z/OS 1.10 you use extended address volumes as DFSMShsm primary volumes since only VSAM data sets can reside in the cylinder managed space.

- DFSMShsm adds support for Virtual Concurrent Copy (VCC) for the following functions:
  - Data set backup (Management Class value and data set backup keyword)
  - ABARS (Management Class value)
  - Control Data Set Backup (Management Class value)
  - Volume Dump (SETSYS value).

For more information about these updated DFSMShsm administration tasks, see *z/OS DFSMShsm Storage Administration*.

**DB2 Administration**

DB2 administrators: apply the appropriate toleration PTFs before placing any BSDS or active logs on extended address volumes, or before using administrative enablement procedures on extended address volumes.

---

**Application programming**

After you have enabled support for extended address volumes, you might need to modify applications to work with them.

**Modifying applications to use extended address volumes**

The following types of applications might need modifications for extended address volumes.

1. Applications that open a VTOC that might have format 8 DSCBs or open a data set that has a format 8 DSCB for EXCP access must provide a DCBE macro with the new EADSCB=OK keyword. The EADSCB=OK keyword indicates that the caller supports extended attribute data provided in DSCBs and track addresses greater than 65520. These DSCBs may contain 28-bit cylinder address extent descriptors. Specifying EADSCB=OK indicates that the caller supports 28-bit cylinder addresses and format 8 and 9 DSCBs.

Applications using the following system services might also have to be changed to support extended attribute DSCBs. These services also have the new keyword EADSCB=OK:

- OBTAIN and CAMLST with SEEK or SEARCH options
- CVAFDIR
- CVAFSEQ
- CVAFSM with ACCESS=MAPDATA, MAP=VOLUME, RTA4BYTE=YES
- CVAFFILT.

CVAFDIR also has a new MULTIPLEDSCBS keyword that provides support for the reading and writing of multiple DSCBs in one CVAFDIR call. The CAMLST with SEARCH/SEEK options and OBTAIN macros also have a new NUMBERDSCB keyword that provides support for reading multiple DSCBs in one OBTAIN call.

For more information about changes to these macros, the new keywords and format 8 and 9 DSCBs, see *z/OS DFSMSdfp Advanced Services*.

2. Applications using interfaces that deal with DASD volume capacity:

- The LSPACE macro returns information about a DASD volume. Returned information can be in character or binary format. Returned information may...
include larger fields, as well as new kinds of information. New parameters are added to this macro: XEXPMSG=addr, EXPDATA=addr, DATATYPE=(parm1,parm2,...) .

DATATYPE is an optional keyword that specifies the type of data that is to be returned to the caller. Any combination of the following parameters may be specified:
- VOLUME - Information about free space (tracks and cylinders) on the volume
- VTOC - Information about free space (DSCBs) in the VTOC
- INDEX - Information about free space (index records) in the VTOC index
- FRAGINDEX - The fragmentation index
- ALL - All available LSPACE statistics.

- Output of IEHLIST LISTVTOC – to handle larger numbers, many IEHLIST LISTVTOC fields are shifted right on the page for all volumes (EAV and non-EAV).
- Output of IDCAMS DCOLLECT – the volume information (type V), storage group volume information (type VL), and storage group (type SG) records created by the DCOLLECT command have new space statistics.
- DEVTYPE macro – The existing INFO=DASD parameter of the DEVTYPE macro returns the number of cylinders in a 16-byte area provided by the caller. New fields returned include an indication that cylinder-managed space exists on the volume, an indication that extended attribute DSCBs, format 8 and format 9 DSCBs, are allowed on the volume, the minimum allocation size in cylinders for cylinder-managed space, the first cylinder address where cylinder managed space begins, and the block size of the index.
- ANTRQST API used to query Peer-to-Peer Copy relationship information – the ANTPQMAP mapping has new fields to accommodate larger volume capacity.

3. Programs that use DASD channel programs and messages that might deal with a track in the cylinder-managed space.

You can compare two of these track addresses for equality but you cannot reliably use a simple comparison for greater than or less than. Any arithmetic must take this special format 28-bit cylinder number into consideration. IBM recommends using the new TRKADDR macro for all track address comparisons and calculations. This macro is available in z/OS V1R10 but the expansion will run equally well on downlevel systems if the high order 12 bits of the track number are zero. Programs that are written in a high level language such as C, C++, Cobol, or PL/I can call a new routine named IECTRKAD that performs the same functions as the TRKADDR macro. This routine is available only in z/OS V1R10, but programs linked with it can run on downlevel releases.

For programs that use messages: the track addresses in the output of IDCAMS LISTCAT continue to be printed in eight hex digits in the native format of the device, which is the format of CCCCcccH. This means that they can be compared for equality but cannot be compared for greater than or less than unless your program converts them to another format such as with the new TRKADDR macro or IECTRKAD routine. The track addresses in the output of IDCAMS LISTDATA PINNED are presented in a modified report that lists each data set name associated with a range of pinned tracks (previously the report listed each data set name associated with each pinned track address). The pinned track addresses are in the format CCCCcccH. The modified report format applies to all volumes, not only EAVs.

4. NaviQuest provides the following new storage group attributes:
• Breakpoint value.
• Migration high track. This has the same meaning as the existing migration high number but it applies only to track-managed space.
• Migration low track. This has the same meaning as the existing migration low number but it applies only to track-managed space.

5. Change programs that examine SMF record types 14, 15, 18 and 19.

New fields are added to SMF record types 14, 15, 18, and 19 to support extended address volumes.
• The EOV and CLOSE functions for a DCB write a type 14 or 15 record. These record types add a flag to indicate whether the DCBE has EADSCB=OK specified; this flag is intended to help find programs that do not have this specification, to help with migration to EAV.
• Record types 14 and 15 also add another new flag to indicate that an application has opened a VSAM dataset or VTOC specifying a basic access method (BSAM, or QSAM) but also issued an EXCP or XDAP macro. Hence, the application might need modifications to support an EAV eligible data set.
• Record type 18 has a new flag to indicate that the record is a continuation record for a multi-volume data set, where the volume being processed by Rename is not the first volume.
• Record type 19 has expanded fields to describe free space for the volume, with new fields for free space from track-managed space, and for the total space on the volume.

For more information about the SMF record changes, see [z/OS MVS System Management Facilities (SMF)](https://www.ibm.com/support/knowledgecenter/en/SSSHCR_2.3.0/cr15066.html).

Finding and Changing Programs that Issue EXCP, EXCPVR, or XDAP

Programs that build channel programs for VSAM data sets must be changed if the data set has cylinders with numbers greater than 65,535. This is because it is unlikely that such a program would work without a change to support 28-bit cylinder numbers for the track address. In a later z/OS release IBM expects to support non-VSAM data sets that will find this problem.

In order to facilitate migration of programs that build channel programs, the SMF type 14 and 15 records will have a new bit that indicates whether one or more instances of EXCP or XDAP were issued for a non-EXCP DCB. The XDAP macro results in issuing EXCP, so they are indistinguishable.

Finding Programs that Do Not Issue OPEN But Issue EXCP, EXCPVR or XDAP

Each of these macros requires a DCB and a DEB. If the application program builds its own DEB, it can do that only if it is authorized. The operating system cannot detect these types of instances when they are directed to an extent that has 28-bit cylinder numbers. In other words the system cannot tell whether the program has been upgraded to handle 28-bit cylinder numbers. It will be necessary for a programmer to search source code and run tests to find these programs.

Finding Programs that Issue an OPEN and Issue EXCP, EXCPVR or XDAP

Two kinds of program issue an OPEN macro and then issue EXCP, EXCPVR or XDAP. Both types of programs can be found by reading source code and by examining SMF records. The types are:
The DCB has MACRF=E, which signifies that the DCB is for EXCP. This signifies that the DCB is only for EXCP and bit 0 (DCBMRECP) in DCBMACRF is on.

The DCB has a MACRF value other than E, which signifies that the DCB is for a regular access method.

**EXCP OPEN and Issue EXCP, EXCPVR or XDAP:** To find instances of these EXCP DCBs the programmer can search source code and he also can examine SMF type 14 and 15 records. One of these records is written when the user makes the transition to another volume or issues a CLOSE macro for a non-VSAM data set. The two-byte SMFDCBMF field contains a copy of the DCBMACRF field. If bit 0 is on, it means that the DCB is only for EXCP (MACRF=E). The JFCB will contain the data set name unless the OPEN is for a partitioned concatenation.

If the program issues an OPEN for an EXCP DCB, OPEN can determine whether the program has been upgraded to handle EAV because the program must supply a DCBE with the EADSCB=OK keyword coded. In z/OS 1.10 if an attempt is made to open an EAS eligible data set (VSAM) where a format-8 DSCB exists or open the VTOC and this EADSCB=OK keyword is not coded, a new ABEND and MSGIEC142I 113-44 will be issued. In a future release where other data set types will be EAS eligible, then this condition will occur for them also.

If the new bit SMF14EADSCB in the SMF type 14 or 15 record is zero, then the program did not specify EADSCB=OK on the DCBE macro. If you expect to use this program with a data set that has a format 8 DSCB in z/OS 1.10 or later, then IBM recommends that you upgrade the program to handle 28-bit cylinders and code EADSCB=OK.

**Non-EXCP OPEN and Issue EXCP or XDAP:** The DCB is for an access method other than EXCP. In this case bit 0 of DCBMACRF and in SMFDCBMF are not 1. z/OS DFSMSdfp Advanced Services states that it is valid to issue EXCP but not EXCPVR in this case. An EXCPVR would result in ABEND 400. Programmers should search source code for these instances of EXCP and XDAP. In addition, the following data in SMF will be recorded:

- New SMF14/15 flag, SMF14EXCPBAM, is defined to aid in the finding of programs that issue a Non-EXCP OPEN and issue EXCP or XDAP. This flag will be set on when the access method of BSAM, QSAM or BPAM was used and the user program issued one or more instances of the EXCP or XDAP macro since the DCB was opened.

If the flag SMF14EADSCB in this SMF record is zero, then the program did not specify EADSCB=OK on the DCBE macro. Upgrade the program to handle 28-bit cylinders and code EADSCB=OK.

**Changing programs that issue EXCP, EXCPVR or XDAP**

The simplest way to handle 28-bit cylinder numbers is to use the TRKADDR macro. It will work equally well with track addresses that contain 28-bit and 16-bit cylinder numbers and with any DASD data set. It also will work correctly on systems before EAV is supported in z/OS 1.10. Switching to TRKADDR is planning work for a future release where non-VSAM data sets will be supported in EAS.

In a future release, issuances of EXCP or XDAP will be failed for a non-VSAM data set when (1) the access method is BSAM, BPAM or QSAM, and (2) the data set has a format 8 DSCB. The techniques described in the prior sections should be used to
find and upgrade these programs and add the EADSCB=OK option to the DCBE macro. The upgraded program will continue to operate correctly on downlevel systems.

Finding Programs that may need to be changed to support EAV – DFSMS Migration Assistance

The EAV migration assistance tracker has three objectives. The first is to identify executions of select systems services by job and program name where the invoking programs may require analysis for changes to use new services provided by the system service. These program calls and the reported output are not considered in error as valid information is returned. These will be considered as informational instances.

The second objective is to identify the possible improper use of returned information, like parsing 28-bit cylinder numbers in output as 16-bit cylinder numbers. These will be considered as warning instances.

The third objective is to identify instances that will either be failed or identified with an informational message if they are run on an extended address volume. These will be considered as error instances. These errors apply to the following functions when the target volume of the operation is a non-EAV and the function invoked did not specify the EADSCB=OK keyword:

- DADSM OBTAIN
- CVAFDIR
- CVAFSEQ
- CVAFDSM
- CVAFFILT
- OPEN of VTOC
- DCB OPEN of an EAS eligible data set.

By adding these instances to the EAV migration assistance tracker allows the system programmer to identify these instances by job and program name without failing programs and provides the capability to exclude instances that may not be currently ready for evaluation.

This set of errors will continue to result in failed programs, as implemented today, if the system service is issued for an EAV without the EADSCB=OK keyword specified.

General information about the tracker

The tracking of EAV migration assistance instances will be done using the Console ID Tracking facility that was provided in z/OS 1.6.

The tracker function will allow component code to register tracking information as a text string of its choosing up to 28 characters. The tracker will record this as a unique instance appending additional information to it like job name, program name, and count of occurrences to avoid duplicates. The tracker will allow an exclusion list, via a SYS1.PARMLIB member, to be specified that will prevent an instance from being recorded. Instances already verified would then not show up in the tracker. The exclusion list filters will be the registered tracking information, job name and program name. Wild-carding (? and *) will be allowed for these fields in the exclusion list.
In addition to the above services, the tracker will allow with an operator command
to activate the tracker, activate new exclusion lists, report on recorded instances,
abend and dump on recorded instances, deactivate the tracker and provide an IBM
internet ID where a customer can provide instances of programs recorded in the
tracker to IBM.

The tracking facility can be manipulated with the following commands:
• The SETCON operator command, which is used to activate and deactivate the
  Console ID Tracking facility.
• The DISPLAY OPDATA, TRACKING operator command, which is used to
display the current status of the Console ID Tracking facility, along with any
recorded instances of violations.
• The CNIDTRxx dynamic parmlib member, which is used to list violations that
  have already been identified in order to prevent them from being recorded
  again.

See Appendix A in z/OS MVS Planning: Operations for more information on this
tracking facility.

**DFSMS Instances to track**
The following are the EAV migration assistant instances that are to be recorded.

See the EAV migration assistant section in z/OS DFSMSdfp Advanced Services for
details on the instances that would be tracked.

**LSPACE (SVC 78):** An LSPACE request with the DATA=, MSG=, or EXPMSG=
keywords was issued. Additional data from track-managed space is available with
the EXPDATA= and XEXPMSG= keywords.

When this instance occurs for any volume type, it will be recorded in the tracker as
an informational message.

**DEVTYPE (SVC 24):** A DEVTYPE request with DEVTAB or UCBLIST without
INFOLIST, returns the number of cylinders on the volume. This is in a two-byte
field at offset 8, which is too small if the volume has more than 65 520 cylinders.
Consider using INFO=DASD which returns the number of cylinders in a four-byte
field.

When this instance occurs for any volume type, it will be recorded in the tracker as
an informational message.

**IDCAMS LISTDATA PINNED:** A LISTDATA PINNED request was processed.
The track addresses for the PINNED tracks may contain 28-bit cylinder numbers.

When this instance occurs for any volume type, it will be recorded in the tracker as
a warning message.

**IEHLIST LISTVTOC:** An IEHLIST LISTVTOC request was processed. Extent
descriptors may contain cylinder addresses 65 520 or larger. Free space descriptors
may contain track addresses 982800 or larger and/or full cylinders 65 520 or larger.
The generated report will display the information in different columns as
compared to reports generated on releases prior to z/OS V1.10.

When this instance occurs for any volume type, it will be recorded in the tracker as
a warning message.
**IDCAMS DCOLLECT:** An IDCAMS DCOLLECT request for 'V' (Volume Record Field) and 'VL' (SMS Volume Definition Field) records was processed. Additional data for track-managed space was recorded.

When this instance occurs for any volume type, it will be recorded in the tracker as an informational message.

**IDCAMS LISTCAT:** An IDCAMS LISTCAT request was processed that printed extent descriptors for one or more EAS eligible data sets (VSAM in z/OS V1R10). The returned extent descriptors may contain 28-bit cylinder numbers. This instance is recorded for both EAS and non-EAS capable volumes. Please note that AMS LISTCAT output format may change as a result of service and new function support. IBM recommends applications processing LISTCAT output be updated to obtain results directly from the Catalog Search Interface (CSI). For more information on CSI, see [z/OS DFSMS Managing Catalogs](https://www.ibm.com/support/knowledgecenter/SSY278_6.1.0/com.ibm.zos.zos.doc/index.html) and [HLASM Programmer's Guide](https://www.ibm.com/support/knowledgecenter/SSY278_6.1.0/com.ibm.zos.zos.doc/hlasmpg/g_overview.html).

When this instance occurs for any volume type, it will be recorded in the tracker as a warning message.

**DADSM OBTAIN (SVC 27):** DADSM OBTAIN was issued with the search or seek option to a non-EAV volume. The caller did not specify with EADSCB=OK that it supports the extended attribute DSCBs and the target data set is EAS-eligible.

When this instance occurs for a non-EAV volume type, it will be recorded in the tracker as an error message.

**CVAFDIR:** CVAFDIR was issued with the search or seek option to a volume that does not support extended attribute DSCBs. The caller did not specify with EADSCB=OK that it supports the extended attribute DSCBs and the target data set is EAS-eligible. CVAF return code 4 and CVSTAT of X'52' would have been set if issued to a volume that supports extended attribute DSCBs.

When this instance occurs for a volume that does not support extended attribute DSCBs, it will be recorded in the tracker as an error message.

**CVAFSEQ:** CVAFSEQ was issued for physical sequential or index order to a volume that does not support extended attribute DSCBs. The caller did not specify with EADSCB=OK that it supports the extended attribute DSCBs and the target data set is EAS-eligible. CVAF return code 4 and CVSTAT of X'52' would have been set if issued to a volume that supports extended attribute DSCBs.

When this instance occurs for a volume that does not support extended attribute DSCBs, it will be recorded in the tracker as an error message.

**CVAFDSM:** CVAFDSM was issued to retrieve unallocated space on a volume that does not support extended attribute DSCBs. The caller did not specify with EADSCB=OK that it supports the extended attribute DSCBs. CVAF return code 4 and CVSTAT of X'52' would have been set if issued to a volume that supports extended attribute DSCBs.

When this instance occurs for a volume that does not support extended attribute DSCBs, it will be recorded in the tracker as an error message.
**CVAFFILT**: CVAFFILT was issued to obtain DSCB information for fully or partially qualified data set names on a volume that does not support extended attribute DSCBs. The caller did not specify with EADSCB=OK that it supports the extended attribute DSCBs and the qualified data set is EAS-eligible. CVÅF return code 4 and CVSTAT of X'56' along with data set name status in the FCL (FCLDSNST) of X'06' would have been set if the request was issued to a volume that supports extended attribute DSCBs.

When this instance occurs for a volume that does not support extended attribute DSCBs, it will be recorded in the tracker as an error message.

**DCB Open of VTOC**: A DCB Open of a VTOC was issued to a volume that does not support extended attribute DSCBs. The caller did not specify with EADSCB=OK on the DCBE macro that it supports the extended attribute DSCBs in the VTOC. Open would have issued an ABEND, MSGIEC142I 113-48 if an attempt was made to open the VTOC of a volume that supported extended attribute DSCBs.

When this instance occurs for a volume that does not support extended attribute DSCBs, it will be recorded in the tracker as an error message.

**DCB Open of EAS eligible data set (VSAM)**: A DCB Open (MACRF = E) of an EAS eligible data set (VSAM) was issued to a volume that does not support extended attribute DSCBs. The caller did not specify with EADSCB=OK on the DCBE macro that it supports the extended attribute DSCBs for an EAS eligible data set. Open would have issued an ABEND, MSGIEC142I 113-44 if an attempt was made to open the EAS eligible data set on a volume that supported extended attribute DSCBs.

When this instance occurs for a volume that does not support extended attribute DSCBs, it will be recorded in the tracker as an error message.

### Operating

A number of operator commands are updated to allow for extended addressable volumes in their input or output.

**Using commands with extended address volumes**

The following MVS operator commands are revised to support extended address volumes:

- DEVSERV PATHS operator command response has been enhanced to include the number of cylinders on a device.
- DEVSERV QDASD operator command response has been changed to allow it to display devices with larger number of cylinders.

For more information about these commands, including syntax changes, see z/OS DFSMSdfp Storage Administration and z/OS MVS System Commands.

### Diagnosing problems with extended address volumes

z/OS R10 includes an extensive set of new and changed messages for extended address volumes. Many messages are revised due to changes in the track address, or due to expanding fields to handle the larger volume sizes.
The following ABEND codes indicate that an OPEN operation failed for an extended address volume, because the caller did not indicate support for extended attribute DSCBs (with EADSCB=OK):

- 313-0C
- 113-44
- 113-48

The following return code indicates that a CVAF operation (CVAFDIR, CVAFSEQ, CVAFDSM) failed due to the caller not specifying support for extended attribute DSCBs (with EADSCB=OK):

- Return code 4, CVSTAT STAT082 (X’52’)

The following return code indicates that a CVAFFILT operation failed due to the caller not specifying support for extended attribute DSCBs (with EADSCB=OK):

- Return code 4, CVSTAT STAT086 (X’56’), with the data set name status in the FCL (FCLDSNST) set to a status value of (x’06’). This status code indicates that a data set name is described by a format 8 DSCB and the caller did not specify support for an EAV with the EADSCB=OK keyword.

The following return codes indicate that an OBTAIN macro operation failed due to caller not specifying support for extended attribute DSCBs (EADSCB=OK):

24(X’18’)

The SEEK option was specified. OBTAIN was issued to an extended address volume (EAV) and the caller did not specify with EADSCB=OK that it supports the possible return of a format 8 or format 9 DSCB as described by the target seek address and the data set described by this DSCB is a format 8 or 9 DSCB.

24(X’18’)

The SEARCH option was specified. OBTAIN was issued to an extended address volume (EAV) and the caller did not specify with EADSCB=OK that it supports the possible return of a format-8 DSCB and the target data set is described by a format-8 DSCB.

Related reading: For information about the 313 ABEND, see the description of message IEC144I in z/OS MVS System Messages, Vol 7 (IEB-IEE). For ABENDs 113-44 and 113-48, see the explanation for message IEC142I. For information about the return codes, see z/OS DFSMSdfp Advanced Services.
Chapter 10. Using VSAM RLS secondary lock structures

VSAM record-level sharing (VSAM RLS) is an extension to VSAM that provides direct record-level sharing of VSAM data sets from multiple address spaces across multiple systems. VSAM RLS uses the z/OS coupling facility for cross-system locking, local buffer invalidation, and cross-system data caching. Primary users of VSAM RLS include high-volume applications that access VSAM data sets, such as CICS applications.

In previous releases of z/OS, an installation’s VSAM applications used a single CF lock structure, the IGWLOCK00 lock table, for all record level sharing — regardless of the number of systems that might need this function. This limitation affected system and application availability, and inhibited the isolation of workloads. With only one lock structure to use for a sysplex, an installation’s test workloads could degrade the performance of its production workloads. And, if an application obtained a large number of record locks between ‘commit’ operations, it could fill up the lock structure, causing other applications to fail. Starting in z/OS R10, you can protect against these limitations by using the new VSAM RLS secondary lock structures.

With z/OS V1R10, you can define multiple, secondary lock structures for VSAM RLS workloads, to reduce locking constraints. A new SMS storage class attribute called lock set lets you specify a DFSMS lock structure to be used for VSAM record locks. An installation can define up to 256 lock sets per sysplex. You can place different lock structures in different coupling facilities and balance the usage of the CFs. When an application opens a VSAM data set, RLS processing determines which lock structure to use by checking the storage class defined for the data set. If the storage class specifies a secondary lock structure, RLS processing uses the secondary lock structure for serializing access to records in the data set. Otherwise, RLS processing uses IGWLOCK00 for all record locking (as done in previous releases).

A secondary lock structure connection persists beyond data set closure. Secondary lock structures are disconnected when the SMSVSAM address space is terminated. After the address space is terminated, you can delete the lock structure’s persistent connections using the V SMS,SMSVSAM,FORCEDELETELOCKSTRUCTURE command.

Secondary lock structures are intended for record locks only. Other types of locks continue to use IGWLOCK00, which is still required for the initialization of the SMSVSAM address space. The usual tasks related to the definition, administration, and use of the primary lock structure, IGWLOCK00, do not change with this enhancement.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation” on page 50</td>
<td>• “Planning to use VSAM RLS secondary lock structures” on page 50</td>
</tr>
<tr>
<td></td>
<td>• “Setting up VSAM RLS secondary lock structures” on page 50</td>
</tr>
</tbody>
</table>
### Planning and Installation

This section describes the planning and setup tasks that you would perform to use VSAM RLS secondary lock structures.

### Planning to use VSAM RLS secondary lock structures

**Before you begin:** To understand the concepts of VSAM RLS, read [z/OS DFSMS Using Data Sets](/z/OS DFSMS Using Data Sets) and [z/OS DFSMSdfp Storage Administration](/z/OS DFSMSdfp Storage Administration).

**Software requirements:** Using VSAM RLS secondary lock structures requires z/OS V1R10 and the supplied PTFs.

**Coexistence requirements:** By default, VSAM RLS continues to use the IGWLOCK00 lock table.

If you plan to enable secondary lock structures in a sysplex containing both z/OS V1R10 and pre-V1R10 systems, note that lower releases of z/OS cannot share a VSAM data set with systems that are using secondary lock structures. If an application on a lower-level system is the first to open a VSAM data set, the z/OS V1R10 system reverts to using IGWLOCK00 for all record locks, regardless of whether a lock set is defined for the storage class. PTFs on lower releases prevent the opening of a data set if it is already open using a secondary lock structure on a z/OS V1R10 system.

Perform the following steps to plan for using secondary lock structures:

1. If lock contention has occurred for VSAM RLS workloads at your installation, plan to add additional lock structures and define them to the storage classes for the affected data sets. For example, consider defining separate lock structures for test and production workloads that may have previously been in contention for locks in the IGWLOCK00 lock table.

2. For the new lock structures, decide on their names, size and location. Lock structure names can be 1 to 16 characters in length, starting with an uppercase alphabetic character (A-Z) and including numeric characters, uppercase alphabetic characters, national characters ($, @, #) and the underscore (_). Also, decide on the corresponding lock set names (1-8 characters) to be specified on the storage class and mapped to the lock structure names in the base configuration information.

### Setting up VSAM RLS secondary lock structures

**Before you begin:** To understand the setup tasks required for VSAM RLS, read [z/OS DFSMS Using Data Sets](/z/OS DFSMS Using Data Sets)
Related reading: For information about modifying DFSMS options, see "Initializing SMS through the IGDSMSxx member" in z/OS DFSMSdfp Storage Administration and the description of member IGDSMSxx in z/OS MVS Initialization and Tuning Reference.

To use VSAM RLS secondary lock structures, you must do the following:

- Set up the storage class definition for a lock set or sets.
- Define the lock sets to the base configuration definition, and to the storage classes for the appropriate data sets.

Specifically, perform the following steps to enable secondary lock structures for VSAM RLS processing:

1. Define the names, sizes and locations of the new lock structures in the coupling facility resource management (CFRM) policy, using the administrative data utility IXCMIAPU. For details on defining lock structures to the CFRM, see z/OS MVS Programming: Sysplex Services Guide and z/OS MVS Programming: Sysplex Services Reference.

2. Activate the new CFRM policy, using the SETXCF START,POLICY command. For details, see z/OS MVS Programming: Sysplex Services Reference.

3. Define the same names in the SMS control data set (CDS) through ISMF (the CDS application selection panel) or Naviquest. Specify the name of a lock set and one lock structure per lock set (the same lock structure can be used for multiple lock sets). If no lock set names are specified, RLS processing uses lock structure IGWLOCK00 for all record locking, as in previous releases.

4. Add the lock set name to a storage class through ISMF (the Storage Class Define/Alter panel) or Naviquest. If no name is specified, RLS processing uses IGWLOCK00 for all record locking.

5. Validate and activate the SMS configuration, which causes the new definitions in the Source Control Data Set (SCDS) to take effect. For details, see the chapter on Activating SMS Configurations in z/OS DFSMSdfp Storage Administration.

6. Ensure that VSAM data sets are allocated using those storage classes.

7. Update ACS routines as needed.

Administering

For VSAM RLS secondary lock structures, the administration tasks are as follows.

Monitoring the use of secondary lock structures

To monitor the use of secondary lock structures at your installation, do the following:

1. For statistics on secondary lock structures as well as IGWLOCK00, check subtypes of SMF record Type 42, as follows:
   - Subtype 15 (DFSMS storage Class summary) contains new statistical sections for the secondary lock structure connected for the storage class.
Subtype 16 (Data Set Summary) contains the secondary lock structure name and statistical information for that secondary lock structure.

Subtype 17 (DFSMS lock table summary) contains new statistical sections for each secondary lock structure known to DFSMS.

2. To view detailed status and lock related information for DFSMS lock structures, issue the D SMS,CFLS(ALL | lockstructurename) command.

3. D SMS,SMSVSAM displays system connect and quiesce status for all lock structures connected by the system issuing the command.

4. The D SMS,SMSVSAM, ALL command summarizes the system connect and quiesce status for all lock structures connected in the sysplex.

Operating

You can use operator commands to activate and deactivate secondary lock structures, display active lock structures, and delete lock structures for VSAM RLS.

Before you begin: For more information about using system commands to operate DFSMS, see z/OS MVS System Commands.

Activating and deactivating secondary lock structures

Use the VARY SMS command to activate (enable) or deactivate (quiesce) a secondary lock structure. Specify the lock structure name as shown in the following examples (this command has no effect on IGWLOCK00).

1. Enter the VARY SMS command with the Enable keyword. The Enable keyword allows new access to the VSAM RLS secondary lock structure.

   Example:
   V SMS,CFLS(lockstructurename),Enable
   This example enables the specified lock structure.

2. Enter the VARY SMS command with the Quiesce keyword. The Quiesce keyword stops new access to the VSAM RLS secondary lock structure. Any spheres which open for VSAM RLS access are not allowed to select the specified lock structure name. Existing usage of this secondary lock structure is not affected. The specified lock structure transitions to a fully quiesced state after all of the existing data sets using the lock structure have closed.

   Example:
   V SMS,CFLS(lockstructurename),Quiesce
   This example deactivates the specified lock structure.

Displaying secondary lock structures

You can use the D SMS, CFLS command to display individual VSAM RLS lock structures, or all current VSAM RLS lock structures. If you do not specify a lock structure name or the ALL keyword, IGWLOCK00 data is returned.

1. Enter the DISPLAY SMS, CFLS with a single lock structure name or the ALL keyword:
Example:
D SMS,CFLS(ALL|lockstructurename)

Output:
This example displays status for the specified lock structure or all VSAM RLS lock structures.

Deleting secondary lock structures
To delete secondary lock structures, you can specify a lock structure name on the V SMS,SMSVSAM,FORCEDELETELOCKSTRUCTURE command. Use this command after all data sets using the lock structure have closed, and the SMSVSAM address space(s) which connected the lock structure have been terminated.

Example:
V SMS,SMSVSAM,FORCEDELETELOCKSTRUCTURE(lockstructurename)
Chapter 11. Using the PDSE enhancements

z/OS V1R10 enhances the availability of PDSE caching statistics, providing a new command to display the caching statistics including new details at the data set level. This improved access to the statistics makes it easier to determine if you need to change your current caching specifications.

In z/OS V1R8, new keywords allowed the retention of directory and member data in hiperspace memory cache after the close of a PDSE data set. Retaining this data can improve performance for programs that repeatedly open, read, and close the same members of a PDSE. To monitor PDSE caching statistics, new sections in SMF records 14 and 15 were provided, allowing you to monitor the statistics by OPEN or by storage group. Overall PDSE caching statistics were already available in SMF record 42 subtype 1. For more information about these previous caching enhancements, see Chapter 24, “Using PDSE enhancements,” on page 117.

In z/OS V1R10, you can now monitor statistics by individual data set, and you can obtain the statistics dynamically using the DISPLAY SMS command. You can also display the current utilization of PDSE 64-bit virtual storage buffers. The following section describes the tasks involved.

Note: PDSE member-level caching must be active before any caching statistics can be displayed. The PDSE_HSP_SIZE or PDSE1_HSP_SIZE parameters must be specified in the IGDSMSxx member of SYS1.PARMLIB. For details, see Chapter 24, “Using PDSE enhancements,” on page 117 and z/OS MVS Initialization and Tuning Reference.

Operating

Use the HSPSTATS parameter on the DISPLAY SMS,PDSE or DISPLAY SMS,PDSE1 commands to obtain information related to the use of member caching in a hiperspace, including the size of the hiperspace, the current LRUTIME value, the current LRU CYCLE value, data sets eligible for caching, and data sets which are in cache. These DISPLAY commands generate a scrollable list on the operator's console showing the current caching statistics.

Before you begin: For general information about the DISPLAY command, see z/OS MVS System Commands.

Displaying PDSE 64-bit virtual storage buffer usage

You can display current PDSE 64-bit buffer virtual storage usage, to determine if more virtual storage needs to be added. Use the VSTOR parameter in one of the following commands:

DISPLAY SMS,PDSE,VSTOR

The above example displays 64-bit buffer virtual storage usage for PDSEs in the SMSPDSE address space.

DISPLAY SMS,PDSE1,VSTOR

The above example displays 64-bit buffer virtual storage usage for PDSEs in the SMSPDSE1 address space.
Displaying caching statistics for PDSE data in hiperspace

You can display caching statistics for all SMS-managed and/or non-managed PDSEs, or for individual PDSEs by data set name.

1. To display statistics for all PDSEs that are eligible for hiperspace caching, use one of the following commands:

   **Example:**
   ```
   DISPLAY SMS,PDSE,HSPSTATS
   ```
   This example displays statistics for PDSEs in the SMSPDSE address space that are eligible for hiperspace caching.

   **Example:**
   ```
   DISPLAY SMS,PDSE1,HSPSTATS
   ```
   This example displays statistics for PDSEs in the SMSPDSE1 address space that are eligible for hiperspace caching.

2. To display statistics for all PDSEs under an SMS-managed storage class, use the STORCLAS keyword:

   **Example:**
   ```
   DISPLAY SMS,PDSE,HSPSTATS,STORCLAS(sclass1)
   ```
   This example displays statistics for PDSEs in the SMSPDSE address space under the storage class named `sclass1`.

   **Example:**
   ```
   DISPLAY SMS,PDSE1,HSPSTATS,STORCLAS(sclass1)
   ```
   This example displays statistics for PDSEs in the SMSPDSE1 address space under the storage class named `sclass1`.

3. To display statistics for all non-SMS managed PDSEs, use the UNMANAGED keyword:

   **Example:**
   ```
   DISPLAY SMS,PDSE,HSPSTATS,UNMANAGED
   ```
   This example displays statistics for PDSEs in the SMSPDSE address space that are not SMS-managed.

   **Example:**
   ```
   DISPLAY SMS,PDSE1,HSPSTATS,UNMANAGED
   ```
   This example displays statistics for PDSEs in the SMSPDSE1 address space that are not SMS-managed.

4. To display statistics for a specific PDSE, use the DSN keyword with a full or partially qualified data set name:

   **Example:**
   ```
   DISPLAY SMS,PDSE,HSPSTATS,DSN(user1.pdse.dsn1)
   ```
This example displays statistics for the PDSE named *user1.pdse.dsn1* in the SMSPDSE address space.

---

**Example:**
DISPLAY SMS,PDSE1,HSPSTATS,DSN(*user1.pdse.dsn1*)

This example displays statistics for the PDSE named *user1.pdse.dsn1* in the SMSPDSE1 address space.

---

**Example:**
DISPLAY SMS,PDSE,HSPSTATS,DSN(*user1.*)

An asterisk can be used as a wild card in the data set name, to represent all values for a qualifier level. This example displays statistics for all PDSEs with a high level qualifier of *user1* in the SMSPDSE address space.

---

**Command output example**
The following is an example of a DISPLAY command with the HSPSTATS parameter and its output:

```
D SMS,PDSE1,HSPSTATS,UNMANAGED,dsn(sys1.*),maxdsns(2)
IGW048I PDSE HSPSTATS Start of Report(SMSPDSE1)
HiperSpace Size: 16384 MB
LRUTime : 60 Seconds  LRUcycles: 15 Cycles
BMF Time interval 3600 Seconds
---------data set name-----------------------Cache--Always-DoNot
Elig---Cache--Cache
SYS1.SHASLNKE        Y  N  N
SYS1.SIEALNKE        Y  N  N
PDSE ANALYSIS End of Report(SMSPDSE1)
```

The output fields contain the following values:

**Address Space**
SMSPDSE or SMSPDSE1 as specified in the DISPLAY SMS command.

**HiperSpace Size**
HiperSpace size value in megabytes (MB) or Gigabytes (GB) as defaulted, or as specified in the IGDSMSxx member of SYS1.FARMLIB.

**LRUtime**
LRU algorithm time value in seconds as defaulted or specified in the IGDSMSxx parmlib member.

**LRUCycles**
LRU algorithm cycle value as defaulted or specified in the IGDSMSxx parmlib member.

**BMFTime**
Time in seconds between SMF recording intervals.
Chapter 12. Using the SMS enhancements

In z/OS V1R10, SMS introduces the following enhancements:

1. **Data class space override**: starting in z/OS V1R10, you can specify that space attributes set in the data class should override corresponding attributes set in JCL DD statements or IDCAMS control statements. This new function makes it easier to enforce installation standards for space attributes. In previous releases, JCL specifications always overrode the corresponding data class space attributes.

2. **Data class specification of system-determined blocksize**: This enhancement allows your installation to enforce the implementation of system-determined blocksize, something which occurs in previous releases only when there is no user-specified block size.

3. **New read-only variables for ACS routines**: z/OS V1R10 makes the user-specified secondary space quantity available to the ACS routines, using a pair of new read-only variables.

4. **Message enhancements**: certain important messages are now written to the job log and hardcopy log to help ensure that they are not missed, and reach the attention of the storage administrator.

**Note**: the two new parameters in the data class, Override Space and System Determined Blocksize, are independent of one another.

**Activating the Data Class Space Override**

The ISMF Data Class Define and Data Class Alter panels have a new parameter (Override Space) that can be set to Yes or No. The default value is No, which causes user-defined space values to override corresponding values in the data class, as they do in previous releases of z/OS. A value of Yes indicates that space values in the data class override the following user-specified space values:

- **Overridden JCL space subparameters** (including dynamic allocations such as TSO ALLOCATE):
  - Space type (CYL, TRK, Block length, or record length plus AVGREC)
  - Primary Quantity
  - Secondary Quantity
  - Directory blocks.

- **Overridden IDCAMS DEFINE CLUSTER space parameters**:
  - Cylinders (Primary, Secondary)
  - Tracks (Primary, secondary)
  - Kilobytes (Primary, Secondary)
  - Megabytes (Primary, Secondary)
  - Records (Primary, Secondary)
  - Controlinterval (CI-size Data)
  - Freespace (CI-percent CA-percent).

The space information in the data class must be all-inclusive, otherwise jobs may fail or produce unexpected results. Primary space, secondary space, and allocation units are all determined either from the JCL, or from the data class.
If you set Override Space to Yes in the data class, then you must explicitly specify the following parameters (set them to a valid non-blank value): Avgrec, Avg Value, Primary, and Secondary. If Recorg is specified, then Clsize Data must be explicitly specified. If Recorg is specified and set to KS, then %Freespace CI and %Freespace CA must also be explicitly specified (set to a valid non-blank value). Note that ‘0’ is a valid non-blank value for any of these fields.

For more details on specifying the data class override value, see the section on Defining Record and Space Attributes for Data Class in z/OS DFSMSdfp Storage Administration.

**Specifying a system-determined block size using data class**

System Determined Blocksize is a new parameter in the data class, with two possible values (YES or NO). The default is ‘NO’. With the ‘NO’ value in effect, there is no change in the block size processing used in previous releases – block sizes are determined by the system only when the user-specified block size is zero (no block size was specified). When set to ‘YES’, system-determined block size processing occurs even if a user-specified block size exists.

Note: This function does not override a blocksize value that is set by a program. It overrides a BLKSIZE value set on the DD statement or dynamic allocation.

For more details on specifying the System Determined Block size value in the data class, see the section on Defining Data Classes in z/OS DFSMSdfp Storage Administration.

**Using the new read-only variables for ACS routines**

SMS has added two new read-only ACS variables. In combination, they represent the secondary space quantity requested by the user either on the JCL or by means of IDCAMS control statements. These two new variables are:

- &SPACE_TYPE – specifies the allocation unit (from JCL or IDCAMS control statements) to be used in conjunction with the secondary space allocation amount. Valid values, other than blank which represents ‘unspecified’, are TRK, CYL, K, M, U, and BLK.
- &SECOND_QTY – represents the SECONDARY allocation Quantity (from JCL or IDCAMS control statements). It has meaning only in conjunction with the &SPACE_TYPE variable.

These two variables allow the data class ACS routine to make the appropriate data class assignment. Users need to modify their existing ACS routines ONLY if they wish to use these two variables.

You can examine the new read-only variables in your data class selection ACS routines and use the information to influence the data class to be selected. For example, if the user is requesting a very large secondary quantity, the data class ACS routines may choose to override this by assigning a data class that has Override Space and specific space values specified. This gives you a level of control over the user-specified secondary quantity that was not previously available.

For more information, see the section on Writing ACS Routines in z/OS DFSMSdfp Storage Administration.
Using the message enhancements

Starting with z/OS V1R10, certain SMS messages are now written out to the job log and to the hardcopy log. This change is made to ensure that these important messages get to the attention of the storage administrator. The messages include ones related to Delete and Rename processing, and messages pertaining to storage constraints which need to be addressed to prevent storage related failures from occurring in the near future. The constraint-relief messages now written to the job log and hardcopy log include the following:

- IGD17286I
- IGD17287I
- IGD17288I
- IGD17289I
- IGD17291I
- IGD17292I.

Note: The Delete and Rename processing messages now written to these logs are too numerous to list here.
Chapter 13. Using the object access method enhancements

z/OS DFSMS V1R10 provides the following enhancements to the object access method (OAM):

- Increases the maximum size for objects that OAM can accept and manage. The previous maximum object size of 256 MB is increased to 2000 MB (2,097,152,000 bytes) for the DASD level of the OAM hierarchy.
- The OSREQ API is extended to include the new store-related functions STOREBEG, STOREPRT, and STOREEND, which facilitate the storing of objects greater than 256 MB and up to 2000 MB in size, by allowing for the object to be stored in individual parts.
- The MODIFY OAM,UPDATE,VOLUME command allows an individual volume’s full status to be marked as permanently full. In previous releases, tape and optical volumes that were not filled to their capacity could be artificially marked as full by using the MODIFY OAM,UPDATE command to update FULL to ‘Y’. (Full volumes are eligible for recycle and expiration processing, and are not considered as candidates for write requests.) However, during OAM initialization, that temporary setting could be undone for tape volumes as the amount of free space is recalculated for the volumes. The MODIFY OAM,UPDATE,VOLUME command now lets you mark an individual tape or optical volume’s full status to permanently full by updating FULL to ‘P’. OAM initialization will not reset the full status of any volume that has been marked permanently full.
- The MODIFY OAM,STOP,OSMC command is updated with the FORCE keyword to provide a hard stop function, allowing the operator to stop all OSMC processing faster.
- A new optional keyword is added to the IEFSSNx PARMLIB member to indicate whether or not OAM is to retrieve backup volume information when processing the OSREQ QUERY command.
- The new value LOST is added to the list of valid reasons for the MODIFY OAM,START,AB and MODIFY OAM,STOP,AB operator commands. The new LOST keyword value allows automatic retrieval of the backup copy of an object when the primary copy has been marked ‘lost’ or ‘not-defined.’
- Automatic access to backup can now be configured in the CBROAMxx PARMLIB member. In previous releases, automatic access to backup could only be turned on and off using operator commands. These commands had to be issued every time the system was IPLed. Configuring automatic access to backup facilities in the CBROAMxx PARMLIB member alleviates the need to repeatedly enter the same operator commands and the possibility of forgetting to issue one or more of the commands.
- A new ONLYIF keyword in the CBROAMxx PARMLIB member allows statements in the PARMLIB member to be associated to a specific system. This new keyword allows an installation to share a single CBROAMxx PARMLIB member across all the systems in a sysplex to improve usability. Prior to this release, OAMplex installations were required to have a separate CBROAMxx member for each system in the OAMplex, to accommodate system-unique statements such as OAMXCF OAMMEMBERNAME(name). Additionally, in a mixed-level system environment, some statement keywords introduced in upper level systems might be invalid (and not tolerated) on lower level systems. The ONLYIF keyword allows these system specific statements and keywords to coexist in a single CBROAMxx PARMLIB member.
The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“System Programming”</td>
<td>• “Specifying object sizes up to 2000 MB”</td>
</tr>
<tr>
<td></td>
<td>• “Specifying whether to retrieve backup volume information on the OSREQ QUERY command” on page 65</td>
</tr>
<tr>
<td></td>
<td>• “Configuring automatic access to backup for OAM” on page 65</td>
</tr>
<tr>
<td></td>
<td>• “Limiting CBROAMxx statements to one system, or all systems in an OAMplex” on page 65</td>
</tr>
<tr>
<td>“Administering” on page 66</td>
<td>• “DB2 administration” on page 66</td>
</tr>
<tr>
<td>“Application Programming”</td>
<td>• “Specifying storage for large objects in parts” on page 67</td>
</tr>
<tr>
<td>“Operating” on page 67</td>
<td>• “Marking tape and optical volumes as permanently full” on page 67</td>
</tr>
<tr>
<td></td>
<td>• “Displaying the new automatic access to backup settings” on page 68</td>
</tr>
<tr>
<td></td>
<td>• “Retrieving backup copies of objects on volumes marked lost or not defined” on page 68</td>
</tr>
<tr>
<td></td>
<td>• “Stopping all OSMC processing” on page 68</td>
</tr>
</tbody>
</table>

**System Programming**

After installing z/OS DFSMS V1R10, and completing any migration actions, you need to complete the following system programming tasks to take advantage of the new functions.

For information about required migration actions, see [z/OS Migration](#).

**Specifying object sizes up to 2000 MB**

To implement the increased OAM object size, a system programmer must take the following steps:

In the IEFSSNxx PARMLIB member, update the MOS= keyword on the OAM1 entry to specify the maximum object size that can be stored through the OSREQ programming interface. The MOS= keyword now allows you to specify object sizes greater than 256 MB and up to 2000 MB (2,097,152,000 bytes).

Update the LOB= keyword on the OAM1 entry in the IEFSSNxx to specify partial (P) if only some storage groups have the required DB2 configuration for objects greater than 256 MB in size, or to specify all (A) if all storage groups have the required DB2 configuration for these objects.

For details, see the updated OAM syntax and parameters for IEFSSNxx in [z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support](#).
If either of the above updates are made to IEFSSNxx, it will not take effect until the next IPL.

**Specifying whether to retrieve backup volume information on the OSREQ QUERY command**

Use the new optional keyword QB= in the IEFSSNxx PARMLIB member to indicate whether or not OAM is to retrieve backup volume information when processing the OSREQ QUERY command. This specification is at the global level and pertains to all OSREQ QUERY processing.

- QB=Y indicates that OSREQ QUERY requests result in a call into the OAM address space for each backup copy. The OSREQ QUERY will return a complete backup retrieval order key for each backup copy. If a backup copy does not exist, then the OAM address space will not be called and the backup retrieval order key will contain binary zeros. This is the default.
- QB=N indicates that OSREQ QUERY requests will NOT result in a call into the OAM address space for each backup copy. The backup retrieval order key will contain binary zeros for each backup copy regardless if the backup copy exists or not.

For details, see the updated OAM syntax and parameters for IEFSSNxx in z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

**Configuring automatic access to backup for OAM**

In order to configure automatic access to backup for OAM using PARMLIB, you must code new keywords in a CBROAMxx member of SYS1.PARMLIB. These keywords are added to the SETOPT statement:

- ABUNREAD
- ABOFFLINE
- ABNOTOPER
- ABDB2ERROR
- ABLOST
- ABALL

If one of the automatic access to backup reasons is not specified with an associated SETOPT automatic access to backup keyword, then OAM defaults to not setting automatic access to backup on for that specific reason.

If SETOPT automatic access to backup keywords are added to the CBROAMxx member of PARMLIB and the system is backed down to a previous level of DFSMS, the changes to PARMLIB must also be removed or OAM address space initialization will fail when the new SETOPT keywords are encountered.

For details on coding the new SETOPT keywords, see z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

**Limiting CBROAMxx statements to one system, or all systems in an OAMplex**

Use the new ONLYIF keyword to associate statements in the CBROAMxx PARMLIB member to a specific system or to all systems in an OAMplex. The scope of the ONLYIF statement is in effect until the next ONLYIF is encountered.
To specify that a statement applies only to System xxxx, code the following before the statement:
_______________________________________________________________

Example:
ONLYIF SYSNAME(xxxx) ...

This example associates the following CBROAMxx statements with system xxxx only.
_______________________________________________________________

SYSNAME specifies the name of the system that the following set of statements are to be processed on. Valid values are 1-8 character system name or the reserved string "ALL". If a system_name is specified then the following statement(s) will only be processed if the system OAM is initializing on has a matching system name. The system name is defined via the SYSNAME parameter in the IEASYMxx or IEASYSxx PARMLIB members. If "ALL" is specified then the statement(s) will be processed on all systems. If the SYSNAME keyword is not specified the default value is "ALL".

For more information, see z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

Administering

This section describes tasks that DB2 administrators must perform to enable the support for increased OAM object sizes.

DB2 administration

To enable the larger OAM object sizes on DB2, do the following:

- Create the necessary OAM DB2 infrastructure if you have not already done so, or review the existing OAM DB2 infrastructure. The OAM DB2 infrastructure introduced in z/OS DFSMS 1.8 includes the OAM DB2 LOB database table definitions and the associated DB2 view (V_OSM_LOB_BASE_TBL) which are now required for each storage group that will contain these objects. Sample job CBRILOB, provided in SAMPLIB, will create the necessary LOB storage structures.
- Modify user LOB Value Storage (LOBVALA) and System LOB Value Storage (LOBVALS) fields from DB2 installation panel DSNTIP7. Sizes Panel 2, to establish proper limits for the amount of storage that can be used for storing LOB values.
- Required migration step: Update and execute SAMPLIB job CBRPBIND to create the packages with member CBRIEDBS for the CBRIDBS plan.
- Required migration step: Update and execute SAMPLIB job CBRABIND and/or CBRIBIND to include CBRIEDBS as a package for the CBRIDBS plan.

Note that the z/OS DFSMS 1.8 DB2 infrastructure is available in sample job CBRILOB. CBRILOB will create the LOB storage structures required for OAM to exploit OAM Large Object Phase 2 support. This job can be used to delete existing LOB storage structures as well as create LOB storage structures, which include:

- LOB tablespaces
Base tables
Base table views
Auxiliary tables
LOB indexes

For more information about these administration tasks, see the KB Tracking section in z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

Application Programming

After activating the increased maximum object size as described in “Specifying object sizes up to 2000 MB” on page 64, you can use new functions in the OSREQ API to facilitate the storing of objects greater than 256 MB and up to 2000 MB in size, by allowing for the object to be stored in individual parts.

Specifying storage for large objects in parts

You can use the following new OSREQ API functions to store large objects in separate parts.

- STOREBEG – begin a store operation for an object whose total size is greater than 256 MB
- STOREPRT – store the next sequential contiguous part of an object whose total size is greater than 256 MB
- STOREEND – end the storage of an object whose total size is greater than 256 MB either to complete the storage of the object or to effectively cancel the storage of the object.

Use these new functions in a sequence, starting with the OSREQ STOREBEG to initiate the store operation and provide much of the information that would be provided on an OSREQ STORE. Then code one or more OSREQ STOREPRT function invocations to provide each individual part of the object from the beginning of the object to the end contiguously and in sequence. Finally, code a OSREQ STOREEND function invocation to complete the storage of the object after all of the individual parts have been provided to OAM.

For more information about how to code and use these new functions, see z/OS DFSMS OAM Application Programmer’s Reference.

Operating

You can use operator commands to perform many of the new functions that OAM provides in z/OS V1R10.

Before you begin: For more information about using the system commands, see z/OS MVS System Commands.

Marking tape and optical volumes as permanently full

You can use the MODIFY OAM,UPDATE,VOLUME command to mark tape and optical volumes as permanently full, by specifying a value of "P" for the FULL field of the DB2 Tape Volume Table and the DB2 Volume Table (optical).

1. Specify UPDATE,VOLUME,volser, with the FULL keyword set to P as shown in the following example:
Example:
F OAM,UPDATE,VOLUME,volser,FULL,P
This example marks the volume specified by volser as permanently full, therefore its full status is not reevaluated during subsequent OAM initializations and is retained across OAM initializations.

Displaying the new automatic access to backup settings
The F OAM,DISPLAY command shows the values of the new automatic access to backup keywords in the SETOPT statement in CBROAMxx.
1. Specify SETOPT with a scope of "ALL".

Example:
F OAM,D,SETOPT,ALL
The output from this example includes the following keywords and their specified values if any in CBROAMxx: ABUNREAD, ABOFFLINE, ABNOTOPER, ABDB2ERROR, ABLost, and ABALL.

Retrieving backup copies of objects on volumes marked lost or not defined
The command for automatic access to backup copies is expanded to include a new reason value of LOST, for objects that are on volumes which are marked as lost or not defined. If this value is specified, when a retrieve for an object is attempted and the optical or tape volume on which the object resides is marked lost or not-defined, the backup copy of the object is retrieved.
1. Specify LOST on the MODIFY OAM,START,AB command.

Example:
F OAM,START,AB,LOST,BACKUP1 | BACKUP2
This command specifies that backup copies objects on volumes marked lost or not defined be retrieved when retrieval is attempted.

Stopping all OSMC processing
The MODIFY OAM,STOP,OSMC command is updated to allow the operator to stop all OSMC processing quickly, using the new FORCE keyword. Previously, the STOP,OSMC only canceled all unscheduled work. Now, with the FORCE keyword, OAM cancels all scheduled and unscheduled work. Work currently in progress must complete before OSMC will terminate.
1. To stop all OSMC processing more quickly, specify the FORCE keyword to the MODIFY OAM,STOP,OSMC command.

Example:
F OAM,STOP,OSMC,FORCE
Chapter 14. Using the DFSMSHsm and DFSMSdss enhancements

In z/OS V1R10, DFSMSHsm and DFSMSdss are enhanced with new backup functions and other improvements.

- **NEWNAME keyword added to the BACKDS command**

  The DFSMSHsm (H)BACKDS command allows users to create a backup version of a specified data set. This data set backup function is enhanced in z/OS V1R10 to allow users to assign a new data set name to a backup version, using the NEWNAME keyword. New DATE and TIME keywords are also added to the (H)BACKDS command. These keywords allow you to assign a specific backup date and time to the backup copy.

  The NEWNAME and associated keywords are also added to the ARCHBACK macro, to specify a new data set name for a backup version of a data set. TIME and DATE keywords are also added to the (H)BDELETE command, to allow deletion of a backup version that was created with the NEWNAME keyword.

  The output for the (H)LIST DATASET BCDS/BOTH and LIST LEVEL BCDS/BOTH commands indicate whether the NEWNAME keyword was specified when the backup version was created.

- **NEWNAMEUNCONDITIONAL keyword added to the DFSMSdss DUMP command**

  Corresponding new name support is also added to the DFSMSdss DUMP command, using a NEWNAMEUNCONDITIONAL keyword. This new keyword (abbreviated NEWNAMEU) lets you assign new names to data sets that are being dumped. In previous releases, a DFSMSdss user could only rename a data set during COPY DATASET or RESTORE DATASET processing. DUMP NEWNAMEU might be useful to installations that need to track their inventory of backup data sets that are dumped, especially in cases in which the installation keeps its dumped data sets cataloged to avoid name contention between the data sets that are backed up (dumped) and the data sets that are active on their system.

  The NEWNAMEUNCONDITIONAL keyword can only be used when invoking DFSMSdss through the Application Programming Interface (API).

- **Control data set backup enhancements**

  When backing up DFSMSHsm control data sets (CDSes), DFSMSHsm must first quiesce all activity to the CDSes by acquiring an exclusive enqueue on ARCGPA/ARCCAT, and might have to wait until any other functions that have serialized the CDSes are completed. In z/OS V1R10, DFSMSHsm now uses cross-system coupling facility (XCF) services to have any host performing CDS backup communicate to all other hosts in an HSMplex that the resources needed to serialize the CDSes for backup must be temporarily released. The goal of this enhancement is to reduce the time that CDS backup must wait for the necessary resources to begin processing, allowing CDS backup to start more quickly and reducing the likelihood that the CDS backup function will appear to be hung when it actually is waiting for a long running function to release the ARCGPA/ARCCAT resource.

  Most long-running functions currently use periodic checks to release the ARCGPA/ARCCAT resource based on interval settings set internally in DFSMSHsm or settings patchable by the system programmer. If so, the functions temporarily release the resource and obtain it again in an effort to let CDS
backup begin. In a multi-host environment, long running functions must rely on intervals that are either set by the system programmer or defaulted by best estimate. With this new enhancement, when XCF is available, the need to experiment to define the right intervals for long-running functions to release ARCGPA/ARCCAT, or to contact IBM support is greatly diminished.

Note: if you have previously patched interval values or enabled support, you should leave the existing settings as they are. When XCF services are unavailable, DFSMSshm continues to use the pre-V1R10 methods of periodically releasing ARCGPA/ARCCAT.

For more information about the available patches and default settings for control data set backup, see z/OS DFSMSshm Implementation and Customization Guide.

- **Support for the DFSMSrmm report generator**

  In this release, the DFSMSrmm report generator provides an alternative method of gathering information about DFSMSshm activities from SMF records. Note: the DFSMSrmm report generator is distinct from the DFSMSshm REPORT command. The DFSMSrmm report generator supports a list of up to five assembler macros to map the data in records to be used for reporting. The macros are assembled and the assembler listing used to extract field information. The offset for each field, its characteristics and length are saved to be used for selection by the dialog user.

  To support the DFSMSrmm report generator, DFSMSshm has shipped new sample reports for DCOLLECT records and FSR records in SYS1.SAMPLIB, with members named ARCGxxxx.

  Starting in z/OS R10, the DFSMSrmm report generator lets you specify for each macro, one or more keywords with values to be used with the macro name at assembly time.

  The DFSMSshm macro ARCUTILP is updated to provide new keyword options so that only a single type of record can be mapped, to simplify its use under the DFSMSrmm report generator:

  `ARCUTILP IDCOUT= YES/NO, TYPE= ALL/M/B/C/T`

  Before using any DFSMSShm SMF records containing FSR and WWFSR records in the DFSMSrmm report generator, you must convert them to FSR2 and WFSR2 format, or you must request that the generated reporting JCL create the reporting data. To support the record reformatting, JCL skeletons are provided, which are used by the DFSMSrmm Report Generator, or you can use the JCL.

  The two new skeletons are:
  - ARCGFSRC, which converts ARCFSR records to ARCFSR2 records.
  - ARCGWFSC, which converts ARCWFSR records to ARCWFSR2 records.

  For more information about the DFSMSrmm report generator, see z/OS DFSMSrmm Reporting. For more information about using the DFSMSrmm report generator to create customized DFSMSShm reports, see z/OS DFSMSshm Storage Administration.

- **Using the SETSYS-specified unit name in tape copies generated by duplex failures**

  In this release, DFSMSShm is updated to pass the unit name associated with the appropriate SETSYS parameter (such as MIGUNITNAME for migration) rather than the corresponding generic name (such as 3590-1) when a tape copy is created for a duplexing failure. This change affects DFSMSShm’s backup, migration, and recycle functions, but not the ABARS function.

- **Using new and enhanced DFSMSdss commands**

  DFSMSdss provides new functions with the following new and enhanced commands:
- New DFSMSdss CONSOLIDATE command – performs both extent consolidation and reduction for data sets that occupy multiple extents on a single volume, and lets you specify which data sets are to be processed. This command tries to reduce the number of extents of a data set as much as possible even when the entire data set cannot be reduced to one extent.

- DFSMSdss DEFRAG command – a new version of this command is added to support 28-bit cylinder addressing for volumes larger than 65,520 cylinders and to provide performance improvement for some workloads. The new version of DEFRAG drops the CONSOLIDATE parameter, which is replaced by the new CONSOLIDATE command. You can still invoke the old version of DEFRAG by specifying the VERSION1 parameter. Other new parameters of the DEFRAG command are:
  - DEBUG(TRACE)
  - MMOVPCT(n,p)
  - MAXTIME(nummins)

- DFSMSdss COPY and DUMP commands – The existing CONCURRENT keyword is enhanced for the DFSMSdss COPY and DUMP commands to allow you to request concurrent copy (or virtual concurrent copy) for your copy and dump operations, and whether DFSMSdss should retry with different methods.

For details on these new and updated DFSMSdss commands, see z/OS DFSMSdfp Storage Administration.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation”</td>
<td>• “Planning to use the DFSMS backup enhancements”</td>
</tr>
<tr>
<td>“Administering” on page 72</td>
<td>• “Specifying a new name for DFSMSdss dumped data sets” on page 72</td>
</tr>
<tr>
<td>“Programming” on page 72</td>
<td>• “Specifying a new name on the (H)BACKDS command” on page 73</td>
</tr>
<tr>
<td>“Operating” on page 73</td>
<td></td>
</tr>
</tbody>
</table>

### Planning and Installation

This section describes the planning and installation tasks that you would perform for the DFSMShsm usability enhancements.

For CDS backup, if you are not running in a sysplex environment and therefore not using the CDS backup enhancement, you will need to make alternative decisions about how to tune the intervals and about which functions should not run concurrently with CDS backup. See z/OS DFSMShsm Implementation and Customization Guide for new information to help make these decisions.

### Planning to use the DFSMS backup enhancements

**Before you begin:** Be familiar with the following topics:
• For information about the DFSMShsm (H)BACKDS and (H)BDELETE commands, and the DFSMSdss DUMP command, see z/OS DFSMSdfp Storage Administration.
• For information about CDS backup, see z/OS DFSMSshm Storage Administration

**Software requirements:** The backup enhancements require z/OS V1R10.

**Coexistence requirements:** The CDS backup enhancement requires pre-V1R10 DFSMSshm hosts to be updated to ignore XCF service notifications from V1R10 hosts that are performing CDS backup. The pre-V1R10 methods of releasing the ARCGPA/ARCCAT resource will not interfere with the new methods employed in V1R10.

**Administering**

To specify a data set name for backup data sets (making a backup of *dsname* look like a backup for data set *newdsname*), you can use either of the following methods:

• (H)BACKDS command

• ARCHBACK macro.

**Related reading:**

• For information about using the (H)BACKDS command, see z/OS DFSMSdfp Storage Administration

• For information about using the ARCHBACK macro, see z/OS DFSMSshm Managing Your Own Data

**Programming**

**Specifying a new name for DFSMSdss dumped data sets**

You can use the NEWNAMEUNCONDITIONAL (NEWNAMEU) keyword on the DFSMSdss DUMP command to assign new names to data sets being dumped. Specify the old data set name and the new name on the keyword as shown in the example below.

**Example:**

```
DUMP DATASET(INCLUDE(datasetname)) -
   OUTDDNAME(ddn) CANCELERROR OPTIMIZE(n) -
   NEWNAMEU(datasetname,newdatasetname)
```

Optionally, you can specify a data set prefix to be used in place of the current first-level qualifier of the data set.

The NEWNAMEU keyword is available to user-supplied programs or installation exit routines using the DFSMSdss DUMP command through the DFSMSdss application programming interface (API). It is not invocable through operator commands.

**Related reading:** For information about using NEWNAMEU on the DFSMSdss DUMP command, see z/OS DFSMSdss Storage Administration and z/OS DFSMSdfp Storage Administration
Operating

Specifying a new name on the (H)BACKDS command

You can specify a new name for back-up data sets using the (H)BACKDS NEWNAME operator command:

Example:

```
BACKDS dsname NEWNAME(newname)
```

New DATE and TIME keywords are also added to the (H)BACKDS command. You can use these to assign a specified backup date and time to the backup copy created via the (H)BACKDS NEWNAME command.

Example:

```
BACKDS dsname NEWNAME(newname) DATE(yyyy/mm/dd) TIME(hhmms)
```

The output for the (H)LIST DATASET BCDS/BOTH and LIST LEVEL BCDS/BOTH commands indicates whether the NEWNAME keyword was specified when the backup version was created.
Chapter 15. Using the DFSMSrmm enhancements

The functional enhancements available with z/OS V1R10 DFSMSrmm provide you with these benefits:

- **DFSMShsm report generator**
  The DFSMSrmm Report Generator is updated to support keywords for assembler macros from which report types are derived and also to add new built-in data extract steps. This is in support of the new SMF record types from DFSMSrmm and for DFSMSHsm and DCOLLECT reporting. New report types and report samples are provided.

- **DFSMShsm enterprise enablement**
  - **SMI-S Storage Library Profile:** Support is provided for the latest CIM/SMI-S levels; CIM 2.14, and SMI-S 1.2 profile levels. There are also changes to the RMM CIM classes and providers. Also, the DFSMSrmm CIM agent now has an option to register itself using the SMI-S Storage Library profile so that storage management clients (and TPC in particular) can use Service Location Protocol (SLP) to detect the CIM agent and determine the registered profiles that it can support.
  - **Support of the latest CIM Schema:** Support is provided for the latest CIM Schema level - 2.14. This will cause a change in the key processing of the existing CIM classes provided by RMM CIM Agent.
  - **Integration with IRMM:** Support is provided for the integration of IRMM and DFSMSrmm.
  - **DFSMShsm TSO Command Changes for IRMM:** New subcommands are provided for IRMM and other applications on other platforms to be able to maintain similar information to that recorded automatically by DFSMSrmm on z/OS.

- **Optimization and management capabilities**
  - **DELETE Disposition Support for Tape Data Sets:** Support is provided for managing deleted tape data sets. DFSMSrmm checks the normal disposition at CLOSE time, and if this is 'DELETE', a 'deleted' flag is recorded in the data set record. Subsequent use of a data set cannot change this. However, you can use the CHANGÉDATASET command to reset the flag.
  - **CDS Fast Replication:** You can request that DFSMSrmm backup processing use DFSMSdss COPY services instead of using DUMP services. Use of copy services enables the system to exploit the fast replication services of the DASD subsystems, such as Flashcopy V2 data set level flashcopy. Copy services enable an almost instant copy of the DFSMSrmm CDS to be created, that is ready to use, without the need to first restore a backup copy.
  - **Forward Recovery of CDS from SMF Records:** To forward recover the CDS from SMF records, you must be using the standard IBM SMF record type for DFSMSrmm. You can now avoid DFSMSrmm SMF records requiring record types from the user-written range. DFSMSrmm now uses the IBM assigned record type 42, subtypes 22 and 23.
  - **Managing Retention Exceptions During Inventory Management:** New parmlib options are added that can be used to ensure that data is being retained as expected and only released by DFSMSrmm processing if the numbers or percentages of volumes are within policy limits. EDGHSKP VRSEL and EXPROC processing counts the numbers of volumes that are:
    - VRS retained; initial count and dropped from retention
Newly assigned since the last VRSEL run; initial count and number newly VRS retained  
- EXPDT retained; initial count and dropped from EXPDT retention.

DFSMSrmm applies the new parmlib options to determine the action to be taken.

**DFSMSrmm ease of use enhancements**

- **Library Partitioning:** DFSMSrmm provides new parmlib commands that provide more flexibility over partitioning and controlling use of tape volumes. Through use of the PARTITION and OPENRULE parmlib commands, you can now simplify maintenance of parmlib members as your libraries and volume ranges change.

- **Report Extract Tailoring:** You can now be selective about which records are written to the report extract file. You can use the RPTEXT command in the EDGHSKP SYSIN file to select the new extract processing. The default processing for the XREPTEXT DD statement is changed so that only extended records are created.

- **Volume Replacement Policies:** DFSMSrmm supports policies for volume replacement. You define them in the MEDINF commands in parmlib. This enables you to define different replacement policies based on media type and recording format. The replacement policies are implemented for all types of volume other than logical volumes, and for both private and scratch volumes, but not for those which are pending release. Until you define one or more MEDINF commands in parmlib with the REPLACE operand DFSMSrmm processing is unchanged; a hard-coded value of PERM(1) is used.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning to use DFSMSrmm enhancements”</td>
<td>“Report generator enhancements”</td>
</tr>
<tr>
<td>Operating</td>
<td>“DFSMSrmm Enterprise enablement” on page 77</td>
</tr>
<tr>
<td>Operating</td>
<td>“DFSMSrmm optimization and management capabilities” on page 78</td>
</tr>
<tr>
<td>Operating</td>
<td>“DFSMSrmm ease of use enhancements” on page 79</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>“Operating” on page 80</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>“Diagnosing” on page 80</td>
</tr>
</tbody>
</table>

**Planning and Installation**

This section describes the planning and installation tasks that you would perform to use the DFSMSrmm enhancements.

**Planning to use DFSMSrmm enhancements**

**Report generator enhancements**

**Before you begin:** Be familiar with information about the following topic:

- DFSMSrmm report generator; see z/OS DFSMSrmm Reporting
The DFSMSrmm report generator is updated to support keywords for assembler macros in the report type definitions. When you add a new report type to your library using the Add a Report Type panel, or change a report type using the Change a Report Type panel, you can optionally specify keywords for each of the macro names you enter. The macro keywords determine which subset of the possible range of mappings are to be used for the specified report type.

The report generator is also updated with new line commands to aid with problem diagnosis when assembling macros. The L line command lets you view the assembler listing created by the report generator dialog assembling the macros, and their keywords if any. The M line command lets you view the macros specified for the report type. You can use the M line command to review the entire macro and determine which keywords and values might be valid.

The report generator cannot easily process DFSMSHsm’s ARCFSR records or ARCWFSR records, so DFSMSHsm provides two new jobs to convert those records to ARCFSR2 and ARCWFSR2 respectively. The new jobs are provided as skeletons in SYS1.DGTSLIB. The ARCGFSRC job converts ARCFSR records to ARCFSR2 records, and the ARCGWFSC job converts ARCWFSR records to ARCWFSR2 records. When you use the report generator to report on DFSMSHsm records and request that the report data is created, these skeletons are automatically included in the generated JCL. If you want to report independently of the report generator, you will need to use this skeleton JCL to reformat the records.

DFSMSHsm also provides a number of new report types and sample reports, named ARCGxxxx in SYS1.SAMPLIB. For a list of the new sample reports, see z/OS DFSMSHsm Storage Administration.

Any new report type or report definitions created on z/OS V1R10 which include macro keyword information can be used by a lower level release; however, the macro keywords are ignored and are removed if the report type or report definition is changed or updated. The original input report type or report definition is unaffected and can be reused later on a supporting release to exploit the macro keywords.

IBM recommends that you use z/OS V1R10 or later release to update/customize report types and reports which require macro keywords to be specified. Once the report JCL is generated you can run that JCL on any supported release.

**DFSMSrmm Enterprise enablement**

In z/OS V1R10, DFSMSrmm supports the latest CIM/SMI-S levels (CIM 2.14 and SMI-S 1.2 profile levels), and provides updates to the DFSMSrmm CIM classes and providers. The DFSMSrmm CIM agent now has an option to register itself using the SMI-S Storage Library profile so that storage management clients (and IBM Tivoli Productivity Center in particular) can use Service Location Protocol (SLP) to detect the CIM agent and determine which registered profiles it can support. For details, see the section on setting up the DFSMSrmm CIM Provider in z/OS DFSMSrmm Implementation and Customization Guide.

There are required migration steps that you must take if you want to continue to use the z/OS V1R9 level of the CIM provider under R10, and other steps you must take to remove the V1R9 level of the CIM provider before you can use the V1R10 level. For details, see z/OS Migration. If you are migrating to z/OS V1R10 from z/OS V1R8, first follow the documented steps for migration to V1R9 before completing the V1R10 registration steps.
Support for the latest CIM Schema level, 2.14, is provided. This has caused a change in the key processing of the existing CIM classes provided by RMM CIM Agent. For details, see the section on setting up the DFSMSrmm CIM Provider in z/OS DFSMSrmm Implementation and Customization Guide.

In support of the integration with IRMM, DFSMSrmm now provides support for the DFSMSrmm web service to run without requiring the complete WebSphere Application Server to be installed. Instead, the web service can be implemented using the Apache Tomcat server or a similar middleware web serving environment.

DFSMSrmm TSO subcommand changes – the following DFSMSrmm TSO subcommands have updates to support IRMM: ADDVOLUME (AV), CHANGEVOLUME (CV), CHANGEDATASET (CD). For details, see z/OS DFSMSrmm Managing and Using Removable Media.

Software requirements: The DFSMSrmm enterprise enablement enhancements require z/OS V1R10 and either the z/OS WebSphere Application Server or the Apache Tomcat server.

**DFSMSrmm optimization and management capabilities**

Starting in z/OS V1R10, DFSMSrmm adds support for managing deleted tape data sets. Tape data sets are subject to VRS processing as specified by your retention and movement policies. To ensure that ‘deleted’ data sets are managed differently, you must create special ‘DELETED’ VRSes, otherwise they are managed by the normal matching VRS. The following DFSMSrmm subcommands have new “Deleted” operands, to specify whether they should apply to deleted data sets or not:

- ADDDASSET
- CHANGEDASSET
- SEARCHDASSET.

A new REXX variable, DLTD, is also added. For details on this and the subcommand changes that support deleted tape data sets, see z/OS DFSMSrmm Managing and Using Removable Media.

Any tape data sets processed on z/OS V1R10 or later release with the ‘deleted’ attribute set are processed unchanged by lower level releases. Any special ‘DELETED’ VRSes created are handled as management class or VRS management value VRSes on lower level releases and not as the special VRS. If you wish to exploit the new support run Inventory Management on z/OS V1R10 or later release.

**CDS Fast replication:** In previous releases, CDS backup was supported using EDGHSKP and EDGBKUP, offering options to use either AMS REPRO or DFSMSdss DUMP. Non-intrusive backup was possible using the DFSMSdss CONCURRENT DUMP option. In z/OS V1R10, CDS backup can now additionally be performed using copy services. DFSMSdss is used with Fast Replication to enable a CDS ‘backup’ to be created. New parameter options are provided to request this method for CDS backup. For details, see the sections on inventory management and maintaining the control data set in z/OS DFSMSrmm Implementation and Customization Guide.

**Forward recovery of the DFSMSrmm CDS from DFSMSrmm audit SMF records:** DFSMSrmm creates SMF records when requested, but can now optionally use the IBM assigned record type 42, instead of using SMF record types from the user-range as in previous releases. SMFAUD and SMFSEC options each use a
record type unless the IBM record type is used, in which case they use subtypes within record type 42; the subtypes used by DFSMSrmm are 22 for audit records and 23 for security records. For EDGAUD processing, the record type is optional. Whether a user-written type is specified or not, EDGAUD also uses the IBM SMF record type. For forward recovery in EDGBKUP there is now sufficient information in the audit SMF records created using record type 42. EDGBKUP reads the SMF records, sorts them into the correct sequence and then processes them as if they were journal records. For more information, see z/OS DFSMSrmm Implementation and Customization Guide and z/OS DFSMSrmm Reporting.

To create audit or security reports for releases before z/OS V1R10 that should include the new IBM SMF type 42 records created by z/OS V1R10, run EDGAUD on z/OS V1R10 or higher. Note that new type of SMF records are only created when you update the DFSMSrmm parmlib options SMFSEC and SMFAUD to use IBM-assigned SMF record types.

Managing retention exceptions during inventory management: New EDGRMMxx OPTION command operands can be used to ensure that data is being retained as expected and only released by DFSMSrmm processing if the numbers or percentages of volumes are within policy limits. The new operands are:

**VRSDROP**
- specifies a maximum number or percentage of existing VRS retained volumes that can be dropped from vital records retention, and the action to be taken by DFSMSrmm

**VRSRETAINT**
- specifies a minimum number or percentage of newly assigned volumes that are to be retained by vital records retention, and the action to be taken by DFSMSrmm.

**EXPDTDROP**
- specifies a maximum number or percentage of existing expiration date retained volumes that can be dropped from retention, and the action to be taken by DFSMSrmm.

For more information about the new parmlib options for retention, see the section on SYS1.PARMLIB member EDGRMMxx in z/OS DFSMSrmm Implementation and Customization Guide

Software requirements: The DFSMSrmm optimization and management capabilities require z/OS V1R10.

DFSMSrmm ease of use enhancements
The PRTRITION and OPENRULE parmlib commands provide more function and greater flexibility and help to minimize administration, including the maintenance of the parmlib members, as your libraries and volume ranges change. Operands on the OPENRULE and PRTRITION commands allow global actions to be set. You can use one or more specific overrides based on volume sets that have different requirements. Use of OPENRULE can help avoid the need to modify JCL and to customize EDGXU100. For more information on these commands, see z/OS DFSMSrmm Implementation and Customization Guide

If you use REJECT commands, you must convert from the use of REJECT commands in order to use the PRTRITION and OPENRULE commands. Each system in the RMMplex uses its own parmlib options to determine the processing required. For releases prior to z/OS V1R10, the REJECT commands are used for both partitioning and for Open/Close/EOV volume use decisions. On z/OS V1R10
and later releases, the REJECT commands are only used if specified and neither PRTITION nor OPENRULE commands are defined. When you define REJECT commands and PRTITION or OPENRULE commands, the REJECT commands cause an DFSMSrmm start-up error and you must reply to EDG0239E to continue the start-up or restart DFSMSrmm with a corrected parmlib. If there are no REJECT commands the PRTITION and OPENRULE command defaults are used. For more information, see the section on converting REJECT commands to PRTITION and OPENRULE commands in z/OS DFSMSrmm Implementation and Customization Guide.

For volume replacement policies, you can set threshold values for volumes to be identified for replacement, by changing the release value from SCRATCH to REPLACE when one or more of the thresholds are exceeded. For details, see the section on volume replacement policies in z/OS DFSMSrmm Implementation and Customization Guide.

With the new report extract tailoring function, you can now be selective about which records are written to the report extract file, and avoid creating extended records if appropriate. Use the new RPTEXT command in the SYSIN file to select which records you want written to the report extract file. The extracted records are written to the XREPTEXT DD if it exists. If XREPTEXT DD does not exist the REPTEXT DD is used. When there is no RPTEXT command in SYSIN, or the RPTEXT command specifies no RECORDS operand, the default processing for the REPTEXT DD statement is the same as in previous releases, but the default processing for the XREPTEXT DD statement is changed so that only extended records are created.

For more information on these and other DFSMSrmm ease of use enhancements, see updates for z/OS V1R10 in z/OS DFSMSrmm Managing and Using Removable Media and z/OS DFSMSrmm Implementation and Customization Guide.

**Software requirements:** The DFSMSrmm ease of use enhancements require z/OS V1R10.

### Operating

To run the DFSMSrmm application programming interface via web services, you must be running WebSphere® Application Server or an equivalent such as Apache Tomcat server. (DFSMSrmm now ships the web service for installation either as an Enterprise Archive (EAR) under WebSphere or as a Web Archive (WAR) under Tomcat). To run the CIM provider, you must be running an xmlCIM compliant product that supports Java, such as the OpenPegasus C++ CIMOM from The Open Group.

**Before you begin:** For additional information, including steps for using the Tomcat server, see z/OS DFSMSrmm Implementation and Customization Guide.

### Diagnosing

For information on diagnosing DFMSrmm problems, see z/OS DFSMSrmm Diagnosis Guide.
Part 3. Using New DFSMS Functions in z/OS V1R9

Chapter 16. Using BSAM and QSAM Enhancements 83
Using long-term page fixing for BSAM data buffers 83
QSAM support for MULTSDN 83
BSAM and QSAM support for MULTACC on Tape 84

Chapter 17. Using VSAM System-Managed Buffering enhancements 87
Administering 87
  Specifying a SMBVSP value for a data class 87

Chapter 18. Using SMS volume selection preference for the same storage facility image 89

Chapter 19. Using the Object Access Method enhancements 91
System Programming 91
  Defining tape sublevel parameters to OAM 91
  Update SMS Storage Class constructs 92
Operating 92
  Modifying the SETOAMxx keywords 93
  Displaying the new OAM tape level settings 93

Chapter 20. Using the DFSMSrmm enhancements 95
Planning and Installation 96
  Planning to use DFSMSrmm enhancements 96
  CIM Provider enhancements 96
  DFSMSrmm Task Management enhancements 97
  DFSMSrmm utility and inventory management 97
  DFSMSrmm control data set serialization 98
  DFSMSrmm usability items 98
Operating 98
Diagnosing 98

Chapter 21. Using the DFSMSHsm enhancements 99
Chapter 16. Using BSAM and QSAM Enhancements

z/OS V1R9 provides the following enhancements to BSAM and QSAM:
- Long-term page fixing for BSAM data buffers
- QSAM support for MULTSDN
- BSAM and QSAM support for MULTACC.

Using long-term page fixing for BSAM data buffers

To improve performance, in z/OS V1R9 BSAM allows certain calling programs to specify that all their BSAM data buffers have been page fixed. This specification frees BSAM from the CPU-time intensive work of fixing and freeing the data buffers itself.

The calling program must be APF-authorized, or be in system key or supervisor state. The format of the data set must be either basic format, large format, PDS, or extended format. Compressed format data sets are not supported.

To specify that the calling program has done its own page fixing, code the following option on the DCBE macro:

FIXED=USER

indicates that the user has page fixed all BSAM data buffers.

A bit in the DCBE control block (DCBEBENEFIX) is set on to indicate that a program can benefit from specifying FIXED=USER on the DCBE macro. In addition, a bit in the Data extent block (DEB2XUPF) is set on to indicate that FIXED=USER is in effect.

To check whether there is any benefit to fix data pages by using the DCBEBENEFIX bit, you must define a DCB OPEN exit routine. To provide a DCB OPEN exit routine, follow the steps for the DCB OPEN Installation Exit in z/OS DFSMS Installation Exits.

Related reading: For details about the FIXED=USER option on the DCBE macro instruction, see z/OS DFSMS Macro Instructions for Data Sets. For details on the related control block fields, see z/OS DFSMSdfp Diagnosis.

QSAM support for MULTSDN

In z/OS V1R9, you can use the MULTSDN parameter of the DCBE macro with QSAM. In previous releases, QSAM ignored the MULTSDN parameter. This new support for MULTSDN allows the system to calculate a more efficient default value for the DCB's BUFNO parameter, and reduces the situations where you need to specify a BUFNO value.

QSAM accepts a MULTSDN value for the following data sets:
- tape data sets
- DASD data sets of the following types:
  - basic format
  - large format
For the supported types of data sets, the system uses MULTSDN to calculate a more efficient value for BUFNO when the following conditions are true:

- The MULTSDN value is not zero.
- DCBBUFNO has a value of zero after completion of the DCB OPEN exit routine
- The data set block size is available.

When MULTSDN is specified, note that the default number of buffers may be less than what would have been derived without MULTSDN, as shown in Table 2.

<table>
<thead>
<tr>
<th>Data Set Type</th>
<th>DCBBUFNO default without MULTSDN</th>
<th>DCBBUFNO default with MULTSDN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDSE Member</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Extended format data set in the compressed format</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>UNIX® file</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Extended format data set (not in the compressed format)</td>
<td>2 * number of stripes * number of blocks per track</td>
<td>MULTSDN * number of stripes * number of blocks per track</td>
</tr>
<tr>
<td>Block size equal to or greater than 32 KB (tape)</td>
<td>2</td>
<td>MULTSDN value</td>
</tr>
<tr>
<td>Block size less than 32 KB (tape)</td>
<td>5</td>
<td>MULTSDN * number of blocks in 64 KB</td>
</tr>
<tr>
<td>IBM 2540 card reader or card punch</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>PS, PDS</td>
<td>5</td>
<td>MULTSDN * number of blocks per track</td>
</tr>
<tr>
<td>Others, including dummy data sets</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>TSO terminal</td>
<td>5</td>
<td>1</td>
</tr>
</tbody>
</table>

For more information, see the description of the MULTSDN parameter of the DCBE macro in Z/OS DFSMS Macro Instructions for Data Sets.

**BSAM and QSAM support for MULTACC on Tape**

In z/OS V1R9, the MULTACC parameter of the DCBE macro is expanded, to optimize performance for tape data sets with BSAM, and to support QSAM with optimized performance for both tape and DASD data sets. The calculations used to optimize performance for BSAM with DASD data sets are also enhanced.

For BSAM in V1R9, if you code a nonzero MULTACC value, OPEN calculates a default number of READ or WRITE requests that you are suggesting the system queue more efficiently. OPEN calculates the number of BLKSIZE-length blocks that can fit within 64 KB, then multiplies that value by the MULTACC value. If the block size exceeds 32 KB, then OPEN uses the MULTACC value without modification (this can happen only if you are using LBI, the large block interface). The system then tries to defer starting I/O requests until you have issued this...
number of READ or WRITE requests for the DCB. BSAM will never queue (defer) more READ or WRITE requests than the NCP value set in OPEN.

For QSAM in V1R9, if you code a nonzero MULTACC value, OPEN calculates a default number of buffers that you are suggesting the system queue more efficiently. OPEN calculates the number of BLKSIZE-length blocks that can fit within 64 KB, then multiplies that value by the MULTACC value. If the block size exceeds 32 KB, then OPEN uses the MULTACC value without modification (this can happen only if you are using LBI, the large block interface). The system then tries to defer starting I/O requests until that number of buffers has been accumulated for the DCB. QSAM will never queue (defer) more buffers than the BUFNO value that is in effect.

IBM recommends setting MULTACC to one half of the MULTSDN value.

If you code a MULTACC value that is too large for the system to use, the system ignores the excess amount. However, the absolute upper limit for MULTACC is 255.

For more information, see the description of the MULTACC parameter of the DCBE macro in [z/OS DFSMS Macro Instructions for Data Sets](https://www.ibm.com).
Chapter 17. Using VSAM System-Managed Buffering enhancements

The JCL AMP parameter's SMBVSP keyword lets you limit the amount of virtual buffer space to acquire for direct optimized processing when opening a data set. Before z/OS V1R9, changing that value required editing the JCL statement, which was not practical when running a batch job or a large set of jobs.

In z/OS V1R9, VSAM provides a simpler, quicker way of modifying the SMBVSP value, by specifying it for a data class using ISMF. The System Managed Buffering field on ISMF's DATA CLASS DEFINE/ALTER panel lets you specify the value in kilobytes or megabytes, which SMB then uses for any data set defined to that data class. With this method, the effect of modifying the SMBVSP keyword is no longer limited to one single job step, and no longer requires editing individual JCL statements.

In addition, a new JCL AMP keyword, MSG=SMBBIAS, lets you request a message which displays the record access bias that is specified on the ACCBIAS keyword or chosen by SMB in the absence of a user selection. The message, IEC161I, is issued for each data set that is opened. The new keyword is optional and defaults to not issue a message; you should avoid the keyword when a large number of data sets are opened in quick succession.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administering</td>
<td>• Specifying a SMBVSP value for a data class</td>
</tr>
</tbody>
</table>

Administering

This section describes the administration tasks you must perform to use SMBVSP with a data class.

Specifying a SMBVSP value for a data class

Before you begin: To understand the concepts of VSAM system-managed buffering, read the topic "Tuning for System-Managed Buffering" in z/OS DFSMS Using Data Sets.

Software requirements: Using VSAM SMBVSP in a data class requires z/OS V1R9 and the supplied PTFs.

Perform the following steps to plan for using the new function:
1. Determine what data sets currently have SMBVSP specified in their JCL, and whether the values need to be modified on a regular basis. Also consider other JCL AMP parameters that are specified for these data sets, and whether they would override corresponding values set in a data class.
2. For data sets whose SMBVSP values need to be changed periodically, group together those data sets whose SMBVSP values can or should stay the same.

3. For such data sets, create a data class or update an existing data class with the appropriate value for system-managed buffering. For example, on the Data Class Define/Alter enter a System Managed Buffering value as shown below:

Panel Utilities Scroll Help

DGTDCC2 DATA CLASS DEFINE/ALTER Page 2 of 4

Command ===>

SCDS Name . . . : Y421252.MYSCDS
Data Class Name : DC1

To DEFINE/ALTER Data Class, Specify:

Data Set Name Type . . . . . (EXT, HFS, LIB, PDS or blank)
If Ext ............... (P=Preferred, R=Required or blank)
Extended Addressability . . N (Y or N)
Record Access Bias . . . . . (S=System, U=User or blank)
Space Constraint Relief . . N (Y or N)
Reduce Space Up To (%) . . . (0 to 99 or blank)
Dynamic Volume Count . . . (1 to 59 or blank)
Compaction . . . . . . . . . (Y, N, T, G or blank)
Spanned / Nonspanned . . . . (S=Spanned, N=Nonspanned or blank)
System Managed Buffering . . . (1K to 2048M or blank)

Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels; Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
Chapter 18. Using SMS volume selection preference for the same storage facility image

The DFSMSdss data set fast replication function requires that all volumes of a data set reside in the same storage facility image (SFI). To take advantage of data set fast replication, SMS volume selection is enhanced to prefer candidate volumes that are in the same SFI, when allocating or extending an SMS-managed multi-volume data set that has point-in-time copy volumes requested. The point-in-time copy volumes are requested when the Accessibility field in the storage class is set to CONTINUOUS or CONTINUOUS PREFERRED.

To make the best use of the fast replication function, ensure that enough volumes are on one SFI to accommodate the likely increase in allocations to volumes in these storage groups and SFIs. SMS will not prefer volumes in the same SFI when no SFIs have a sufficient number of volumes to meet the number of volume required for allocation.

For more information, see the topic "SMS Volume Selection for Data Set Allocation" and information about the Accessibility field in the storage class, in z/OS DFSMSdfp Storage Administration.
Chapter 19. Using the Object Access Method enhancements

z/OS DFSMS V1R9 provides Object Access Method (OAM)'s Object Tape Enhancements Stage 2 – tape sublevels. Two new sublevels are provided for OAM tape support. Previously, the OAM storage hierarchy consisted of three levels: disk, optical, and tape. The addition of the tape sublevels effectively creates four levels:

- Disk
- Optical
- Tape sublevel 1 (TSL1)
- Tape sublevel 2 (TSL2)

In addition to being able to write and read object data directly to and from a given tape sublevel, the new sublevels give you the ability to transition object data within the tape family, for example, from VTS to native tape. Previously, moving data within a storage family required more manual efforts using the MOVEVOL command. Now, OSMC storage group processing can be used to automatically offload object data from one tape sublevel to another. A new OAM Sublevel (OSL) parameter is added to the SMS storage class construct in order to associate a storage class with a given tape sublevel.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;System Programming&quot;</td>
<td>• &quot;Defining tape sublevel parameters to OAM&quot;</td>
</tr>
<tr>
<td></td>
<td>• &quot;Update SMS Storage Class constructs&quot; on page 92</td>
</tr>
<tr>
<td>&quot;Operating&quot; on page 92</td>
<td>• &quot;Modifying the SETOAMxx keywords&quot; on page 93</td>
</tr>
<tr>
<td></td>
<td>• &quot;Displaying the new OAM tape level settings&quot; on page 93</td>
</tr>
</tbody>
</table>

System Programming

After installing z/OS DFSMS V1R9, and completing any migration actions, you need to complete the following system programming tasks to take advantage of the new tape sublevels.

For information about required migration actions, see z/OS Migration.

Defining tape sublevel parameters to OAM

In order to define tape sublevel parameters to OAM, you must code or modify certain keywords in a CBROAMxx member of SYS1.PARMLIB.

1. New SETOAM keywords L2TAPEUNITNAME and L2DATACLASS are provided to define TSL2-related parameters to OAM. These new keywords let you specify a tape sublevel that will be used at the global and storage group level.
At the storage group level, L2TAPEUNITNAME(unitname) is an optional subparameter of the STORAGEGROUP parameter. L2TAPEUNITNAME specifies the type of tape drive that OAM uses when writing data to an object storage group using tape sublevel 2. L2TAPEUNITNAME must be specified in order to write to tape sublevel 2. L2DATACLASS(name) is an optional subparameter of the STORAGEGROUP parameter. L2DATACLASS specifies the SMS data class to be associated with an object storage group when using tape sublevel 2.

At the global level, L2DATACLASS(name) is an optional parameter that specifies the SMS data class to be used when storing objects to tape sublevel 2, for object storage groups that do not have their own L2DATACLASS specified on the STORAGEGROUP subparameter.

2. Existing SETOAM keywords need to be modified to correspond to the desired tape sublevel support hierarchy.

3. If the SETOAM statements are added to the CBROAMxx member and the system is backed down to a previous level, the changes to PARMLIB must also be removed.

For details on the new and changed SETOAM keywords, see z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

**Update SMS Storage Class constructs**

Set the new OAM Sublevel parameter in ISMF to modify existing storage class constructs or create new storage class constructs associated with tape sublevels TSL1 and TSL2. Pre-existing tape storage classes will default to TSL1. These storage class construct changes may necessitate updates to ACS routines.

Each level of the storage hierarchy is associated with SMS storage class (SC) constructs. These constructs were previously defined with the Initial Access Response Seconds (IARS) and the Sustained Data Rate (SDR) keywords in the ISMF storage class definition panels. The tape sublevel support adds a new OAM Sublevel (OSL) parameter to the storage class, to indicate the sublevel associated with that storage class.

- **Initial Access Response Seconds (IARS):**
  - 0 = DASD
  - 1-9999 = Removable Media

- **Sustained Data Rate (SDR):**
  - 0-2 = Optical
  - 3-999 = Tape

- **OAM Sublevel (OSL):**
  - 1 = OAM Sublevel 1 (default)
  - 2 = OAM Sublevel 2

For example: A storage class defined with IARS=1, SDR=3 and OSL=2 would equate to TSL2.

For more information, see z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support.

**Operating**

You can use operator commands to modify and display the new tape sublevel 2 settings for OAM.
**Modifying the SETOAMxx keywords**

You can use the `F OAM,UPDATE,SETOAM` command to add or change the current `L2DATACLASS` and `L2TAPEUNITNAME` specifications for a storage group, or to add or change the `L2DATACLASS` specification at the global level.

1. Specify `UPDATE,SETOAM` with the `L2DATACL` or `L2TAPEUN` keyword as shown in the following examples:

   **Example:**
   ```
   F OAM,UPDATE,SETOAM,scope,L2DATACL,dataclass
   ```
   This example updates the `L2DATACLASS` value of the SETOAM statement in the `CBROAMxx PARMLIB` member, with `dataclass` signifying the data class to be used for `scope`.

   **Example:**
   ```
   F OAM,UPDATE,SETOAM,scope,L2TAPEUN,unitname
   ```
   This example updates the `L2TAPEUNITNAME` value of the SETOAM statement in the `CBROAMxx PARMLIB` member, with `unitname` signifying the unit name to be used for `scope`.

**Displaying the new OAM tape level settings**

The `F OAM,DISPLAY` commands show the new tape level 2 settings, if any, in effect at the global and group levels.

1. Specify `SETOAM` with a scope of "global".

   **Example:**
   ```
   F OAM,D,SETOAM,GLOBAL
   ```
   The output from this example includes the global value in effect for `L2DATAACL`.

2. Specify `SETOAM` with a group name.

   **Example:**
   ```
   F OAM,D,SETOAM,groupname
   ```
   The output from this example includes the `L2TAPEUN` and `L2DATAACL` values in effect at the specified storage group level.

Other `OAM,DISPLAY` command keywords that include new tape sublevel information are `GROUP`, `VOLUME`, and `OSMC,taskname`. 
Chapter 20. Using the DFSMSrmm enhancements

The functional enhancements available with z/OS V1R9 DFSMSrmm provide you with these benefits:

- **CIM provider extension**
  The CIM subclasses provided in V1R8 are extended to differentiate the resources/objects that DFSMSrmm manages. This enhancement is intended to ensure that all resources known to DFSMSrmm are now available via CIM, and can be managed via the CIM Agent provider interface.

  The CIM provider is updated to support out-of-process mode functionality added to OpenPegasus CIM Server with 2.5.1. This has an impact on the way that the DFSMSrmm provider code is installed and run.

- **Performance, RAS, operability items**
  - **Task management:** You can now control long-running subsystem requests. These requests can be ended, held, and released. This enables better management when required either by system automation or by the operator because of operational priorities.
  - **Utility and inventory management:** EDGUTIL and EDGHSKP interaction with system managed volumes in an IBM system managed library is improved through multiple changes that should, especially in larger VTS installations, result in shorter elapsed time and more flexibility.
  - **Control data set serialization:** The serialization used by DFSMSrmm for its CDS is changed to require a CDSID. This avoids conflicts when multiple RMMplexes run in the same sysplex.

- **DFSMSrmm usability items**
  - The data set naming requirements for tape data sets supported by the TSO subcommands and API are relaxed so that any 44 character string can be used. This change enables use of unqualified data set names such as those that are accepted by the MVS™ JCL DSNAME keyword.
  - TSO subcommand parsing rules are further relaxed to support different product versions and declassification of data sets and volumes.
  - You can now more easily share the DFSMSrmm settings in SYS1.PARMLIB because system symbols are supported in EDGRMMxx parmlib member, and you can use indirection to an additional PARMLIB member that might contain system-specific options.
  - SEARCHxxxx subcommands with CLIST can now optionally append to an existing CLIST data set.
  - All SEARCHxxxx subcommands support a way to break large numbers of search results into manageable quantities, enabling you to scan through results by specifying the search argument, a continuation point, and a limit on the number of results to return.
  - Processing for SEARCHxxxx subcommands with CLIST is enhanced so that almost any format of CLIST data set is supported.
  - The SEARCHVOLUME subcommand now supports search by storage group name.
  - A new EDGRRPTE Rexx report is provided to aid with stacked volume management.
The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning to use DFSMSrmm enhancements”</td>
<td>• “CIM Provider enhancements”</td>
</tr>
<tr>
<td></td>
<td>• “DFSMSrmm Task Management enhancements” on page 97</td>
</tr>
<tr>
<td></td>
<td>• “DFSMSrmm utility and inventory management” on page 97</td>
</tr>
<tr>
<td></td>
<td>• “DFSMSrmm usability items” on page 98</td>
</tr>
<tr>
<td>Operating</td>
<td>• “Operating” on page 140</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>• “Diagnosing” on page 140</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and installation tasks that you would perform to use the DFSMSrmm enhancements.

Planning to use DFSMSrmm enhancements

CIM Provider enhancements

**Before you begin:** Be familiar with information about the following topic:

- CIM providers; see [z/OS DFSMSrmm Implementation and Customization Guide](#).

**Software requirements:** The enterprise level interface function requires z/OS V1R9. If the OpenPegasus Server runs on a different platform or system than DFSMSrmm, a fully functional Web service as well as an xmlCIM compliant product that supports Java, such as the OpenPegasus C++ CIMOM from The Open Group, is required to use the DFSMSrmm CIM provider. However, if the DFSMSrmm CIM Provider runs on OpenPegasus on the same z/OS system as DFSMSrmm, the Web service is not needed; instead the DFSMSrmm high level language API is used directly.

A z/OS version of the OpenPegasus CIMOM is available under the Unix System Services. If running the CIM provider under Linux®, the OpenPegasus CIMOM has to be downloaded from the OpenGroup’s web site at:

[http://www.opengroup.org](http://www.opengroup.org)

Either way, the CIMOM must be up and running in order to work with the DFSMSrmm CIM provider.

Now you are ready to replace the CIM provider code and classes.

**Replacing the previous CIM provider code and classes:** After the first IPL of z/OS V1R9, you must unregister the V1R8 CIM providers and unload all V1R8 CIM classes, using the rmmutil.sh tool. For more information, see: [z/OS DFSMSrmm Implementation and Customization Guide](#).

Next, you need to register the complete set of V1R9 providers and load the V1R9 CIM classes, using the same tool.
If you are migrating to z/OS V1R9 from z/OS V1R7, first follow the documented steps for migration to V1R8 before completing the V1R9 registration steps. For details on the migration steps, see: z/OS Migration.

Related reading: For more information about setting up the DFSMSrmm application programming interface via web services, see z/OS DFSMSrmm Application Programming Interface.

**DFSMSrmm Task Management enhancements**

**Before you begin:**

In previous releases, if a task or address space ended while waiting for DFSMSrmm processing to complete, the processing would continue and only when it was completed would DFSMSrmm check and discover that the requester had ended. In V1R9, long-running requests now check on the requester’s status and if the requester has ended, the current processing is interrupted and ended early. For example, if a batch inventory management job is cancelled, DFSMSrmm is notified and inventory management ends early.

In z/OS V1R9, DFSMSrmm operator commands also allow long running requests to be held, cancelled, and released, and allow new requests to be held and released. STOP command processing is changed to prevent DFSMSrmm from stopping if inventory management is running. The operator must now request that inventory management is to end, in order for STOP and MODIFY to process immediately.

For more information, see z/OS DFSMSrmm Implementation and Customization Guide.

**Software requirements:** The DFSMSrmm task management enhancements require z/OS V1R9.

**DFSMSrmm utility and inventory management**

Starting in z/OS V1R9, the EDGUTIL and EDGHSKP functions are independent. You can run them at the same time, which was previously not possible. Other utility and inventory management improvements include:

- When EDGUTIL runs with VERIFY(SMSTAPE) you can optionally generate control statements to enable mismatches to be synchronized later using the new utility EDGSPLCS.
- VERIFY(SMSTAPE) and MEND(SMSTAPE) now generate less I/O to system managed tape libraries because they retrieve information for multiple volumes at one time.
- A subset of volumes can be specified on invocations of EDGUTIL and EXPROC.
- A new option of EXPROC gives you the choice of scratching system managed volumes either synchronously as in the past, or after EXPROC has completed. The new EDGSPLCS utility can be used to scratch volumes and supports multiple executions in parallel using the same input file.
- You can run multiple copies of EDGUTIL. If you use each copy to process a different subset of volumes you can potentially reduce the time to complete processing of all volumes.

For more information, see z/OS DFSMSrmm Managing and Using Removable Media.

**Software requirements:** The DFSMSrmm utility and inventory management functions require z/OS V1R9.
**DFSMSrmm control data set serialization**

Starting in z/OS V1R9, you must supply a control data set identifier in the CDSID parameter of SYS1.PARMLIB member EDGRMMxx, specifying the identifier of the CDS to be used on that system. DFSMSrmm now uses the CDSID as part of the ENQ name used to serialize updates to the RMM CDS. The DFSMSrmm CIM Provider also uses the CDSID to distinguish between multiple control data sets. If the CDSID is not yet set in the CDS you can optionally use EDGUTIL UPDATE to set it, otherwise DFSMSrmm will set the CDSID when first started. IBM recommends that you use a unique CDSID for each CDS.

If you have entries in GRSRNLLxx for SYSZRMM you must review the need to update this information based on your CDSID and your existing entries. Refer to z/OS DFSMSrmm Implementation and Customization Guide for information on updating GRSRNLLxx. The entries for DFSMSrmm serialization should be updated on toleration systems and on z/OS V1R9 systems.

A toleration PTF for lower-level systems in an RMMplex allows DFSMSrmm to start without CDSID specified in EDGRMMxx, if the control data set itself has no CDSID. In that case, a warning message is issued and startup continues.

For more information, see z/OS DFSMSrmm Managing and Using Removable Media.

**DFSMSrmm usability items**

For additional information, see z/OS DFSMSrmm Implementation and Customization Guide.

**Software requirements:** The DFSMSrmm usability items require z/OS V1R9.

---

**Operating**

To run the DFSMSrmm application programming interface via web services, you must be running WebSphere Application Server or an equivalent. To run the CIM provider, you must be running an xmlCIM compliant product that supports Java, such as the OpenPegasus C++ CIMOM from The Open Group.

**Before you begin:** For additional information, see z/OS DFSMSrmm Implementation and Customization Guide.

---

**Diagnosing**

For information on diagnosing DFSMSrmm problems, see z/OS DFSMSrmm Diagnosis Guide.
Chapter 21. Using the DFSMSShsm enhancements

DFSMShsm's Functional Statistics Record (FSR) is essential for performing analysis of DFSMSShsm processing and performance. z/OS V1R9 includes a number of additions to the information recorded in the FSR.

Enhancements to the FSR include the following:

- An indication of when a recall has caused a tape-takeaway. This enhancement indicates that the recall caused another task or function to give up the use of a migration tape so that the recall could be performed. Note: this is the opposite of a recall being done as a "piggy-back" which is indicated in FSRFPIGB.
- Recording in the recall FSR of the number of times a migrated data set had been recycled before it was eventually recalled.
- Adding the recycle source volume's volser to the FSR. Although all the source volumes for a recycle are included in the FSR at FSRTAPE, this data is not easy to process.
- Providing the CPU time used for a partial release (FSR type 18) and for the expiration of a backup version (FSR type 19)
- Recording the number of tracks needed when an error occurred due to insufficient ML1 space.

For details of the FSR, including these enhancements, see [z/OS DFSMS Shsm Diagnosis](#).
## Part 4. Using New DFSMS Functions in z/OS V1R8

### Chapter 22. Using DFSMShsm fast replication, tape support
- Planning and Installation .................................................. 103
  - Planning to use DFSMShsm fast replication, tape support ........ 104
  - Setting up DFSMShsm fast replication, tape support .............. 104
- Administering ............................................................... 106
  - Performing fast replication backup and dumping to tape ......... 106
  - Dumping a volume backup copy from a specific date .............. 106
  - Recovering a volume from a dump tape ............................. 106
  - Deleting unneeded dump copies by token .......................... 106
  - Making dump tapes available for mount ............................ 107
- Operating ................................................................. 107
  - Listing backup dump information for copy pools .................. 107
  - Checking fast replication activity for copy pools ............... 108
  - Auditing fast replication CDS records for copy pools .......... 108

### Chapter 23. Using DFSMShsm fast replication, data set recovery
- Related DFSMSdss enhancements ..................................... 109
- Overview of tasks and procedures .................................... 110
- Planning and Installation .................................................. 110
  - Planning to use DFSMShsm fast replication, data set recovery .... 110
- Administering ............................................................... 111
  - Protecting fast replication commands with discrete profiles .... 111
  - Recovering a data set from a fast replication backup .......... 112
- Operating ................................................................. 112
  - Holding data set recoveries ......................................... 113
  - Releasing data set recoveries ....................................... 113
  - Canceling queued data set level copy pool recovery requests .... 113
  - Limiting the number of concurrent data set level recoveries .... 114
  - Selecting a data set level recovery method ....................... 114
  - Listing information for a copy pool ............................... 114
  - Performing a physical data set copy .............................. 115

### Chapter 24. Using PDSE enhancements
- Planning and Installation .................................................. 117
  - Planning to use PDSE 64-bit directory buffers ................... 117
  - Setting up the PDSE enhancements .................................. 118
- Administering ............................................................... 119
  - Monitoring the use of PDSE data set caching ....................... 119
- Operating ................................................................. 119
  - Changing the size of PDSE 64-bit directory storage ............. 119
  - Controlling the caching of PDSE data in memory after data set close 120

### Chapter 25. Using catalog enhancements
- Planning and Installation .................................................. 121
  - Planning to specify a dynamic service task count ................ 121
  - Setting the dynamic service task count ........................... 122
- Operating ................................................................. 122

### Chapter 26. Using the new diagnostic tools
- Chapter 27. Communicating with the device manager address space (MODIFY DEVMAN command) ........................................... 127
- Chapter 28. Creating an ACDS directly from a SCDS .................. 129

### Chapter 29. Enhanced volume selection performance
- Activating and displaying fast volume selection ................... 132

### Chapter 30. Using the Object Access Method enhancements
- Application Programming .................................................. 134
  - Implementing the LOB function ..................................... 134
  - Implementing the automated selection of RECYCLE volumes function .... 135
  - Implementing the immediate backup copy function ................ 135

### Chapter 31. Using the DFSMSrmm enhancements
- Planning and Installation .................................................. 138
  - Planning to use DFSMSrmm enhancements ........................... 138
  - Enterprise level interface ............................................. 138
  - DFSMSrmm UTC implementation ...................................... 139
  - DFSMSrmm vital record specification policy management simplification 139
  - DFSMSrmm usability items ............................................ 140
- Operating ................................................................. 140
  - Diagnosing ............................................................. 140

### Chapter 32. Using Tape Data Set Authorization
- Planning and Installation .................................................. 141
  - Planning to use tape data set authorization ....................... 141
  - Setting up tape data set authorization ............................. 142
- Administering ............................................................... 143
  - Implementing tape data set authorization ......................... 143

### Chapter 33. Using the DFSMShsm usability enhancements
- Planning and Installation .................................................. 145

© Copyright IBM Corp. 2004, 2009
Planning to use the DFSMShsm usability enhancements ......................................................... 146
Administering ......................................................................................................................... 147
Using the ARECOVER command to recover individual data sets from an aggregate backup . 147
  Recovering an individual data set from an aggregate backup .............................................. 147
  Recovering a group of data sets from an aggregate backup .................................................. 147
Using ISMF to recover individual data sets from an aggregate backup .................................. 148
  Altering the priority of queued requests ........................................................................... 148
Operating ............................................................................................................................... 149
  Marking migration tapes full for errors ............................................................................. 149
  Altering the priority of queued requests ........................................................................... 149
Chapter 22. Using DFSMShsm fast replication, tape support

In DFSMS, a copy pool is a defined set of pool storage groups that contains data that DFSMShsm can backup and recover collectively, using fast replication. DFSMShsm manages the use of volume-level fast replication functions, such as FlashCopy® and SnapShot. These functions provide point-in-time copy services that can quickly copy data from a source location to a target location.

In previous releases of z/OS, you could use DFSMShsm fast replication to create backups for a set of storage groups in a copy pool on target volumes. You could not, however, use DFSMShsm to dump the fast replication backups to tape.

Now, using DFSMShsm fast replication, tape support in z/OS V1R8, you can use DFSMShsm to dump the fast replication backup copies to tape. You can create these dumps on tape through the FRBACKUP command or through automatic dump processing. You can recover fast replication backups at the volume level from dumps or DASD copies, or at the copy pool level from DASD copies.

A further enhancement in z/OS V1R8 allows you to recover individual data sets from fast replication backups; see Chapter 23, “Using DFSMShsm fast replication, data set recovery,” on page 109.

The following table lists the types of tasks and associated procedures that you must complete to fully use DFSMShsm fast replication, tape support.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation” on page 104</td>
<td>• “Planning to use DFSMShsm fast replication, tape support” on page 104</td>
</tr>
<tr>
<td></td>
<td>• “Setting up DFSMShsm fast replication, tape support” on page 104</td>
</tr>
<tr>
<td>“Administering” on page 106</td>
<td>• “Performing fast replication backup and dumping to tape” on page 106</td>
</tr>
<tr>
<td></td>
<td>• “Dumping a volume backup copy from a specific date” on page 106</td>
</tr>
<tr>
<td></td>
<td>• “Recovering a volume from a dump tape” on page 106</td>
</tr>
<tr>
<td></td>
<td>• “Deleting unneeded dump copies by token” on page 106</td>
</tr>
<tr>
<td></td>
<td>• “Making dump tapes available for mount” on page 107</td>
</tr>
<tr>
<td>“Operating” on page 107</td>
<td>• “Listing backup dump information for copy pools” on page 107</td>
</tr>
<tr>
<td></td>
<td>• “Checking fast replication activity for copy pools” on page 108</td>
</tr>
<tr>
<td></td>
<td>• “Auditing fast replication CDS records for copy pools” on page 108</td>
</tr>
</tbody>
</table>
Planning and Installation

This section describes the planning and installation tasks that you would perform to use DFSMSShsm fast replication, tape support.

Planning to use DFSMSShsm fast replication, tape support

Before you begin: Read the topic on "Managing Volume Backups with Fast Replication" in z/OS DFSMSShsm Storage Administration to understand the concepts of DFSMSShsm fast replication.

Related reading:
- For information about how DFSMSdss supports DFSMSShsm fast replication functions, see z/OS DFSMSdss Storage Administration.
- For information about FlashCopy and SnapShot, see z/OS DFSMS Advanced Copy Services.

Software requirements: DFSMSShsm fast replication, tape support requires z/OS V1R8 and the supplied PTFs.

Coexistence requirements: In an HSMplex, apply toleration PTFs for any supported systems below z/OS V1R8.

Perform the following steps to plan for using the new function:
1. Determine whether you have copy pool backups that should be dumped to tape. If so, identify which copy pools will use the new function.

Now you are ready to set up copy pools to use DFSMSShsm fast replication, tape support.

If it becomes necessary to back-out to a lower level release, you can use the DFSMSdss COPY command to recover individual volumes if the output is available on a V1R8 system. You can also recover individual volumes from dump copies using the DFSMSdss RESTORE command.

Setting up DFSMSShsm fast replication, tape support

Before you begin:
1. To learn how to define a copy pool backup storage group, read the information on defining copy pools in z/OS DFSMSdss Storage Administration.
2. Back up the SMS Source Control Data Set (SCDS).

**Related reading:** For information about using ISMF, see [z/OS DFSMS Using the Interactive Storage Management Facility](z/OS DFSMS Using the Interactive Storage Management Facility).

Using a copy pool, you can specify the pool storage groups that you want DFSMShsm to process collectively for fast replication.

Perform the following steps to modify a copy pool construct to use DFSMShsm fast replication, tape support:

1. **Access the ISMF Primary Option Menu for Storage Administrators, as shown in z/OS DFSMSdfp Storage Administration**

2. Select option P, Copy Pool, to view the Copy Pool Application Selection panel.

3. Specify the CDS Name and Copy Pool Name parameters to select a copy pool to modify.

4. Select option 4, Alter, to view page 1 of the Copy Pool Alter Panel.

5. For the copy pool, specify the desired values for the new tape support options: Auto Dump, Dump Sys/Sys Group Name, and up to five dump classes, as described in z/OS DFSMSdfp Storage Administration. If you want the FlashCopy target volumes to be withdrawn immediately after the dump is complete, specify 0 (zero) for 'Number of Backup Versions.' This feature enables the target volumes to be used to create a version for another copy pool, but removes the ability to recover the backup version from disk using FlashCopy.

6. Enter the DOWN command to display page 2 of the panel. Specify the names of one or more valid pool storage groups. You can specify up to 256 valid pool storage group names on this and the pages that follow.

7. Enter the END command to save and exit the panel. You should validate and activate the SCDS before the SMS changes are effective.

To verify your changes to the copy pool environment, enter the FRBACKUP COPYPOOL PREPARE command for each copy pool that you modified.

**Example:**

```
FRBACKUP CP(COPYPOOL1) PREPARE
```

If you specified one or more backup versions for COPYPOOL1, this command verifies that a valid environment exists and assigns target volumes to each source volume. If you specified no backup versions for COPYPOOL1, this command verifies that sufficient target volumes exist for the backup copies, but does not assign target volumes to source volumes.

After you have finished modifying the copy pool constructs to use DFSMShsm fast replication, tape support, you can use DFSMShsm storage administrator commands to create, recover, and delete fast replication backup versions.
Administering

You can use the following DFSMShsm storage administrator commands to work with fast replication dumps:

- **FRBACKUP** command to dump a fast replication backup copy to tape
- **FRRECOV** command to recover a dump copy to a volume
- **FRDELETE** command to delete an unneeded fast replication dump copy
- **DEFINE DUMPCLASS** and **SETSYS** commands to customize the data set level copy pool recovery environment.

**Before you begin:** For information on using DFSMShsm storage administrator commands to control fast replication, see [z/OS DFSMSdfp Storage Administration](https://www.ibm.com/support/knowledgecenter/SSEPGG_1.10.0/com.ibm.zos.v1r10.doc/tmp/zos_basics/using.html).

### Performing fast replication backup and dumping to tape

1. Enter the **FRBACKUP** command.

   **Example:**
   
   ```
   FRBACKUP CP(COPYPOOL1) EXECUTE DUMP RETURNCONTROL(FREND)
   ```
   
   In this example, DFSMShsm creates a fast replication backup for each volume in each storage group defined in the COPYPOOL1 copy pool. Control is returned to the user after the fast replication backup version is created. Then, DFSMShsm dumps each of the target volumes to tape.

### Dumping a volume backup copy from a specific date

1. Enter the **FRBACKUP** command.

   **Example:**
   
   ```
   FRBACKUP CP(COPYPOOL1) DUMPONLY(DATE(2006/06/02))
   ```
   
   In this example, DFSMShsm dumps to tape the backup of COPYPOOL1 that was created on June 2, 2006.

### Recovering a volume from a dump tape

1. Enter the **FRRECOV** command.

   **Example:**
   
   ```
   FRRECOV TOVOLUME(SRC001) FROMDUMP(DUMPCLASS(DCLASS1))
   ```
   
   In this example, DFSMShsm recovers from dump tape the most recent version of volume SRC001. The DUMPCLASS subparameter is used to restrict the dump class used for the volume recovery to DCLASS1.

### Deleting unneeded dump copies by token

1. Enter the **FRDELETE** command.

   **Example:**
In this example, DFSMShsm deletes the dump copy of any fast replication backup for copy pool COPYPOOL1 associated with token FRBACKUP_JUNEDA. Use the DUMPONLY parameter to limit deletions to tape backups only, while preserving any DASD backups that might be associated with this token. Omitting DUMPONLY from this command causes both tape and DASD backups associated with this token to be deleted.

Making dump tapes available for mount

When defining a dump class, you can specify whether dump tapes created for the dump class are to be made available for mount. On the DEFINE DUMPCLASS command, specify the keyword FRRECOV with the parameter AVAILABLEFORMOUNT. You might find this function to be useful when you need to distinguish between dump tapes that will be kept on-site and dump tapes that will be taken off-site.

To indicate the mount choice for a dump class, do the following:
1. Enter the DEFINE DUMPCLASS command, as follows.

   Example:
   ```
   DEFINE DUMPCLASS(ONSITE FRRECOV(AVAILABLEFORMOUNT(YES)))
   ```

   In this example, the DEFINE DUMPCLASS command is used to define the dump class ONSITE and indicate that any tape copies created for this dump class will be available for mount.

   ```
   DEFINE DUMPCLASS(OFFSITE FRRECOV(AVAILABLEFORMOUNT(NO)))
   ```

   In this example, the dump class OFFSITE is defined. Here, the parameter AVAILABLEFORMOUNT(NO) is used to indicate that the tape copies for OFFSITE will not be available for mount. To use the dump tapes associated with this dump class for recovery, specify FROMDUMP(DUMPCLASS(OFFSITE)) on a FRRECOV request.

Operating

You can monitor the use of fast replication dump copies through the following DFSMShsm operator commands:

- LIST and QUERY commands to display information that helps you monitor fast replication dump copies
- AUDIT command to check the fast replication control data set (CDS) records for copy pools.

Before you begin: For information on using DFSMShsm operator commands to monitor the use of fast replication, see z/OS DFSMSdfp Storage Administration

Listing backup dump information for copy pools

1. Enter the LIST COPYPOOL command.

   Example:
LIST CP(COPYPOOL1) DUMPVOLS

In this example, DFSMShsm displays a list of source and volume information for all dumps for copy pool COPYPOOL1. To display only the dump class information for any dump copies that exist, use the NOVOLS parameter in place of DUMPVOLS.

**Checking fast replication activity for copy pools**

1. Enter the QUERY COPYPOOL command.

Example:

```
QUERY CP(COPYPOOL1)
```

In this example, DFSMShsm displays pending fast replication requests to indicate the status of copy pool COPYPOOL1.

**Auditing fast replication CDS records for copy pools**

To have DFSMShsm verify that the fast replication CDS record relationships are correct for a particular copy pool, or for all copy pools managed by DFSMShsm, enter the AUDIT command with the new COPYPOOL parameter. The AUDIT function identifies any discrepancies that are discovered.

1. Enter the AUDIT COPYPOOL command.

Example:

```
AUDIT CP(COPYPOOL1)
```

In this example, DFSMShsm audits the copy pool records for copy pool COPYPOOL1. To audit all of the copy pools managed by DFSMShsm, enter AUDIT COPYPOOL without specifying a copy pool name.
Chapter 23. Using DFSMSHsm fast replication, data set recovery

DFSMShsm now supports recovering individual data sets from copy pool backup versions. In previous releases, you could recover backups at the volume or copy pool level, but you could not recover individual data sets.

To recover individual data sets from copy pool backups, you use the FRRECOV command, which is expanded in this release with new parameters that allow you to specify one or more fully or partially qualified data set names. You can recover data sets from FlashCopy target volumes or dump tapes (now supported for recovery; see Chapter 22, “Using DFSMSHsm fast replication, tape support,” on page 103]. Data set recovery can occur through either fast replication or traditional copy methods.

To be recovered, a data set must be cataloged and allocated on the same volumes on which it resided when the backup was created. Do not use this function for data sets that have been reorganized recently and likely to have moved to a source volume other than the one on which it resided for the backup.

The backup version being recovered can reside on either disk or tape. If the version resides on both disk and tape, DFSMShsm recovers the backup from disk (by default). In such cases, the recovery can be performed through either fast replication or traditional copy methods. Up to 64 concurrent data set recoveries are allowed.

Other DFSMShsm commands are enhanced to help you set up and manage the use of data set level copy pool requests, as follows:

- HOLD and RELEASE operator commands to control data set recoveries
- CANCEL command to cancel queued data set level copy pool recovery requests
- LIST and QUERY commands to display information about data set level copy pool recovery requests.

This release changes the FACILITY class resource names for DFSMShsm fast replication commands. The names are shortened to allow security administrators to specify the full names of copy pools in profiles for fast replication commands.

A related enhancement in z/OS V1R8 allows you to use DFSMShsm to dump fast replication backup copies to tape; see Chapter 22, “Using DFSMShsm fast replication, tape support,” on page 103.

Related DFSMSdss enhancements

To support DFSMShsm fast replication, data set recovery, DFSMSdss adds a new function called physical data set copy. This function allows you to use DFSMSdss to copy a single volume data set, or the parts of a multivolume data set, from one DASD volume to another DASD volume. To perform a physical data set copy, you use the COPY DATASET command.

DFMSdss supports physical data set copy from traditional volumes and conditioned volumes created through the DFSMSdss COPY FULL DUMPCONDITIONING command.
Overview of tasks and procedures

The following table lists the types of tasks and associated procedures that you must complete to fully use DFSMShsm fast replication, data set recovery.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation”</td>
<td>• “Planning to use DFSMShsm fast replication, data set recovery”</td>
</tr>
<tr>
<td>“Administering” on page 111</td>
<td>• “Protecting fast replication commands with discrete profiles” on page 111</td>
</tr>
<tr>
<td></td>
<td>• “Recovering a data set from a fast replication backup” on page 112</td>
</tr>
<tr>
<td>“Operating” on page 112</td>
<td>• “Holding data set recoveries” on page 113</td>
</tr>
<tr>
<td></td>
<td>• “Releasing data set recoveries” on page 113</td>
</tr>
<tr>
<td></td>
<td>• “Canceling queued data set level copy pool recovery requests” on page 113</td>
</tr>
<tr>
<td></td>
<td>• “Limiting the number of concurrent data set level recoveries” on page 114</td>
</tr>
<tr>
<td></td>
<td>• “Selecting a data set level recovery method” on page 114</td>
</tr>
<tr>
<td></td>
<td>• “Listing information for a copy pool” on page 114</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and installation tasks that you must perform to use DFSMShsm fast replication, data set recovery.

Planning to use DFSMShsm fast replication, data set recovery

Before you begin: To understand the concepts of DFSMShsm fast replication, see the topic on "Managing Volume Backups with Fast Replication" in z/OS DFSMShsm Storage Administration.

Related reading: For information about how DFSMSdss supports DFSMShsm fast replication functions, see z/OS DFSMSdfp Storage Administration.

Software requirements: DFSMShsm fast replication, data set recovery requires z/OS V1R8 and the supplied PTFs.

Coexistence requirements: None. For an HSMplex, no coexistence changes are required for supported pre-z/OS V1R8 systems. Also, your installation’s user interaction modules do not require changes to continue working with DFSMSdss.

z/OS V1R8 restricts the names of any new copy pools that you define to a maximum of 23 characters. z/OS V1R8 continues to support the use of longer names (up to 30 characters) for your existing copy pools.

Perform the following steps to plan for using the new function:

1. Plan to use the new FACILITY class resource names introduced in this release. The new names are shorter to allow you to specify full names of copy pools without exceeding the 39-character limit for FACILITY class profiles.
In mixed releases of z/OS in an HSMplex, you will need to define both the older profiles and the newer profiles. In z/OS V1R8, DFSMSshm checks for the newer profiles, but your pre-V1R8 systems will continue to use the older profiles. For pre-z/OS V1R8 systems in an HSMplex, do not actually convert your existing FACILITY class profiles until you have migrated all such systems to z/OS V1R8. Instead, retain the older profiles until all of your systems are at the z/OS V1R8 level. Saving this task until after the migration will allow you to recover individual data sets from any copy pool backup version made on any supported system in the HSMplex (z/OS V1R5 and later).

2. When defining new copy pools, be aware that ISMF now limits your name selections to 23 characters. The shorter length allows you to specify full names of copy pools in your installation’s FACILITY class resource names for DFSMSshm fast replication commands.

If you do not plan to allow data set recoveries from tape at your installation, you can use the SETSYS command to prevent such requests (see “Limiting the number of concurrent data set level recoveries” on page 114).

Administering

This section describes the administration tasks that you must perform to use DFSMSshm fast replication, data set recovery.

Before you begin:
• Read the section on authorizing and protecting DFSMSshm commands in z/OS DFSMSdfp Implementation and Customization Guide
• Read the section on managing volume backups with fast replication in SC35-0421

Related reading:
• For information about defining FACILITY class profiles, see z/OS Security Server RACF Security Administrator’s Guide
• For information on using DFSMSshm storage administrator commands to control fast replication, see z/OS DFSMSdss Storage Administration

Protecting fast replication commands with discrete profiles

This release changes the FACILITY class resource names for DFSMSshm fast replication commands. The names are shortened to allow you to specify the full names of copy pools.

To authorize or deny the use of DFSMShm fast replication commands, define the profiles shown in Table 3.

Table 3. Profiles for DFSMShm fast replication commands

<table>
<thead>
<tr>
<th>Command Name</th>
<th>FACILITY Class Resource Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRBACKUP</td>
<td>STGADMIN.ARC.FB.*, or</td>
</tr>
<tr>
<td></td>
<td>STGADMIN.ARC.FB.cpname</td>
</tr>
<tr>
<td>FRRECOV</td>
<td>STGADMIN.ARC.FR.*, or</td>
</tr>
<tr>
<td></td>
<td>STGADMIN.ARC.FR.cpname</td>
</tr>
<tr>
<td>FRDELETE</td>
<td>STGADMIN.ARC.FD.*, or</td>
</tr>
<tr>
<td></td>
<td>STGADMIN.ARC.FD.cpname</td>
</tr>
</tbody>
</table>
To enable the security environment for these commands, you must activate the FACILITY class before starting DFSMShsm. When started, DFSMShsm checks to determine whether the FACILITY class is active. If you have not defined the new profiles, the DFSMShsm commands fail.

Recovering a data set from a fast replication backup

Use the FRRECOV storage administrator command to recover individual data sets from a copy pool backup.

On FRRECOV, you can specify one or more data set names (fully or partially qualified). If you specify a partially qualified data set name with a filter (wild card), DFSMShsm processes every cataloged data set in the standard search order that matches your filtering criteria.

FRRECOV processes each data set independently. If a data set is not found, FRRECOV continues processing until the other specified data sets are processed. On completion, FRRECOV processing issues messages to indicate which data sets were recovered and which requests failed. For wait-type requests, FRRECOV returns control when all of the data set recoveries have completed.

For a specified data set name, the high-level qualifier must be fully or partially qualified. You cannot specify two asterisks ("**") as the high-level qualifier or as the only qualifier. You can specify "*" the high-level qualifier as long as it is not the first character specified. The remainder of the data set name can include any arrangement of wildcard characters. These restrictions help to prevent inadvertent requests for large numbers of data sets.

You cannot recover (individually) the following types of data sets:
- User catalog
- VVDS
- VTOC index
- GDG base
- VSAM key-range
- Migrated data sets

To recover individual data sets from a fast replication backup, do the following.

1. Enter the FRRECOV command.

Example:

```
FRRECOV DSNAME(DSNAME1,DSNAME2) REPLACE
```

In this example, DFSMShsm requests a fast replication recovery for two data sets (DSNAME1 and DSNAME2).

Operating

This section describes the operations tasks that you must perform to use DFSMShsm fast replication, data set recovery.

DFSMShsm provides the following operator commands to help you set up and manage the use of data set level copy pool requests:
- HOLD and RELEASE commands to control data set recoveries
- CANCEL command to cancel queued data set level copy pool recovery requests
• LIST and QUERY commands to display information about data set level copy pool recovery requests.

Related reading: For information about using DFSMShsm operator commands to control the use of fast replication, see z/OS DFSMSdfp Storage Administration.

Holding data set recoveries
To hold data set recoveries from DASD, do the following.
1. Enter the HOLD FRRECOV command, as follows.

   Example:
   
   ```
   HOLD FRRECOV(DATASET)
   ```

   In this example, optional parameter DATASET is used to hold all data set recoveries (from both DASD and tape). To hold only data set recoveries from tape, specify the parameter TAPE instead of DATASET.

Releasing data set recoveries
To release previously held data set recoveries from DASD, do the following.
1. Enter the RELEASE command, as follows.

   Example:
   
   ```
   RELEASE FRRECOV(DATASET)
   ```

   In this example, optional parameter DATASET is used to release data set recoveries. To release held data set recoveries from tape, specify the parameter TAPE instead of DATASET.

Canceling queued data set level copy pool recovery requests
Canceling a queued recovery request is a two-step process. Use the QUERY command to obtain the request number of the command to be canceled. Then, enter the CANCEL command for the request number.

Use the following command to identify commands that can be canceled.

Example:

```
QUERY REQUEST
```

In this example, the QUERY command returns a list of the requests that are waiting to be processed.

Use the following command to cancel a waiting request.

Example:

```
CANCEL REQUEST(num)
```

In this example, the CANCEL command cancels the request identified by `num`.
**Limiting the number of concurrent data set level recoveries**

To limit the number of concurrent data set recoveries that can be performed, do the following.

1. Enter the SETSYS command, as follows.

   **Example:**
   ```
   SETSYS MAXDSRECOVERTASKS(50)
   ```

   In this example, existing keyword `MAXDSRECOVERTASKS` is used to limit the total number of concurrent recovery requests to 50 (up to 64 are allowed). The value you specify for `MAXDSRECOVERTASKS` applies to data set recovery tasks for both DASD and tape. To limit tape requests to a subset of this value, use the new keyword `MAXDSTAPERECOVERTASKS` instead.

   The maximum number of tasks from DASD is the value of `MAXDSRECOVERTASKS` minus the current number of tape tasks, which is limited by `MAXDSTAPERECOVERTASKS`. To prevent any data set recoveries from tape, specify `SETSYS MAXDSTAPERECOVERTASKS(0)`.

**Selecting a data set level recovery method**

If a backup version resides on disk, DFSMS/SHsm can perform data set recovery through either fast replication or traditional copy methods. You can use the SETSYS command to select a preferred method for data set recoveries.

To indicate the method in which data set recoveries are to be performed, do the following.

1. Enter the SETSYS command, as follows.

   **Example:**
   ```
   SETSYS FASTREPLICATION(DATASETRECOVERY(PREFERRED))
   ```

   In this example, `PREFERRED` is specified to indicate that fast replication should be used whenever possible. For situations in which fast replication cannot be used to recover a data set, recovery is done through traditional copy methods.

   To use only fast replication, specify `FASTREPLICATION(DATASETRECOVERY(REQUIRED))`. Here, if fast replication cannot be used, the recovery request fails. To use only traditional copy methods, specify `FASTREPLICATION(DATASETRECOVERY(NONE))`.

**Listing information for a copy pool**

To list the information for a copy pool, do the following.

1. Enter the LIST COPYPOOL command, as follows.

   **Example:**
   ```
   LIST COPYPOOL(COPYPOOL1) ALLVOLS(TOKEN(token))
   ```

   In this example, the new keyword `TOKEN` is used to limit the LIST information to recoveries associated with a specific FRRECOV request.
Performing a physical data set copy

To perform a physical data set copy, do the following.
1. Enter the COPY DATASET command, as follows.

```
COPY DATASET(INCLUDE(**)) -
  PHYSINDDNAME(DASD1) OUTDYNAM(VOLS02) -
  REPLACE
```

Example:

In this example, only one volume may be specified on the PHYSINDD or PHYSINDYNAM keyword.
Chapter 24. Using PDSE enhancements

z/OS V1R8 provides several enhancements to improve the capacity, performance, and monitoring of partitioned data set extended (PDSE) processing.

In previous releases of z/OS, the number of concurrently opened PDSE members that could exist on a system was limited by the PDSE address spaces’ use of 31-bit addressable directory buffers. Now, with z/OS V1R8, you can specify an amount of 64-bit virtual storage to be used to cache PDSE directory buffers in the PDSE address spaces. Specifying 64-bit virtual storage can help you increase the maximum number of concurrently open PDSE members and avoid possible directory space constraints.

In addition, z/OS V1R8 allows you to retain directory and member data in memory cache after the close of a PDSE data set. Specifying this option can improve performance for programs that repeatedly open, read, and close the same members of a PDSE. A new subtype record in SMF records 14 and 15 also allows you to monitor PDSE data set caching statistics, by OPEN or by storage group.

In previous releases, you could specify the size of hiperspace storage for the SMSPDSE1 address space only as a parameter in IGDSMSxx. With z/OS V1R8, a new parameter on the SETSMS command allows a more dynamic change of this hiperspace size value (the SMSPDSE1 address space must be restarted for the change to take effect).

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation”</td>
<td>• “Planning to use PDSE 64-bit directory buffers”</td>
</tr>
<tr>
<td>“Administering” on page 119</td>
<td>• “Monitoring the use of PDSE data set caching” on page 119</td>
</tr>
<tr>
<td>“Operating” on page 119</td>
<td>• “Changing the size of PDSE 64-bit directory storage” on page 119</td>
</tr>
<tr>
<td></td>
<td>• “Controlling the caching of PDSE data in memory after data set close” on page 120</td>
</tr>
<tr>
<td></td>
<td>• “Specifying the hiperspace size for caching PDSE member data” on page 120</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and setup tasks that you would perform to use PDSE 64-bit directory buffers.

Planning to use PDSE 64-bit directory buffers

Before you begin: To understand the concepts of PDSEs, read z/OS DFSMS Using Data Sets

Software requirements: Using PDSE 64-bit directory buffers requires z/OS V1R8.
**Coexistence requirements:** By default, the system attempts to limit each PDSE address space (SMSPDSE and SMSPDSE1) to two gigabytes of 64-bit virtual storage for directory buffers. You can specify a value of between 64 megabytes and 16 gigabytes for 64-bit virtual storage.

If you plan to use this function in a sysplex containing both z/OS V1R8 and pre-V1R8 systems, note that only the V1R8 systems can use 64-bit directory buffers. In pre-V1R8 systems, directory buffers continue to reside in a 2-gigabyte data space.

Perform the following steps to plan for using the new function:

1. Determine whether to limit the amount of 64-bit virtual storage used for PDSE directory buffers. IBM recommends that you start with the default of two gigabytes.

2. Determine whether to keep PDSE directory and member data in memory cache beyond the last close of a PDSE data set on the system. By default, a PDSE’s directory and member data are purged from the in-memory cache after the last close (when a PDSE is closed and no longer open by any DCB on the system). Your planning should include checking the PDSE caching statistics, as shown in SMF record Types 14 and 15, subtype 6, and in SMF record type 42. A large number of reads and few cache hits, for example, might indicate the need for data to be retained in memory cache.

Now you are ready to set up the PDSE enhancements.

**Setting up the PDSE enhancements**

**Before you begin:** To understand the setup tasks required for PDSEs, read z/OS DFSMS Using Data Sets.

**Related reading:** For information about modifying DFSMS options, see “Initializing SMS through the IGDMSxx member” in z/OS DFSMSdfp Storage Administration and the description of member IGDMSxx in z/OS MVS Initialization and Tuning Reference.

To specify an amount of 64-bit virtual storage to be used for PDSE directory buffers, perform the following steps as appropriate for the PDSE address spaces (SMSPDSE, SMSPDSE1, or both):

1. For the SMSPDSE address space, define the keyword PDSE_DIRECTORY_STORAGE(nn|2G) in the active IGDMSxx parmlib member. After the next IPL, the specified number of megabytes (nnM) or gigabytes (nnG) of 64-bit virtual storage will be used to cache PDSE directory buffers in the SMSPDSE address space. By default, two gigabytes of 64-bit virtual storage will be used.

2. For the SMSPDSE1 address space, define the keyword PDSE1_DIRECTORY_STORAGE(nn|2G) in the active IGDMSxx parmlib member. After the next IPL, the specified number of megabytes (nnM) or gigabytes (nnG) of 64-bit virtual storage will be used to cache PDSE directory buffers in the SMSPDSE1 address space. By default, two gigabytes of 64-bit virtual storage will be used.
To use the new option of retaining PDSE buffers in memory after a data set is closed, perform the following steps as appropriate for the PDSE address spaces (SMSPDSE, SMSPDSE1, or both):

1. For the SMSPDSE address space, define the keyword
   PDSE_BUFFER_BEYOND_CLOSE(YES) in the active IGDSMSxx parmlib member. After the next IPL, directory and member data will be retained in memory cache beyond the last close of each PDSE data set.

2. For the SMSPDSE1 restartable address space, define the keyword
   PDSE1_BUFFER_BEYOND_CLOSE(YES) in the active IGDSMSxx parmlib member. After the next IPL, directory and member data will be retained in memory cache beyond the last close of each PDSE data set.

Administering

For PDSE enhancements in z/OS V1R8, these are the administration tasks.

Monitoring the use of PDSE data set caching

To monitor the use of PDSE data set caching, do the following:

1. If your installation specified a hiperspace for caching PDSE data, you can monitor the count of read requests and read hits for directories or members by checking subtype 6 of SMF records 14 and 15, as follows:
   - Offset 4 (SMF14DRD) indicates the number of directory read requests
   - Offset 8 (SMF14DRDH) indicates the number of directory read hits
   - Offset 12 (SMF14MRD) indicates the number of member read requests
   - Offset 16 (SMF14MRDH) indicates the number of member read hits

2. You can monitor overall PDSE caching statistics using SMF record 42.

Operating

You can use operator commands to change the size of PDSE 64-bit directory storage, to specify whether PDSE buffers are cached in memory after the last PDSE close, and specify the size of a hiperspace to be used for caching PDSE member data for the SMSPDSE1 address space.

Before you begin: For more information about using system commands to operate DFSMS, see z/OS MVS System Commands.

Changing the size of PDSE 64-bit directory storage

You can use operator commands to change the size of 64-bit directory buffer storage that was defaulted for the SMSPDSE1 address space at IPL or specified in the IGDSMSxx member of SYS1.PARMLIB. Note: it is necessary to restart the SMSPDSE1 address space in order for this change to take effect.

1. Enter the SETSMS command.

Example:
   SETSMS PDSE1_DIRECTORY_STORAGE(500M)
   This example specifies a 500 megabyte size for the SMSPDSE1 address space’s 64-bit virtual directory cache.
Controlling the caching of PDSE data in memory after data set close

You can use an operator commands to change the buffer-beyond-close option that was defaulted for the SMSPDSE1 address space at IPL or specified in the IGDSMSxx member of SYS1.PARMLIB. Note: it is necessary to restart the SMSPDSE1 address space in order for this change to take effect.

1. Enter the SETSMS command.

   Example:
   SETSMS PDSE1_BUFFER_BEYOND_CLOSE(YES)
   This example keeps directory and member data in memory beyond the last close on this system of a PDSE data set.

2. Enter the SETSMS command.

   Example:
   SETSMS PDSE1_BUFFER_BEYOND_CLOSE(NO)
   This example causes PDSE directory and member data to be purged from the in-memory cache when the last close of the data set occurs.

Specifying the hiperspace size for caching PDSE member data

You can use operator commands to specify a hiperspace size for SMSPDSE1 to use for caching PDSE member data. Specifying a size of zero indicates that a hiperspace is not to be created and the system will not cache PDSE member data. Note: it is necessary to restart the SMSPDSE1 address space in order for this change to take effect.

1. Enter the SETSMS command.

   Example:
   SETSMS PDSE1_HSP_SIZE(2047)
   This example requests 2047 megabytes (the maximum) for the PDSE1 hiperspace.

2. Enter the SETSMS command.

   Example:
   SETSMS PDSE1_HSP_SIZE(0)
   This example specifies that no hiperspace be created for the SMSPDSE1 address space to use in caching member data.
Chapter 25. Using catalog enhancements

z/OS V1R8 provides several enhancements to improve the capacity and performance of catalog processing.

In previous releases of z/OS, the size of page spaces was limited to four gigabytes. In z/OS V1R8, this limit is removed to allow larger page spaces, to take advantage of larger DASD capacities. There is no change to the method for creating page spaces (the DEFINE PAGESPACE command), other than removal of the previously documented size limit.

In addition, z/OS V1R8 increases the current maximum number of concurrent catalog requests from the current limit of 180 to a new default of 200. You can specify a maximum number of up to 999, by updating the SYSCATxx member of SYS1.NUCLEUS. Specifying a higher number of concurrent requests can improve performance, as more catalog service tasks are available to process requests, and tasks need not wait until a service task is available to process the request.

The IDCAMS LISTCAT command is enhanced to improve the output display of catalog entries. In z/OS V1R8, LISTCAT lists entries in sorted order, grouped by the catalog that they belong to. In previous releases, the displayed entries were listed in sorted order regardless of their respective catalogs.

For listings involving only one catalog, this enhancement has no effect on the output. For listings involving multiple catalogs, the order of the display is now sorted within a particular catalog.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation”</td>
<td>• “Planning to specify a dynamic service task count”</td>
</tr>
<tr>
<td>“Operating” on page 122</td>
<td>• “Operating” on page 122</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and setup tasks that you would perform to specify a dynamic service task count.

Planning to specify a dynamic service task count

Before you begin: To understand the service task concept for the catalog address space, see z/OS DFSMS Managing Catalogs.

Software requirements: Using the dynamic service task count requires z/OS V1R8 and the supplied PTFs.

Coexistence requirements:
In z/OS V1R8, the catalog address space uses a default of 200 service tasks. If this appears to be insufficient, plan to increase the number of service tasks. The maximum value, set in the SYSCATxx member of SYS1.NUCLEUS, is 999.

Perform the following steps to plan for using the new function:
1. Determine whether to increase the number of service tasks for the catalog address space. Indicators of an insufficient number of tasks can include:
   - Slow performance, indicated by output from the F CATALOG,REPORT,PERFORMANCE command
   - Output from the F CATALOG,REPORT command showing a value of 180 or more for the highest number of service tasks used.

IBM recommends that you start with the default of 200 service tasks.

Setting the dynamic service task count

Before you begin: To understand the setup tasks required for the catalog address space, read z/OS DFSMS Managing Catalogs.

To specify a number of service tasks for the catalog address space, perform the following steps:
1. In the SYSCATxx member of SYS1.NUCLEUS, columns 65-67 contain a three-character EBCDIC value for the number of service tasks to be defined. The default value is 200, which is used if the value is blank or less than 200. To specify a higher number, up to 999, enter that number in the three-character field.

Of the number specified, 10% of the service tasks are reserved for recursive catalog calls such as those resulting from the need to allocate catalogs. The other 90% are used as the maximum number of user catalog requests that can be processed concurrently.

Note: This parameter cannot be specified in the SYSCAT entry of a LOADxx member.

Operating

This section describes the new format of output returned by the IDCAMS LISTCAT command.

The LISTCAT command is enhanced to sorting catalog entries within the catalogs that they belong to. This new sorting order takes effect when you use the IDCAMS LISTCAT command with the LEVEL parameter, or the ALL parameter without LEVEL or ENTRIES.

For more information about using the LISTCAT command, see z/OS DFSMS Access Method Services for Catalogs.
- For example, enter the LISTCAT command with the LEVEL or ALL parameters:
  
  Example:
  LISTCAT CAT(USERNAME1) ALL
  This example lists all entries for the catalog named username1.
Example:
LISTCAT LVL(SYS1) ALL
This example lists all catalog entries with a high-level qualifier of sys1.

- The output from these commands lists the entries in sorted order, within the catalogs to which they belong.
Chapter 26. Using the new diagnostic tools

z/OS V1R8 DFSMS provides the following new diagnostic tools:

1. A new SMS trace event, DEBUG, has been introduced to allow trace points to be added temporarily by IBM support personnel to aid in problem determination. This approach will significantly reduce the number of trace entries, while providing specific data that the normal trace entries do not provide. To invoke this support, one or more modules will be modified, depending on the problem scenario, to add specific trace points for the DEBUG trace event. After the modified modules are installed, this specific trace can be turned on by issuing the SMS trace command, SETSMS, with ‘DEBUG’ as the selected trace option. For example:

```plaintext
SETSMS DESELECT(ALL)
SETSMS TRACE(ON) TYPE(ALL) SIZE(8M) SELECT(DEBUG)
```

The DEBUG trace event may be specified along with other events. See [z/OS DFSMSdfp Diagnosis, GY27-7618](#) for more information.

2. VSAM Record Management will automatically generate a dump when certain unexpected errors, such as an invalid CI or a physical I/O error, are detected. No action is necessary by the user authorize these dumps.

3. In addition to the automatic dumps previously described, you can also optionally specify whether VSAM Record Management dynamic dumping is to occur for specific events identified by a combination of a return code, error (reason) code, problem determination function (PDF or FUNC) code, and component code returned in the request parameter list (RPL) feedback area. You can request a user directed dump (or change the parameters for an existing dump request) with the MODIFY CATALOG,VDUMPON command. You can turn off the dump request with the MODIFY CATALOG,VDUMPOFF command. See [z/OS MVS System Commands, SA22-7627](#) and [z/OS DFSMSdfp Diagnosis, GY27-7618](#) for a complete description of the MODIFY CATALOG,VDUMPON and MODIFY CATALOG,VDUMPOFF commands.

4. If you suspect that VSAM RLS latch contention is causing a hang or deadlock, use the following operator command:

```plaintext
DISPLAY SMS,SMSVSAM,DIAG(CONTENTION)
```

This command will produce a console message that either:

a. Displays all latch contention active on the system (if any), or
b. Indicates that there is no latch contention on this system.

For more information on the DISPLAY SMS,SMSVSAM,DIAG(CONTENTION) command, refer to [z/OS MVS System Commands, SA22-7627](#) and [z/OS DFSMSdfp Diagnosis, GY27-7618](#).
Chapter 27. Communicating with the device manager address space (MODIFY DEVMAN command)

z/OS now provides a MODIFY DEVMAN command to enable you to display information about, or to request a specified service from, the device manager address space (DEVMAN). This command provides the following functions:

**DUMP**
Captures a diagnostic dump of the device manager address space, including the dataspace that contains device manager ctrace records.

**REPORT**
Provides basic information about the current activity and module levels for the device manager address space.

**RESTART**
Terminates the device manager address space then restarts the device manager in a new address space. Any subtasks that are active in the device manager address space at the time of the restart are allowed to complete. The time allowed for subtask completion is determined by using the average time taken by previous subtasks. Any subtasks that do not complete in time are abnormally ended before the address space is restarted.

For more information on the MODIFY DEVMAN command, refer to [z/OS MVS System Commands, SA22-7627](https://www.ibm.com/support/knowledgecenter/en/SSLTBW_2.4.0/com.ibm.zos.v2r4.mc122r4.doc/sect4modifdevman.htm).
Chapter 28. Creating an ACDS directly from a SCDS

In prior releases of z/OS, when SMS was active, the current ACDS could be saved to another data set by issuing a command, but there was no operator command to create an ACDS directly from a SCDS without also activating the SCDS. As a result, there was no easy way to create an ACDS intended for use on some other SYSTEM or SYSPLEX, which made it difficult to create an ACDS to be used elsewhere, such as in disaster recovery situations.

In z/OS DFSMS V1R8, a new COPYSCDS keyword has been added to the SETSMS command, which will allow the user to create an ACDS from any valid SCDS. The scds_dsn variable must point to a SCDS that has been validated. The acds_dsn variable must be the name of a preallocated data set with space equal to or greater than the input SCDS. The command format is:

\[
\text{SETSMS COPYSCDS} \langle \text{scds	dsn, acds	dsn} \rangle
\]

where scds_dsn is the name of the SCDS that will be copied to the target data set acds_dsn. SMS will verify that acds_dsn is not the current ACDS or COMMDS. SMS will also verify that the scds_dsn is a SCDS data set. If acds_dsn is not the current ACDS or COMMDS, the contents of scds_dsn will be copied to the acds_dsn.

See z/OS MVS System Commands, SA22-7627 for more information on the SETSMS command.
Chapter 29. Enhanced volume selection performance

Because the number of volumes at customer installations has greatly increased, situations have occurred where volume selection has taken an unacceptably long time owing to the number of retries. The problem usually occurs when the available space on candidate volumes is stressed, an allocation requests a large amount of space, or a large data set is restored/recalled by DFSMSdss/DFSMShsm. In prior releases, when these situations occurred, SMS volume selection did not fail an allocation until all eligible volumes had been tried and rejected. To improve efficiency on SMS volume selection, SMS will now:

1. **Exclude a volume from selection when requested space exceeds volume size:**
   For non-best-fit allocations, SMS will exclude the volume from selection when the requested space exceeds its total capacity. If none of the candidate volumes has sufficient total space, SMS will transition immediately to Space Constraint Relief if specified, or fail allocation if Space Constraint Relief is not specified.

2. **Limit the number of retries for both best-fit and non-best-fit allocations:** SMS will limit the number of retries for both best-fit and non-best-fit allocations. For non-best-fit requests, SMS will perform volume selection from the prioritized list until 100 volumes have been rejected by DADSM for insufficient space. When that occurs, SMS will exclude all volumes that do not have sufficient free space based on the volume statistics in the SMS configuration. This ‘fast’ volume selection approach may drastically reduce the number of candidate volumes, and thus the number of retries, while continuing to select volumes by volume preference ranking as in the ‘normal’ volume selection approach. An installation can activate this ‘fast’ volume selection approach by using an SMS parameter or command, which are described in "Activating and displaying fast volume selection" on page 132.

**Notes:**

a. ‘Fast’ volume selection does not apply to striping allocation. SMS striping allocation has already excluded the volumes that are above high threshold from volume selection.

b. ‘Fast’ volume selection may inadvertently exclude volumes that have sufficient free space but whose SMS volume statistics indicate that they do not. Such invalid SMS volume statistics can occur when the VTOC index is broken, when OEM products bypass CVAF processing, or in a SMSplex when the SMS synchronization time interval has not yet been driven to update the SMS configuration with the most current space statistics. However, these special situations should have only a negligible effect on ‘fast’ volume selection, because it will first use the ‘normal’ approach to select volumes and will not exclude the remaining volumes for insufficient space until 100 volumes have been rejected by DADSM for insufficient space.

For best-fit requests, all the space does not have to be gotten on one volume. Space can be spread out on a number of volumes as long as this number does not exceed a ‘maximum count’ that is specified on the call to SMS. Therefore, for best-fit requests, SMS will call DADSM interactively until space requested has been gotten or the maximum number of volumes has been used and requested space has not yet been completely gotten. In this latter case, SMS will reject only the last volume that was used and will re-select another volume. For example, if the maximum volume count was 4 and after SMS allocates space on 4 volumes space is still needed, SMS will reject only the last (fourth volume)
that was selected and will select another one to replace it and then attempt space allocation again. In the non-VSAM path, a storage group is rejected when the maximum volume count associated with a best-fit request has been exhausted without the Primary Space quantity requirement having been met. In the VSAM path, SMS will reject a storage group after 59 volumes that are most preferred are rejected for insufficient space. The remaining volumes in the storage group are not considered for further selection, because they are less preferred and therefore deemed to have less chance to fulfill the allocation request.

3. **Limit the number of retries in both normal and space constraint relief steps:**
   SMS will limit the number of retries as previously stated, regardless of whether an allocation is in the normal volume selection phase or has gone on to the Space Constraint Relief phase.

4. **Select volumes by the amount of free space in the second best-fit step:**
   In prior releases, SMS best-fit volume selection selected volumes based on a combination of volume preference and amount of free space. Volumes with high preference ranking, but low free space, were sometimes preferred over volumes with low preference ranking but plenty of free space, which could result in the failure of best-fit volume selection. In z/OS DFSMS V1R8, SMS will re-rank the volumes by the amount of free space in the second best-fit phase, which is also the last phase, of Space Constraint Relief processing, if the first best-fit phase has failed to allocate sufficient space.

**Activating and displaying fast volume selection**

A new SMS parameter, **FAST_VOLSEL**, can be specified in the IGDSMSxx member to indicate whether the ‘fast’ approach should be used during SMS volume selection. It has two values, ON and OFF. The default is OFF. A new command, **SETSMS FAST_VOLSEL(ON | OFF)**, can be used to turn fast volume selection on or off. The command **D SMS,OPTIONS** will display the setting of the **FAST_VOLSEL** parameter.
Chapter 30. Using the Object Access Method enhancements

New z/OS DFSMS V1R8 Object Access Method (OAM) enhancements provide you with the following benefits:

- **Binary Large Object Support (Stage 1)**
  The OAM DB2 Binary Large Object Support enables objects larger than 32KB to be stored using DB2’s large object (LOB) support and the binary large object (BLOB) data type. The LOB environment is intended to be used for objects larger than 32KB continuing the need for our 4KB and 32KB tables. Today, these same large objects would be stored in multiple rows in the 32KB table. For example, a 256MB object requires 8000+ rows in a conventional 32KB table, but requires only one row when stored in a LOB storage structure in DB2. LOB support can be enabled at a global or storage group level.

- **Object Tape Enhancements (Stage 1)**
  The OAM Object Tape Enhancements Stage 1 consists of two functional enhancements. It will add automatic selection of RECYCLE-eligible tape volumes to the existing MOVEVOL with RECYCLE function and it will also add support for an immediate backup copy.

  - **Tape Recycle**
    This support will expand on the new RECYCLE option of the MOVEVOL utility to allow automatic selection of tape volumes for MOVEVOL with RECYCLE processing, based on installation-specified thresholds and limits.

  - **Immediate Backup Copy**
    This support provides a mechanism whereby OAM can now create the first backup copy of a given object at the time the object is originally stored. Before this support, the first and second backup copies are not created until the OSMC object storage management cycle is run (typically several hours after the object is stored).

    With the 3995 optical library being withdrawn from marketing, tape and the virtual tape server (VTS) will play an increasing role in our support. With 3592 WORM tape support, you may also want a backup copy available on WORM media soon after the primary copy is stored, plus there is added insurance in knowing that the backup copy was made sooner rather than later.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Application Programming” on page 134</td>
<td>• “Implementing the LOB function” on page 134</td>
</tr>
<tr>
<td></td>
<td>• “Implementing the automated selection of RECYCLE volumes function” on page 135</td>
</tr>
<tr>
<td></td>
<td>• “Implementing the immediate backup copy function” on page 135</td>
</tr>
</tbody>
</table>
After installing z/OS DFSMS V1R8, you can take advantage of the new functions in the following ways.

### Implementing the LOB function

In order to implement the LOB function with your application programs, complete the following steps:

1. Modify and run the CBRSMR18 migration job to add the new ODLOBFL column to the DB2 Object Directory tables for all object storage groups.

   **Note:** Whether or not you intend to exploit the new LOB function, this step is mandatory.

2. Use the LOB=A | P | N keyword on the OAM1 statement of the IEFSSNxx PARMLIB member.

   This keyword must be set to either LOB=A or LOB=P to indicate that DB2 LOB support should be used. The default is LOB=N, which disables future stores to the LOB storage structure. Note that objects already stored in the LOB storage structure can still be retrieved regardless of the LOB= keyword setting.

3. Use the supplied CBRILOB sample job to create a LOB storage structure for each object storage group that is to be LOB enabled.

   If you specify LOB=A on the OAM1 statement of the IEFSSNxx PARMLIB member, every object storage group is assumed to be LOB enabled, and therefore needs to have an associated LOB storage structure defined.

   Sample job, CBRILOB, is supplied to create the LOB storage structures required for OAM to exploit DB2 LOB support, including:
   - LOB tablespaces
   - Base tables
   - Base table views
   - Auxiliary tables
   - LOB indexes

4. Modify the sample job CBRPBIND.

   CBRPBIND performs a DB2 bind for the packages needed to access the OAM object storage group, OAM administration, and OAM configuration tables.

   CBRPBIND has been updated to use VALIDATE(RUN) instead of the VALIDATE(BIND) statement.

   Note that the package owner must have authorization to execute all statements embedded in the package for BIND PACKAGE to build a package without producing error messages. For VALIDATE(BIND), DB2 verifies the authorization at bind time. For VALIDATE(RUN), DB2 verifies the authorization initially at bind time, but if the authorization check fails, DB2 rechecks it at run time.

   Statements embedded within various OSR and OSMC packages now reference the V_OSM_LOB_BASE_TBL view, which is created as part of a LOB storage structure for a given object storage group by the CBRILOB sample job.

   However, these views might not be known to DB2 if either LOB support is disabled (LOB=N, default) or LOB support is only partially enabled (LOB=P). Each storage group that does not have the V_OSM_LOB_BASE_TBL created must use VALIDATE(RUN); otherwise, bind authorization occurs at bind time and the bind fails.
However, it might be desirable to use the VALIDATE(BIND) statement for each object storage group that you want to VERIFY that a V_OSM_LOB_BASE_TBL view has indeed been created. This is not required and should only be used as a precautionary measure to ensure the LOB view has been created. The bind fails if the LOB view is not created.

5. OAMplex support.

Prior to enabling the new support, if the user is running in an OAMplex, the system administrator must ensure all systems are capable of LOB support. In order for a pre-V1R8 level system to coexist in an OAMplex with a V1R8 level system or above, coexistence APAR OA12683 must first be applied to the pre-V1R8 level systems.

See z/OS Migration for more information.

**Implementing the automated selection of RECYCLE volumes function**

New keywords, MAXRECYCLETASKS, SGMAXRECYCLETASKS and PERCENTVALID are added to the SETOAM statement in the CBROAMxx member of PARMLIB in support of the new RECYCLE function added in this release. MAXRECYCLETASKS and SGMAXRECYCLETASKS provide the installation with the ability to specify a maximum number of tape volumes that can be recycled concurrently at the global and storage group level respectively. PERCENTVALID specifies a global default percent valid threshold to be used to identify RECYCLE candidate volumes in the event the optional PV= parameter is omitted from the F OAM,START,RECYCLE command invocation.

If the automated selection of RECYCLE volumes function is not to be exploited, there’s no changes necessary. If the SETOAM statements are added to the CBROAMxx member of PARMLIB and the system is backed down to a previous level of DFSMS, the changes to PARMLIB must also be removed, otherwise OAM address space initialization will fail when the new SETOAM statements are encountered.

**Implementing the immediate backup copy function**

To exploit the support of an immediate backup copy of objects, you should update or create the appropriate Management Class Definitions, or both. Existing Management Class Definition construct fields AUTO BACKUP (MCDAUTBK) and BACKUP FREQUENCY (MCDBKFQ) need to be updated to indicate when writing the first backup copy of an object. If the management class of the new stored object specifies auto backup = ‘Y’ and backup frequency = 0, OAM schedules the first backup copy to be written immediately after the primary copy of the object has been successfully stored.

If the immediate backup copy function is not to be exploited, no changes are necessary, and any backup copies will be made during the first storage management cycle after the object is stored, or during the first storage management cycle after a new management class is assigned for the object. To allow applications to know whether a STORE request resulted in scheduling an immediate backup copy of an object, the new optional RETCODE2 parameter can be coded when using the OSREQ STORE function.
Chapter 31. Using the DFSMSrmm enhancements

The functional enhancements available with z/OS V1R8 DFSMSrmm provide you with these benefits:

- **Enterprise level interface**
  The CIM provider has been moved from using the SNIA Java CIMOM to using the OpenPegasus implementation. This is a CIM WBEM implementation based on C++. The DFSMSrmm CIM provider shipped in z/OS V1R7, which enabled display and search of tape data sets and volumes, is extended to support creation, deletion, and update. It is also extended to cover more of the resources managed by DFSMSrmm as part of the CIM object model.
  In addition, a new command was added, SEARCHOWNER, as well as SMTP mail support.

- **DFSMSrmm UTC implementation**
  DFSMSrmm has the option to store all dates and times in common time and provides the base for viewing values in local, or common time. This ensures the consistency of information across multiple systems in an enterprise and across different time zones.

- **Tape data set authorization**
  The improvement of tape data set security through new options for authorizing access to tape data using the RACF® DATASET class profiles. For additional information, see Chapter 32, “Using Tape Data Set Authorization,” on page 141.

- **DFSMSrmm vital record specification policy management simplification**
  The simplification of data set retention using vital record specification policies by enabling data set name masks to be separated from the retention details. This helps to avoid duplication in vital record specification definitions. These changes enable DFSMSrmm vital record specification definitions to be better aligned to SMS ACS filters and SMS management classes.
  The new functions that are only supported with VRSEL(NEW) are:
  - Data set name filter vital record specifications with COUNT(0).
  - Retention NAME vital record specifications with COUNT(0).
  - Application of vital record specification release options for data sets not retained by vital record specification.
  - Use of JOBNAME(ABEND) or JOBNAME(OPEN) with a data set name mask.
  - Maintenance of the vital record specification last reference date and time.
  - Listing of unused vital record specifications in the REPORT file and unused counts in the MESSAGE file.

- **DFSMSrmm usability items**
  A collection of minor changes including:
  - New search options added to SEARCHVOLUME command.
  - How information is presented to the user via the DFSMSrmm ISPF dialog. These simplify the tasks performed by the storage administrator and simplify the analysis of data set and volume retention.
  - Within the DFSMSrmm ISPF dialog, a new SELECT primary command, as well as changes to the search results lists for data sets, volumes, and vital record specifications, were added.
  - Support added for ISPF/PDF option 3.4 (data set list) utility support.
The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning to use DFSMSrmm enhancements”</td>
<td>“Enterprise level interface”</td>
</tr>
<tr>
<td></td>
<td>“DFSMSrmm UTC implementation” on page 139</td>
</tr>
<tr>
<td></td>
<td>“DFSMSrmm vital record specification policy management simplification” on page 139</td>
</tr>
<tr>
<td></td>
<td>“DFSMSrmm usability items” on page 140</td>
</tr>
<tr>
<td>Operating</td>
<td>“Operating” on page 140</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>“Diagnosing” on page 140</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and installation tasks that you would perform to use the DFSMSrmm enhancements.

Planning to use DFSMSrmm enhancements

**Enterprise level interface**

**Before you begin:** Be familiar with information about the following topic:

- CIM providers; see [z/OS DFSMSrmm Implementation and Customization Guide](http://www.opengroup.org).
- SMTP mail. If you do not have an available SMTP server, you must create one. See [z/OS Communications Server: IP Configuration Guide](http://www.opengroup.org). You can use the DFSMSrmm OWNER = SMTP to identify the system on which the SMTP server is running. For additional details, see [z/OS DFSMSrmm Implementation and Customization Guide](http://www.opengroup.org).
- Modifying notify messages; see [z/OS DFSMSrmm Implementation and Customization Guide](http://www.opengroup.org).

**Software requirements:** The enterprise level interface function requires z/OS V1R8. A fully functional Web service as well as an xmlCIM compliant product that supports Java, such as the OpenPegasus C++ CIMOM from The Open Group, is required to use the DFSMSrmm CIM provider. A z/OS version of the OpenPegasus CIMOM is available under the Unix System Services. If running the CIM provider under Linux, the OpenPegasus CIMOM has to be downloaded from the OpenGroup’s web site at:

http://www.opengroup.org

Either way, the CIMOM must be up and running in order to work with the DFSMSrmm CIM provider.

For SMTP mail support, an SMTP mail server is required. The SMTP mail server needs to be either on the same system as DFSMSrmm is running or known via NJE node name.

Now you are ready to set up the enterprise level interface.
Setting up the enterprise level interface: Before you begin: The infrastructure to support the use of web services must be implemented and available on both the application system and the target z/OS system running DFSMSrmm.

Related reading: For more information about setting up the DFSMSrmm application programming interface via web services, see z/OS DFSMSrmm Application Programming Interface and z/OS DFSMSrmm Implementation and Customization Guide. For more information about setting up the DFSMSrmm Common Information Model (CIM) provider, see z/OS DFSMSrmm Implementation and Customization Guide.

DFSMSrmm UTC implementation
Before you begin: Toleration is provided for TZ structured field introducer and the TZ subcommand operand to enable you to update your DFSMSrmm-based applications and tools. The SYSAUTH.EDGTZ variable is ignored on lower level systems.

Toleration supports a mixed environment where some systems have toleration and the rest have z/OS V1R8 installed. Optionally, you can enable UTC(YES) when running coexistence.

For more information, see z/OS DFSMSrmm Implementation and Customization Guide.

Software requirements: The DFSMSrmm UTC implementation function requires z/OS V1R8.

DFSMSrmm vital record specification policy management simplification
Before you begin: Because the DFSMSrmm parmlib OPTION VRSEL(OLD) operand is planned to be removed in a future release, starting in z/OS V1R8, there is a new warning message issued when you run VRSEL(OLD) processing. If you do not migrate from VRSEL(OLD) to VRSEL(NEW) before moving to z/OS V1R8, you will get warning message EDG2317E each time VRSEL is run and EDGHSKP processing will end with job step return code 4.

A toleration APAR is provided to prevent the continued use of the vital record specification operands - STARTNUMBER and LOCATION(BOTH). The toleration APAR includes ++HOLD(ACTION) and requires that vital record specifications are cleaned up before they are able to run EDGHSKP with VRSEL. Also, as long as you do not implement COUNT(0) or JOBNAME(ABEND/OPEN), you can run VRSEL processing on either a toleration system or z/OS V1R8. If you mistakenly run VRSEL processing on a lower level or toleration level system, any data sets matching a vital record specification that includes COUNT(0) is retained by vital record specification, but a null retention date is set.

Starting in z/OS V1R8, any vital record specifications that includes JOBNAME(OPEN) or JOBNAME(ABEND) in its data set name vital record specification are used as policies for retaining data sets that are either OPEN or have been closed by ABEND processing. If you want to continue to use existing vital record specifications to match to real job names of either ABEND or OPEN, you must change the vital record specification to use JOBNAME(OPEN*) or JOBNAME(ABEND*). If you do not do this, DFSMSrmm uses the existing vital record specifications for ABEND and OPEN retention policies.

Use the EDGRVCLN REXX procedure to clean up name vital record specifications. Use the LIST(FILTER) parameter to analyze the existing vital record specifications.
and also to see how many different types of policies you have, and use the FIX(FILTER) parameter to implement the changes recommended by LIST(FILTER).

You can specify that a vital record specification is not to retain a data set by use of COUNT(0). Using COUNT(0) in a vital record specification implies that the vital record specification in question cannot retain a data set, and vital record selection processing continues with the next vital record specification in the chain, if any. In the same way, when you specify COUNT(0) as part of an ANDVRS group, the ANDVRS group cannot retain a data set.

For more information, see z/OS DFSMSrmm Managing and Using Removable Media.

**Software requirements:** The DFSMSrmm vital record specification policy management simplification function requires z/OS V1R8.

**DFSMSrmm usability items**

**Before you begin:** To enable direct entry into the DFSMSrmm ISPF dialog from the ISPF Data Set List Utility, use the ISPF Configuration Utility to update the ISPF Configuration Table.

**Note:** When you use ISPF LIBDEF, whether for normal entry to the DFSMSrmm dialog or from the ISPF DSLIST utility, or from ISMF, you must copy or make available the EDGRMLIB and the EDGRPD34 execs in an exec library normally available to ISPF users.

For additional information, see z/OS DFSMSrmm Implementation and Customization Guide.

**Software requirements:** The DFSMSrmm usability items requires z/OS V1R8.

---

**Operating**

To run the DFSMSrmm application programming interface via web services, you must be running WebSphere Application Server or an equivalent. To run the CIM provider, you must be running an xmlCIM compliant product that supports Java, such as the OpenPegasus C++ CIMOM from The Open Group.

**Before you begin:** For additional information, see z/OS DFSMSrmm Implementation and Customization Guide.

---

**Diagnosing**

For information on diagnosing DFMSrmm problems, see z/OS DFSMSrmm Diagnosis Guide.
Chapter 32. Using Tape Data Set Authorization

Using the tape data set authorization enhancement, OPEN and EOV have the capability to always issue RACROUTE in the DATASET class to enable tape data sets to be authorized in the same way that DASD data sets are authorized. The RACROUTE is issued regardless of the RACF SETROPTS options in use on the system or the label types used. The difference between this capability and the use of the TAPEDSN RACF option is that the RACROUTE is always issued, that DSTYPE=T is not used, and STATUS=ERASE is specified. You request the new function using a new DEVSUPxx option.

The new function in O/C/EOV is not intended to replace all of the functional capabilities that the RACF TAPEDSN option, TAPEVOL class, and TVTOCs provide. However, together with the existing and new functions provided by DFSMSrmm, you do have equivalent capability. The O/C/EOV function does address the authorization requirements for tape data sets and relies on your use of a tape management system, such as DFSMSrmm, to verify the full 44 character data set names, control the overwriting of existing tape files, handle tape data set retention, and control the creation and destruction of tape labels.

A new DEVSUPxx option is provided to request that the user’s authorization to file 1 on the volume is authorized, as well as the file being opened. The RACROUTE parameters used depends on the DEVSUPxx keywords and the SETROPTS options in use. When all the files on a tape volume have a common or similar authorization, there is less chance that an application program will gain access to unauthorized data by repositioning the tape to another file. Tape management system support for this function is required and is provided by DFSMSrmm.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and Installation</td>
<td>• “Planning to use tape data set authorization” on page 142</td>
</tr>
<tr>
<td></td>
<td>• “Setting up tape data set authorization” on page 142</td>
</tr>
<tr>
<td>Administering</td>
<td>• “Implementing tape data set authorization” on page 143</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and installation tasks that you would perform to use tape data set authorization.

Planning to use tape data set authorization

Before you begin: To understand the concepts of tape data set authorization, read the Protecting Data Sets on DASD and Tape chapter of z/OS Security Server RACF Security Administrator’s Guide. This chapter includes sections called Choosing Which Tape-Related Options to Use and Tape Data Set and Tape Volume Protection (TAPEDSN Active and TAPEVOL Active).
**Software requirements:** The tape data set authorization function requires z/OS V1R8.

Enabling this new function varies depending on how you provide tape security today. There are so many variations, including the use of customized RACF installation exits, that this information is for guidance only.

The following are reasons for why you might consider using the tape data set authorization function:

- You no longer need to have the TAPEVOL class active and maintain access lists for volumes.
- You are using TVTOCs today, but find that limitations in profile sizes and numbers of data sets and volumes supported in a multi-volume sets often cause problems.
- You have customized RACROUTE REQUEST=DEFINE and RACROUTE REQUEST=AUTH installation exits to base tape data set access on DATASET profiles.
- You use TAPEDSN only, but would also like to exploit the File 1 authorization checking.

Now you are ready to set up tape data set authorization.

**Setting up tape data set authorization**

**Before you begin:** You should plan to migrate to a system environment where the following are in use:

- DEVSUPxx options:
  - TAPEAUTHDSN=YES
  - TAPEAUTHF1=YES
  - TAPEAUTHRC4=FAIL
  - TAPEAUTHRC8=FAIL
- DFSMSrmm parmlib option:
  - OPTION TPRACF(N)
- RACF options:
  - SETROPTS NOTAPEDSN NOCLASSACT(TAPEVOL)

Use the following to get to this system environment:

**OPTION TPRACF(CLEANUP)**

DFSMSrmm aids in the deletion of existing TAPEVOL profiles and discrete DATASET profiles as tape volumes are recycled to scratch status.

Do not use this DFSMSrmm option unless you have successfully implemented TAPEAUTHDSN.

**TAPEAUTHRC8=WARN**

Use to implement a warning level of authorization checking for tape data sets to enable you to adjust and correct your generic profiles in the DATASET class to handle tape data set authorization as well.

You should only need to use the WARN keyword value if you are using TAPEVOL class without the TAPEDSN option. If you use TAPEDSN today, you should already have the required DATASET class profiles to support the authorization checking for all tape data sets.
Once you are through with your testing and introduction period, you should change to TAPEAUTHRC8=FAIL for production use.

**TAPEAUTHRC4=ALLOW**

Use to allow access to unprotected tape data sets while you determine which new DATASET profiles are required.

Once you are through with your testing and introduction period, you should change to TAPEAUTHRC4=FAIL for production use.

**Related reading:** For more information, see [z/OS MVS Initialization and Tuning Guide](#).

You now have the choice of implementing tape data set authorization checking without consideration for whether the data set is on tape or DASD and without consideration of tape volume profiles. You now have the following options to protect data on tape:

- Use the TAPEAUTHDSN keyword to cause RACROUTE to be issued in the DATASET class just as if it was for a DASD data set. This optionally allows authorization checking for file 1 and supports the use of RACF 'erase on scratch' support by DFSMSrmm.
- Use SETROPTS TAPEDSN to cause RACROUTE to be issued in the DATASET class with DSTYPE=T. This optionally allows the use of TAPEVOL profiles with or without TVTOCs. This optionally allows authorization checking for file 1.
- Use tape volume authorization only via TAPEVOL profiles.

For a complete list of the existing options, refer to [z/OS Security Server RACF Security Administrator’s Guide](#).

### Administering

For tape data set authorization, these are the system programming tasks.

#### Implementing tape data set authorization

In order to implement tape data set authorization, consider the following:

1. **DFSMShsm considerations:**
   - This affects you if you use the SETSYS TAPESECURITY values of either RACF or RACFINCLUDE and have the TAPEVOL class active.

   Prior to activation of the new function, you should ensure that there are generic profiles created in the DATASET class that cover each of the data set name prefixes that DFSMShsm uses.

   DFSMShsm continues to maintain the TAPEVOL profiles for HSM and ABARS tapes until you inactivate the TAPEVOL class. Once the new function is activated and you have inactivated the TAPEVOL class, you can delete the special HSMABR, HSMHSM and DFHSMx tape profiles. You also should change the SETSYS TAPESECURITY value.

   For additional information, see [z/OS DFSMShsm Implementation and Customization Guide](#).

2. **IEHINITT considerations:**
If you plan on using IEHINITT to re-label tape volumes (note that this excludes initializing brand new volumes) and you want to provide protection for labelling and control which users that can do this, you should have the TAPEVOL class active and define TAPEVOL profiles. When you use the TAPEAUTHDSN DEVSUPxx option, or use TAPEDSN SETROPTS option to protect tape data sets, you can define one or more generic TAPEVOL profiles to cover all volumes.

If you use the DFSMSrmm EDGINERS tape labelling and erasing utility, you do not need to have the TAPEVOL class active. DFSMSrmm controls who can use the EDGINERS utility, and you can only label or erase volumes that are either new to DFSMSrmm, or have the INIT or ERASE action pending.

For additional information, see z/OS DFSMSdfp Utilities.

3. DFSMSrmm considerations:

Use OPTION TPRACF(CLEANUP) so that DFSMSrmm aids the deletion of existing TAPEVOL profiles and discrete DATASET profiles as tape volumes are recycled to scratch status. Do not use this DFSMSrmm option unless you have successfully implemented TAPEAUTHDSN.

If a specifically requested volume is authorized to be ignored by DFSMSrmm, and the installation exit EDGUX100 requested the volume be ignored, and the RACROUTE issued by OPEN indicates either not protected (RC4) or not authorized (RC8), DFSMSrmm sets the SAF return code to zero to allow access to the volume. This support is provided independently of the new TAPEAUTHxxx DEVSUPxx options.

For TAPEAUTHDSN=YES, DFSMSrmm tracks the ERASE on scratch decision. This means that when EDGINERS is run, it should also be requested to handle the ERASE action as well as the INIT action so that as tapes return to scratch, DFSMSrmm ensures they are erased.

For additional information, see z/OS DFSMSrmm Implementation and Customization Guide.
Chapter 33. Using the DFSMSHsm usability enhancements

In z/OS V1R8, DFSMSHsm is enhanced with a collection of serviceability, reliability and usability improvements. These changes are designed to improve DFSMSHsm performance when writing to tape, and give you more flexibility in recovering data sets from aggregate backups.

The DFSMSHsm enhancements are described, as follows:

- **Improved error handling for tape duplexing of migration data sets**
  
  If you use tape duplexing for migration data sets, you can now specify that both the original and alternate tapes are to be marked full if the alternate tape volume encounters an error. Previously, this error handling applied only for errors on the original tape volume. This option helps you to ensure that the original and alternate tapes remain synchronized, and allows you to obtain duplex copies without significant delay.

  This option applies to migration tapes, for both initial migrations and subsequent RECYCLE processing. The option is not available for backup tapes.

  This option is recommended for installations that use higher capacity tapes and perform tape duplexing for disaster recovery purposes. If you choose not to use this option, error handling for alternate tapes remains unchanged from previous releases. That is, if the alternate tape volume encounters an error, the alternate is demounted and returned to scratch, but DFSMSHsm continues writing to the original. When the writing is completed, DFSMSHsm schedules a TAPECOPY request to create a new alternate tape.

- **Ability to recover individual data sets from an aggregate group backup**
  
  The ARECOVER command now allows you to specify that individual data sets are to be recovered from the backup copy of an aggregate group. New parameters are added to the ARECOVER command and to ISMF panels to allow the recovery of a single fully-qualified data set name or a list of fully-qualified data set names.

- **Improved performance for RECYCLE processing**
  
  In this release, DFSMSHsm improves its performance for RECYCLE processing of tapes. RECYCLE now uses SYNCDEV requests more efficiently, which allows tape processing to complete more quickly.

- **Faster migration of non-VSAM data sets to tape**
  
  In this release, DFSMSHsm improves its performance for migrating non-VSAM data sets to tape. Scratch requests are now handled asynchronously, which allows other migration work to complete more quickly.

- **Ability to alter priority of queued requests**
  
  In this release, DFSMSHsm adds a new command, ALTERPRI, which alters the priority of certain queued requests. Using this command, you can alter the priority of queued requests by the request number, by the userid which issued the request, or by the data set named in the request. You can alter the priority to make the request either the highest or lowest priority in its queue.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.
Planning and Installation

This section describes the planning and installation tasks that you would perform for the DFSMShsm usability enhancements.

Planning to use the DFSMShsm usability enhancements

Before you begin: Be familiar with the following topics:

- Tape duplex option; see z/OS DFSMShsm Storage Administration
- Performing aggregate recovery; see z/OS DFSMShsm Storage Administration

Related reading:

- For information about making disaster backup copies of DFSMShsm-owned tape volumes, see z/OS DFSMShsm Storage Administration
- For information about aggregate backup and recovery (ABARS), see z/OS DFSMShsm Storage Administration
- For information about Interactive Storage Management Facility (ISMF), see z/OS DFSMS Using the Interactive Storage Management Facility
- For information about altering the priority of queued requests with the ALTERPRI command, see z/OS DFSMSdfp Storage Administration

Software requirements: The new function name requires z/OS V1R8.

Coexistence requirements: None.

Perform the following steps to plan for using the new function:

1. Determine how your installation should handle error conditions that occur on the alternate tape during duplexing of migration tapes. If your installation duplexes migration tapes for disaster recovery purposes and uses higher capacity tapes, it is recommended that you specify the ERRORALTERNATE(MARKFULL) option on the SETSYS DUPLEX(MIGRATION) command. If you choose not to use this option, error handling for alternate tapes remains unchanged from previous releases.

2. If your installation needs to recover individual data sets from the backup copy of an aggregate group, verify that the data sets to be recovered reside on the
correct aggregate version. To do so, check the ABACKUP activity logs or FILTER OUTPUT data set that is associated with the particular version being recovered.

Now you are ready to use the DFSMSHsm enhancements.

**Administering**

To recover individual data sets from the backup copy of an aggregate group, you can use either of the following methods:

- "Using the ARECOVER command to recover individual data sets from an aggregate backup"
- "Using ISMF to recover individual data sets from an aggregate backup" on page 148.

**Related reading:**
- For information about using the ARECOVER command, see z/OS DFSMSdfp Storage Administration
- For information about using ISMF to work with aggregate groups, see z/OS DFSMSdfp Storage Administration.

**Using the ARECOVER command to recover individual data sets from an aggregate backup**

To recover an individual data set from an aggregate backup version, use the ARECOVER command, as shown in the examples that follow.

**Recovering an individual data set from an aggregate backup**

Perform the following steps to recover an individual data set from the backup copy of an aggregate group:

1. Enter the ARECOVER command with the ONLYDATASET parameter specified.
   
   **Example:**
   
   ARECOVER ONLYDATASET(NAME(MY.OWN.DATASET))

   In this example, MY.OWN.DATASET is the fully qualified name of the data set to be recovered from the aggregate version.

2. **Recovering a group of data sets from an aggregate backup**

   Perform the following steps to recover a group of data sets from the backup copy of an aggregate group:

   1. Create a data set to use as the list of data sets to be recovered during ARECOVER processing. The data set must be a sequential data set, fixed-block, with a record size of 80 and be cataloged.
   2. Edit the data set with the names of data sets to be recovered during ARECOVER processing. Specify one fully qualified data set name per record.
   3. Enter the ARECOVER command with the LISTOFNAMES subparameter of the ONLYDATASET parameter specified.
      
      **Example:**
In this example, `MY.DATASET.LIST` is the fully qualified name of the data set that contains a list of data sets to be recovered from the backup copy of an aggregate group.

**Using ISMF to recover individual data sets from an aggregate backup**

To recover an individual data set from an aggregate backup version, you can use ISMF, as described in this section.

Perform the following steps to recover an individual data set from the backup copy of an aggregate group:

1. **Access the ISMF Primary Option Menu for Storage Administrators**, as shown in z/OS DFSMSdfp Storage Administration.

2. Select option 9, Aggregate Group, and press Enter. The Aggregate Group Application Selection panel is displayed.

3. Select option 6, ARECOVER, and press Enter to view the Aggregate Group Recover panel.

4. Specify the values for the aggregate group. Specify Y for the field "Recover Individual Dataset." Press Enter to view the Aggregate Group Name panel.

5. On the Aggregate Group Name panel, specify either a single data set name or the name of a data set that contains a list of data sets to be recovered from the backup copy of an aggregate group.

   To create a list of data sets, use a fixed-block, sequential data set with a record size of 80 and catalog it. Edit the data set and specify one fully qualified data set name per record.

6. Enter the END command to submit the recover request.

**Altering the priority of queued requests**

To alter the priority of queued requests for some DFSMSHsm functions, you can use the ALTERPRI command. This command is security protected, so a security profile must be defined and users must be given access to it before the ALTERPRI command can be issued. For information about defining FACILITY class profiles, see z/OS Security Server RACF Security Administrator’s Guide.

Perform the following steps to define a security profile for the ALTERPRI command:

1. Define the ALTERPRI profile as follows: `STGADMIN.ARC.ALTERPRI`.

2. Give the appropriate users access to the `STGADMIN.ARC.ALTERPRI` profile.
3. Activate the FACILITY class before starting DFSMShsm.

When started, DFSMShsm checks to determine whether the FACILITY class is active. If you have not defined the new RACF profile STGADMIN.ARC.ALTERPRI and provided the appropriate user access, and the generic RACF profile STGADMIN.ARC* access is set to NONE, then all ALTERPRI commands will fail.

Operating

You can control error handling for alternate duplex tapes through the following DFSMShsm operator command:

- SETSYS DUPLEX(MIGRATION) command to have migration tapes marked full for errors.

**Related reading:** For information about using DFSMShsm operator commands, see z/OS DFSMSdfp Storage Administration.

Marking migration tapes full for errors

To have alternate and original migration tapes marked full if an error occurs when writing to the alternate tape volume, enter the SETSYS DUPLEX(MIGRATION(Y)) command with the ERRORALTERNATE(MARKFULL) parameter specified:

**Example:**

```
SETSYS DUPLEX(MIGRATION(Y ERRALT(MARKFULL)))
```

The default for ERRORALTERNATE is CONTINUE, which leaves error handling unchanged from previous releases. That is, if the alternate tape volume encounters an error, the alternate is demounted and returned to scratch, but DFSMShsm continues writing to the original. When the writing is completed, DFSMShsm schedules a TAPECOPY request to create a new alternate tape.

Altering the priority of queued requests

The ALTERPRI command lets you alter the priority of certain queued requests. The following request types can be reprioritized: ABACKUP, ARECOVER, BACKDS, BACKVOL, DELETE, FRBACKUP, FREEVOL, FRRECOV, MIGRATE, RECALL, and RECOVER. Note: you cannot reprioritize BACKVOL CDS commands and requests that have already been selected for processing. Use the ALTERPRI command to alter the priority of queued requests on an as-needed basis. The command should not be used as the primary means of assigning priority values to new requests.

You can alter the priority of queued requests by the request number, by the userid which issued the request, or by the data set named in the request. You can alter the priority to make the request either the highest or lowest priority in its queue.

To alter all requests for a particular data set so that they have the highest priority on their respective queues, enter the ALTERPRI command with the DATASETNAME parameter specified:

**Example:**

```
ALTERPRI DATASETNAME(dsname)
```
To alter all requests with a particular request number so that they have the highest priority on their respective queue, enter the ALTERPRI command with the request number specified:

**Example:**

```
ALTERPRI REQUEST(reqnum) HIGH
```

To alter all requests issued by a particular user so that they have the lowest priority on their respective queues, enter the ALTERPRI command with the userid specified:

**Example:**

```
ALTERPRI USERID(userid) LOW
```
Chapter 34. Using large format data sets

Before z/OS V1R7, most sequential data sets were limited to 65,535 tracks on each volume, although most hardware storage devices supported far more tracks per volume. To support this hardware capability, z/OS V1R7 allows users to create new large format data sets, which are physical sequential data sets with the ability to grow beyond the previous size limit. QSAM, BSAM, and EXCP access methods all support large format data sets, with some limitations for EXCP programs, and for BSAM programs that use the NOTE and POINT macros. Large format data sets reduce the need to use multiple volumes for single data sets, especially very large ones like spool data sets, dumps, logs, and traces. Large format data sets can be either cataloged or uncataloged, SMS-managed or not. Unlike extended-format data sets, which also support greater than 65,535 tracks per volume, large format data sets are compatible with EXCP and don’t need to be SMS-managed.

For programs that use EXCP, or BSAM with the NOTE or POINT macros, some source code changes are required in order to use large format data sets. These programs must use some new interfaces to allow for larger track numbers, and must specify the BLOCKTOKENSIZE=LARGE parameter on the DCBE macro. See “Planning and installation” and “Application programming” on page 156 for more information about the required changes.

The following table lists the types of tasks and associated procedures that you must complete to use large format data sets.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and installation”</td>
<td>• “Planning to use large format data sets”</td>
</tr>
<tr>
<td></td>
<td>• “Setting up support for large format data sets” on page 155</td>
</tr>
<tr>
<td>“Administering” on page 155</td>
<td>• “Creating large format data sets” on page 156</td>
</tr>
<tr>
<td>“Application programming” on page 156</td>
<td>• “Modifying applications to use large format data sets” on page 157</td>
</tr>
<tr>
<td>“Diagnosing problems with large format data sets” on page 158</td>
<td></td>
</tr>
</tbody>
</table>

Planning and installation

This section describes the planning and installation tasks that you must perform to use large format data sets.

Planning to use large format data sets

Before you begin: To understand the concepts of sequential data sets in general, and large format data sets in particular, see z/OS DFSMS Using Data Sets. To understand how to use IGDMSxx parameters, see z/OS MVS Initialization and Tuning Reference.

Software requirements: Large format data sets require z/OS V1R7 and the supplied PTFs.
Coexistence requirements: On a z/OS system lower than V1R7, programs can open large format data sets only for input and only if the data set has no more than 65,535 tracks on each volume. Jobs that specify a DSNTYPE value of LARGE or BASIC on a pre-V1R7 system will fail. If an SMS active configuration data set is shared between a z/OS V1R7 system and a lower-level system, and a data class has a DSNTYPE value of LARGE or BASIC, the lower-level system will ignore the value. Also, an IGDSMSxx member of SYS1.PARMLIB that contains the BLOCKTOKENSIZE parameter cannot be shared with a lower-level system. Note: Services on lower-level systems that display data set information, such as ISPF, ISMF, IEHLLST and LISTDSI, may show incorrect amounts of space used by large format data sets.

Supported environments: z/OS V1R7 supports large format data sets in many areas, including standalone dump, IPCS, the JES2 and JES3 spools, ISPF, and the TSO/E ALLOCATE command. SMS, DFSORT, IDCAMS REPRO, IEBGENER, IEBDG, and IEBCOPY all support large format data sets and are unaffected by the BLOCKTOKENSIZE option in SYS1.PARMLIB. DFSMSshm supports large format data sets for user data sets and for the journal, and DFSMSrmm supports them for the journal and as temporary data sets for inventory management. z/OS components that don’t support large format data sets in release V1R7 include virtual I/O, system dump, password data sets, the BDAM access method, and the TSO COPY command. LISTDSI and the equivalent REXX function may display the wrong size value. The high level languages do not support large format data sets when the new BLOCKTOKENSIZE(REQUIRE) option in the IGDSMSxx member of PARMLIB is in effect. If BLOCKTOKENSIZE(NOREQUIRE) is in effect, then generally you can expect programs that use QSAM or use BSAM without NOTE or POINT to work. Whether they work depends on factors such as whether they use the DS1LSTAR field, the DEBNMTRK field or call a track conversion routine. See below for details.

Perform the following steps to plan for using large format data sets:

1. Examine programs that use the following 16-bit track number fields. These programs will need to be modified to use larger fields (see “Application programming” on page 156 for modification steps) to accommodate the number of tracks that large format data sets might use. These programs must also specify BLOCKTOKENSIZE=LARGE on the DCBE macro to signify that they are using the larger track number fields:
   • Programs that use or examine the DS1LSTAR field in the format 1 DSCB and calculate space usage.
   • Programs that use or examine the DEB to determine the size of each extent, or examine the DEBNMTRK field (tracks in extent). This applies to BSAM, QSAM, and EXCP programs.
   • Programs that use track conversion routines to convert between a relative track and record value (TTR) format and a MBBCCHHR format of disk addresses. See z/OS DFSMSdfp Advanced Services for information about new entry points to the CVTPRVT and CVTPCNVT routines that return track values in the larger TTTR format. This applies to all EXCP programs, and to BSAM and QSAM programs that use the track conversion routines.
   • Programs that use the JES spool with documented interfaces should not need changes.

2. Also, plan to add BLOCKTOKENSIZE=LARGE on the DCBE macro to any EXCP programs and to BSAM programs that use NOTE or POINT macros. See “Application programming” on page 156 for details.
Now you are ready to set up your support for large format data sets.

**Setting up support for large format data sets**

**Before you begin:**

**Related reading:** For more information, see “Initializing SMS through the IGDSMSxx member” in the [z/OS DFSMSdfp Storage Administration](https://www.ibm.com/support/knowledgecenter/SSS7ND_1.10.0/com.ibm.dfsmsdfp.sa/dfsmsdfp_sa.pdf) and [z/OS MVS Initialization and Tuning Reference](https://www.ibm.com/support/knowledgecenter/SSS7N_2.5.0/com.ibm.zos.zos/rzomvsti.pdf).

Perform the following steps to set up support for large format data sets:

1. Decide whether to code BLOCKTOKENSIZE(REQUIRE) in the IGDSMSxx parmlib member.

   When BLOCKTOKENSIZE(REQUIRE) is specified, most applications that use large format data sets must specify BLOCKTOKENIZE=LARGE on the DCBE macro, signifying that the application has made any needed changes to support the data sets. (For details on the application changes that may be required for large format data sets, see “Application programming” on page 156.) The only exception to this rule is if the data set has no more than 65,535 tracks on each volume and is open for input using EXCP, BSAM, or QSAM, or for update using BSAM or QSAM.

   You can specify BLOCKTOKENIZE(NOREQUIRE) in IGDSMSxx to allow applications to access large format data sets under more conditions without having to signify BLOCKTOKENIZE=LARGE on the DCBE macro. The applications and data sets must meet any of the following conditions:
   - The access method is QSAM or it is BSAM without the NOTE or POINT macros.
   - The access method is BSAM with the NOTE or POINT macros (MACRF=xP is coded) and the data set has no more than 65,535 tracks on the volume and the OPEN option is INPUT or UPDAT.
   - The access method is EXCP (MACRF=E is coded) and the data set has no more than 65,535 tracks on the volume and the OPEN option is INPUT.

2. Based on the types of applications that might access large format data sets on your system, and their likely need for updates, either code BLOCKTOKENIZE(REQUIRE) in the IGDSMSxx PARMLIB member or choose the less restrictive option of BLOCKTOKENIZE(NOREQUIRE).

**Note:** In z/OS V1R7, BLOCKTOKEN(REQUIRE) is the default value that takes effect when BLOCKTOKENIZE is not specified in IGDSMSxx. If you expect that you will want the value of REQUIRE in releases after V1R7, IBM recommends that the system programmer set BLOCKTOKENIZE(REQUIRE) in z/OS V1R7. In z/OS V1R8, the default value is changed to BLOCKTOKENIZE(NOREQUIRE).

**Administering**

In addition to large format data sets, in z/OS V1R7 there are other new values that you can specify for the DSNTYPE value on the DD statement and the dynamic allocation equivalent. These values do not designate new types of data set. In addition the storage administrator might want to modify the SMS ACS routines to recognize new values for the &DSNTYPE variable.
Creating large format data sets

To create a large format data set, specify a DSNTYPE value of LARGE on a DD statement, TSO/E ALLOCATE command, IDCAMS ALLOCATE, or dynamic allocation (SVC 99). You can also assign the LARGE value to an SMS data class, which will take effect (for both SMS and non-SMS managed data sets) as the data set type if no other DSNTYPE is specified on the data set allocation. The large format data set must have a DSORG value of PS or PSU, or have no DSORG specified on the allocation. The DSNTYPE of LARGE becomes a permanent attribute of the associated DSCB (bit DS1Large in DS1FLAG1 field). You cannot add or remove the attribute for an existing data set; to change a data set to or from large format, the data set would need to be deleted and recreated.

Creating basic format data sets

z/OS V1R7 introduces a new name for an existing data set type. The term "basic format data set" designates a sequential data set that is neither extended format nor large format. It cannot occupy more than 65,535 tracks per volume. Normally the DSORG value is PS, PSU or omitted. You can specify DSNTYPE=BASIC on a DD statement, TSO/E ALLOCATE command, IDCAMS ALLOCATE, or dynamic allocation (SVC 99). With the DD statement or dynamic allocation, you can use BASIC to override a data class’s DSNTYPE of extended or large format. Conversely, you can set a data class to BASIC by default, by omitting DSNTYPE in the data class definition (you can’t explicitly code a value of BASIC for DSNTYPE in a data class).

Specifying a DSNTYPE of EXTREQ or EXTPREF

z/OS V1R7 also accepts new DSNTYPE values of EXTREQ and EXTPREF on the DD statement, TSO/E ALLOCATE command, IDCAMS ALLOCATE, and dynamic allocation. Previously, these values could only be specified in the data class, as P (preferred) and R (required) for a data set type of EXT. The DSNTYPE of EXTREQ specifies that the allocation is required to be in extended format, and should fail if that is not possible. EXTPREF specifies that allocation is preferred to be in extended format, but if the necessary system resources for extended are not available, then the data set should be allocated as BASIC.

New values for DSNTYPE keyword in ACS routines

The &DSNTYPE read-only variable passed to ACS routines can have new possible values of LARGE and BASIC. These values might be set by the user coding them, the LIKE data set having them, or the data class supplying them. The DSNTYPE variable can also have a value of EXR caused by the user coding DSNTYPE=EXTREQ on the DD statement, or a value of EXP caused by the user coding DSNTYPE=EXTPREF on the DD statement. If the data class has no value for DSNTYPE, the effect is that of a value of BASIC, although BASIC cannot be specified in the data class definition.

Application programming

After you have enabled support for large format data sets, you might need to modify some applications to work with them.
Modifying applications to use large format data sets

Applications can use the QSAM, BSAM, and EXCP access methods to access large format data sets. In many cases, depending on the access method and functions that the application uses, modifications are required. Take the following steps to modify your applications as appropriate:

1. Specify the BLOCKTOKENSIZE=LARGE parameter on the DCEB macro.
   This parameter signifies that the application is prepared to access large format data sets, and includes all the following modifications that apply. By system default in z/OS V1R7 — unless the IGDSMSxx member of SYS1.PARMLIB specifies BLOCKTOKENSIZE(NOREQUIRE) — this parameter is required in order to open a large format data set except when the OPEN is for input only and the data set contains no more than 65 535 tracks on each volume. If the system programmer has activated the IGDSMSxx member of SYS1.PARMLIB with BLOCKTOKENSIZE(NOREQUIRE), then programs that do not have a DCBE with BLOCKTOKENSIZE=LARGE can work under the conditions described earlier in “Setting up support for large format data sets” on page 155.

2. Change BSAM programs that use the NOTE or POINT macros.
   The BSAM NOTE and POINT macros normally use a four-byte field for the relative track and record number of a data block, in the format TTR0. To accommodate longer track numbers for large format data sets, programs that use these macros must specify BLOCKTOKENSIZE=LARGE on the DCBE macro; this causes the macros to provide and accept the track and record values in the format TTTR. The POINT macro also has a new RELNEXT parameter to position the data set after the record indicated in the NOTE value for large format data sets. Update the application program to these new track value formats and macro parameters as appropriate. If you update your program in this way and run it on z/OS V1R7 or later, then it also can run with any other DASD data set that is not large format and supports NOTE and POINT. IBM recommends making this change so that your programs will support the maximum number of types of data set. For more information, see the NOTE and POINT macro descriptions in z/OS DFSMS Macro Instructions for Data Sets.

3. Change programs that use or examine the DS1LSTAR field in the format 1 DSCB.
   The first two bytes of the DS1LSTAR field of the DSCB contains the track number of the last track used. If the DS1LARGE bit is on, then the data set is large format and the third (high-order) byte of the last-used track number is in the new byte DS1TTTHI at offset X’68’ in the DSCB. Update the application program to use or examine this new field as well.

4. Change programs that use or examine the DEBNMTRK field in the data extent block (DEB) for an EXCP, BSAM, or QSAM DCB.
   The DEBNMTRK field of the DEB contains the number of tracks in the extent. This is a two-byte field; for large format data sets, an additional high-order byte of the track number (if any) may be contained in the DEBNmTrkHi field. Update the application program to use or examine this additional field as well.

5. Change QSAM, BSAM, BPAM or EXCP programs that use track address conversion routines CVTPRLTV and CVTPCNVT.
   These routines convert between absolute and relative track and record values, using a three-byte TTR0 format prior to z/OS V1R7. For large format data sets, these routines have new entry points that use three bytes for the track number, in TTTR format. The new entry points work also with all data sets that the old entry points work with. For details about the changes to these routines, see z/OS DFSMSdfp Advanced Services and update any application programs that may use them for large format data sets.
6. Change programs that calculate DASD space usage.

Programs that calculate DASD space usage should check the new DS1Large bit (X'08') in the DSFLAG1 field of the DSCB, which is set when a large format data set is allocated. If the bit is on, the program should allow for the possibility that any extent except for a user label extent might exceed 65535 tracks. A program that scans the output of IEHLIST LISTVTOC might not work as expected because the format of DS1LSTAR output was changed slightly. See z/OS DFSMSdfp Utilities.

7. Change programs that examine SMF type 14 and 15 records.

The SMF type 14 and 15 records have some slightly changed information for large format data sets. These two records have a field that contains the number of tracks released by DADSM for RLSE. For a sequential data set, SMF14NTU is a four-byte field that contains the relative track in TTR0 format of the last processed record. The first bit in the SMF14FG1 field is an indicator that SMF14NTU is in the format of TTTR instead of TTR0. The bit name is SMF14LGE.

**Diagnosing problems with large format data sets**

The following ABEND codes indicate that an OPEN or end-of-volume operation failed for a large format data set:

- 213-10
- 213-14
- 213-15
- 213-16
- 213-17
- 737-44
- 737-45

**Related reading:** For information about the 213 ABEND, see the description of message IEC143I in z/OS MVS System Messages, Vol 7 (IEB-IEE). For ABENDs 737-44 and 737-45, see the explanation for message IEC027I.
Chapter 35. Using the device support address space

z/OS now provides a new device support address space (DEVMAN). It is started during an IPL, is cancelable, non-swapable, and restartable with the START command. z/OS uses the new address space to:

- Capture CTRACE information for CVAF events for first failure data capture.
- Capture CTRACE information for DADSM Create events for first failure data capture.

The started device support address space will be identified in the system as DEVMAN. This name will be seen in system displays and will be used as the target address space name for the Modify commands.

The user tasks required to exploit the new device support address space are:

- Restarting DEVMAN (with the START command) when cancelled.
- Starting the component trace with the TRACE CT,ON,COMP=SYSDMO command and responding to the prompt for operands with OPTIONS=(DADSM1,CVAF1),END.
- Enabling the component trace with external writer by:
  - Starting CTRACE output writer with the TRACE CT,WTRSTART=CTWTR command
  - Starting the component trace with the TRACE CT,ON,COMP=SYSDMO command and responding to the prompt for operands with WTR=CTWTR,OPTIONS=(CVAF1,DADSM1),END.
- Stopping the component trace with the TRACE CT,OFF,COMP=SYSDMO command and responding to the prompt for operands with OPTIONS=(DADSM0,CVAF0),END.
- Stopping the CTRACE output writer with the TRACE CT,OFF,COMP=SYSDMO and TRACE CT,WTRSTOP=CTWTR commands.

See z/OS MVS System Commands, SA22-7627 for information on using the START and TRACE commands.
Chapter 36. Using the DFSMS subchannel set support

In previous releases, there was a 64K subchannel limit for UCBs in z/OS. Now, using the DFSMS subchannel set support enhancement, parallel access volume (PAV) users can define a second set of subchannels to be used with PAV. PAV users will be able to redefine the aliases into the second subchannel set, thus freeing up those device numbers on the first subchannel set for use as additional base devices. As a result, PAV users will be able to increase the number of available devices. The device numbers can be duplicated in the same channel subsystem by being in both subchannel sets.

Restrictions:
1. This support is limited to devices that support alternate subchannel sets.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defining Devices</td>
<td>• PAV alias devices must be defined to an alternative subchannel set using HCD or HCM. For more information, see <a href="https://www.ibm.com">z/OS HCD User's Guide, SC33-7988</a> or <a href="https://www.ibm.com">z/OS and z/VM HCM User's Guide, SC33-7989</a>.</td>
</tr>
</tbody>
</table>
Chapter 37. Using the REPRO MERGECAT FromKey/ToKey Enhancement

This enhancement enables you to repair damaged catalogs by targeting a range of catalog keys in a REPRO MERGECAT command. This ability is provided through the addition of two new keywords for use with the MERGECAT parameter of the REPRO command. Using the MERGECAT FROMKEY and TOKEY parameters, you can copy unbroken segments of a damaged catalog to a new catalog, defining both a starting point and an ending point in the range to be copied. This makes it easier to maintain catalogs and recover from problems.

The new MERGECAT FROMKEY parameter specifies the key of the first record you want to copy. The MERGECAT TOKEY parameter specifies the key of the last record you want copied. You can specify generic keys (a portion of the key followed by *). If you specify generic keys, copying stops after the last record whose key matches that portion of the key you specified. If you specify a key longer than the one defined for the data set, the data set is not copied. If the specified key is not found, copying ends at the next lower key. You can specify the MERGECAT FROMKEY/TOKEY keywords when issuing REPRO in a TSO command or as part of a JCL statement for a batch job step where PGM=IDCAMS.

Administering

The FROMKEY/TOKEY keywords are used with the REPRO MERGECAT command to repair broken catalogs in IDCAMS data sets.

Implementing REPRO MERGECAT FROMKEY/TOKEY

In order to implement this new function, specify the MERGECAT FROMKEY and TOKEY keywords in the REPRO command.

- For example, in a TSO command:
  REPRO MERGECAT INDATASET(OLDCAT) OUTDATASET(NEWCAT) FROMKEY(A.B.*) TOKEY(M.J.*)

  This command copies all catalog entries beginning with A.B up to and including all catalog entries beginning with M.J on OLDCAT and copies them to NEWCAT.

- Or in a JCL statement:
  
  REPRO MERGECAT -
  INFILE(inddname) -
  OUTFILE(outddname) -
  FROMKEY(A.B.*) -
  TOKEY(E.F.*)

  This command copies all entries in the catalog identified by inddname, beginning with A.B up to and including all entries beginning with E.F. The output catalog is identified by outddname.
Chapter 38. Using the catalog enhancements

The catalog enhancements available with z/OS V1R7 DFSMS provide you with these benefits:

- You can specify the amount of space to be used to allocate a VSAM volume data set (VVDS) when the allocation is done implicitly. The default value for an implicit VVDS allocation is TRK(10,10), which may not be adequate for the size of your volumes and number of data sets you are creating. Now, using the implicit VVDS space quantity enhancement, you can specify the amount of space in tracks that you want to use when a VVDS is defined implicitly.

- When catalogs are created, DFSMS tunes some default parameter values automatically, to improve performance by temporarily adding buffers and strings to the catalog attributes as needed. Otherwise, the default values are focused on allowing access to the catalog, not on catalog performance. The temporarily tuned values are displayed in a message, IEC391I. You can later make the changes permanent, if desired, by using the IDCAMS ALTER command. This auto-tuning function is enabled automatically in V1R7; you can disable it or explicitly enable it by specifying AUTOTUNING on the ENABLE or DISABLE operands of the MODIFY CATALOG command.

**Before you begin:** For information on these topics and how they apply to your system:

- The VSAM volume data set, explicit and implicit definition; see "Defining a VVDS" in [z/OS DFSMS Managing Catalogs](#).
- VSAM catalog parameter values; see: [z/OS DFSMS Managing Catalogs](#).

The following table lists the types of tasks and associated procedures that you must complete in order to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedures you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administering</td>
<td>• Specify the amount of space for an implicit VVDS allocation</td>
</tr>
<tr>
<td></td>
<td>• Disabling or enabling catalog auto-tuning</td>
</tr>
</tbody>
</table>

**Administering**

You can implement the functions associated with the catalog enhancements by completing the following tasks:

- Specify the amount of space for an implicit VVDS allocation
- Use catalog auto-tuning

**Specifying the amount of space for implicit VVDS allocation**

To specify the amount of space for an implicit VVDS allocation, use the new VVDSSPACE keyword on the MODIFY CATALOG command. The format is:

```
VVDSSPACE(primary,secondary)
```

Where primary and secondary are the allocation amounts in tracks to use for an implicitly defined VVDS. The default value is ten tracks for both the primary and secondary values.
To see the current values for primary and secondary VVDS space, issue a MODIFY CATALOG,REPORT command. The results appear in message IEC359I, in a new output line:

```
DEFAULT VVDS SPACE   - (pri,sec) TRKS
```

where:

- **pri** is the primary space allocation unit in tracks for an implicit VVDS definition.
- **sec** is the secondary space allocation unit in tracks for an implicit VVDS definition.

**Related reading:** For more information, see [z/OS DFSMS Managing Catalogs](#).

### Using catalog auto-tuning

The DFSMS catalog auto-tuning enhancement automatically tunes some default parameter values to improve performance, temporarily modifying the number of data and index buffers and VSAM strings for the catalog on the current system. When the tuning occurs, message IEC391I displays the temporarily tuned values in the following format:

```
IEC391I CATALOG catname HAS BEEN AUTOTUNED TO:
IEC391I BUFNI: n1 BUFD: n2 STRNO: s1
```

where:

- **catname** is the name of the catalog whose parameters have been temporarily updated on the current system.
- **n1** is the number of new index buffers assigned to the catalog.
- **n2** is the number of new data buffers assigned to this catalog.
- **s1** is the number of strings assigned to this catalog.

You can later make the updates permanent by using the IDCAMS ALTER command.

This auto-tuning function is enabled automatically. If necessary, you can disable it or subsequently re-enable it by specifying AUTOTUNING on the DISABLE and ENABLE operands of the MODIFY CATALOG command.
Chapter 39. Using VSAM extent constraint removal

In previous releases, there was a 255-extent limit for VSAM data sets and, for striped VSAM data sets, a limit of 255 extents per stripe. Now, using the VSAM extent constraint removal enhancement, these extent limits are removed for SMS-managed volumes, if the extent constraint removal parameter in the data class is set to Y (yes). For non-SMS-managed volumes, the previous extent limits still apply.

A VSAM data set can be expanded to 123 extents per volume. This is unchanged from previous releases.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and Installation</td>
<td>• &quot;Planning to use VSAM extent constraint removal&quot;</td>
</tr>
<tr>
<td></td>
<td>• &quot;Setting up VSAM extent constraint removal&quot; on page 168</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and installation tasks that you would perform to use VSAM extent constraint removal.

Planning to use VSAM extent constraint removal

Before you begin: To understand the concepts of VSAM data sets, read z/OS DFSMS Using Data Sets.

Software requirements: The new function name requires z/OS V1R7 and the supplied PTFs.

Coexistence requirements: The default continues to not allow extents beyond 255. Pre-V1R7 systems will not be able to read or write data sets that have been extended beyond 255 extents. PTFs for pre-V1R7 systems are available to prevent the opening of a data set with greater than 255 extents.

Perform the following steps to plan for using the new function:

1. Decide if you want to allow SMS-managed VSAM data sets to be extendable beyond 255 extents. Installations sharing a V1R7 system with any prior DFSMS releases could have data sets that are not usable if they grow beyond the 255-extent limit.

2. Decide which SMS-managed VSAM data sets you want to extend beyond 255 extents.

Now you are ready to set up VSAM extent constraint removal.
Setting up VSAM extent constraint removal

Perform the following steps to set up the VSAM extent constraint removal:

1. Set the VSAM extent constraint removal flag in the SMS data class to Y (yes) via the ISMF panels. The following ISMF panels are updated to include the new Extent Constraint Removal parameter:
   - Data Class Define/Alter
   - Data Class Display
   - Data Class List
   - Data Class List Print
   - Data Class List Sort
   - Data Class List View

2. Repeat the above step for each data class that you want to remove the VSAM extent constraint.

To verify that the VSAM extent constraint is removed, view the ISMF Data Class Display panel for each data class. Ensure that the extent constraint removal flag is set to Y (yes).
Chapter 40. Using VSAM RLS 64-bit data buffers

VSAM record-level sharing (VSAM RLS) is an extension to VSAM that provides direct record-level sharing of VSAM data sets (as opposed to CI-level sharing) from multiple address spaces across multiple systems. VSAM RLS uses the z/OS coupling facility for cross-system locking, local buffer invalidation, and cross-system data caching. Primary users of VSAM RLS include high-volume applications that access VSAM data sets, such as CICS applications.

In previous releases of z/OS, VSAM RLS was limited to using a 31-bit addressable data space for buffering data sets. Now, with z/OS V1R7, you can optionally specify that VSAM RLS uses 64-bit addressable virtual storage for data buffers. Doing so can help you avoid possible buffer space constraints and potentially improve performance for your high-transaction applications.

If you choose not to exploit this function, VSAM RLS continues to use buffers that reside below the 2-gigabyte bar.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation”</td>
<td>• “Planning to use VSAM RLS 64-bit data buffers”</td>
</tr>
<tr>
<td></td>
<td>• “Setting up VSAM RLS 64-bit data buffers” on page 170</td>
</tr>
<tr>
<td>“Administering” on page 171</td>
<td>• “Monitoring the use of 64-bit data buffers” on page 174</td>
</tr>
<tr>
<td>“Operating” on page 171</td>
<td>• “Displaying settings for 64-bit data buffers” on page 171</td>
</tr>
<tr>
<td></td>
<td>• “Controlling the use of 64-bit data buffers” on page 172</td>
</tr>
<tr>
<td></td>
<td>• “Specifying the use of fixed real storage for 64-bit data buffers” on page 172</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and setup tasks that you would perform to use VSAM RLS 64-bit data buffers.

Planning to use VSAM RLS 64-bit data buffers

Before you begin: To understand the concepts of VSAM RLS, read z/OS DFSMS Using Data Sets.

Software requirements: Using VSAM RLS 64-bit data buffers requires z/OS V1R7 and the supplied PTFs.

Coexistence requirements: By default, VSAM RLS continues to use buffers that reside below the 2-gigabyte bar (31-bit buffers).
If you plan to enable this function in a sysplex containing both z/OS V1R7 and pre-V1R7 systems, note that only the V1R7 systems can use 64-bit data buffers. In pre-V1R7 systems, data buffers continue to reside below the 2-gigabyte bar.

Perform the following steps to plan for using the new function:

1. Determine whether to enable 64-bit data buffers for VSAM RLS processing. IBM recommends that you use this option, especially for applications with a high rate of critical CICS transactions.

2. Determine whether to permanently fix some amount of real storage for VSAM RLS processing, regardless of whether you choose to enable 64-bit data buffers. Your planning should include checking the hit ratio for VSAM RLS buffers, as shown in SMF record Type 42, subtype 19. An excessive number of misses, for example, might indicate the need for more real storage to be pinned for the use of VSAM RLS data buffers.

3. Verify that the SMSVSAM server for your system is at the V1R7 level, and thus eligible to use 64-bit data buffers. Enter the DISPLAY SMS command, as follows:

   Example:
   ```
   DISPLAY SMS,SMSVSAM
   ```

   If the display command output indicates that cache feature code level B is in effect, the SMSVSAM server is at the correct level.

Now you are ready to set up VSAM RLS 64-bit data buffers.

**Setting up VSAM RLS 64-bit data buffers**

**Before you begin:** To understand the setup tasks required for VSAM RLS, read [z/OS DFSMS Using Data Sets](#).

**Related reading:** For information about modifying DFSMS options, see “Initializing SMS through the IGDSMSxx member” in [z/OS DFSMSdfp Storage Administration](#) and the description of member IGDSMSxx in [z/OS MVS Initialization and Tuning Reference](#).

To use 64-bit VSAM RLS data buffers on a system for a data set, both of the following must be true:

- RlsAboveTheBarMaxPoolSize in the parmlib member IGDSMSxx specifies a number between 500MB and 2 terabytes for the system
- The data class for the data set specifies RlsAboveTheBar=Y.

Specifically, perform the following steps to enable 64-bit data buffers for VSAM RLS processing:

1. Use ISMF to modify the appropriate data classes to add the new attribute "RLS Above the 2-GB Bar" and set this attribute to Y (yes).

2. Use ISMF to assign VSAM data sets to data classes for which "RLS Above the 2–GB Bar" is set to Y.
3. Define the keyword RlsAboveTheBarMaxPoolSize in the active IGDSMSxx parmlib member, and assign each V1R7 system in the sysplex to a value between 500 megabytes and 2 terabytes. You can specify a different value for each system, or one value for the entire sysplex.

4. Optionally, define the keyword RlsFixedPoolSize in the active IGDSMSxx parmlib member, and assign each V1R7 system an appropriate amount of real storage to be permanently fixed for RLS data buffers. This value may be adjusted by VSAM RLS internally to not exceed 80% of available real storage.

5. Enter the SET SMS=xx command to activate the changed parameters in IGDSMSxx.

---

**Administering**

For VSAM RLS 64-bit data buffers, these are the administration tasks.

**Monitoring the use of 64-bit data buffers**

To monitor the use of 64-bit data buffers at your installation, do the following:

1. If your installation dedicated some amount of fixed real storage to VSAM RLS processing, you can monitor the effectiveness of the storage usage by checking subtypes of SMF record Type 42, as follows:
   - Subtype 16 (Data Set Summary) indicates whether a data set is enabled for 64-bit data buffers and whether it uses storage above 2-gigabytes.
   - Subtype 18 (CF Cache Partition Usage) provides statistics on the use of coupling facility cache storage classes.
   - Subtype 19 (LRU record) provides statistics for data buffers above and below the 2-gigabytes.

2. To see which data classes are enabled for 64-bit data buffers, view the ISMF Data Class Display panel for each data class.

3. To see which VSAM data sets are enabled for 64-bit data buffers, view the ISMF Data Class List panel for each data class.

---

**Operating**

You can use operator commands to modify the size and usage of 64-bit data buffers, and check the level of the SMSVSAM server.

**Before you begin:** For more information about using system commands to operate DFSMS, see [z/OS MVS System Commands](https://www.ibm.com/support/knowledgecenter/SSEPGG_0.0.0/com.ibm.zos.v1r11.doc/dfsmsgmt/chap004500.htm).

**Displaying settings for 64-bit data buffers**

You can use operator commands to display the current settings for VSAM RLS 64-bit data buffers.

1. Enter the DISPLAY SMS command.

---

**Example:**
DISPLAY SMS, SMSVSAM

Output:
This example displays the current settings for the SMSVSAM address space, including the following:
- Whether 64-bit data buffers are enabled
- Whether some amount of real storage is fixed for data buffers
- Whether the SMSVSAM server is current (cache level B or later).

---

Controlling the use of 64-bit data buffers

You can use operator commands to control the use of VSAM RLS 64-bit data buffers.

1. Enter the SETSMS command.

   Example:
   SETSMS, RLSABOVETHEBARMAXPOOLSIZE=Y
   This example enables the use of 64-bit data buffers.

2. Enter the SETSMS command.

   Example:
   SETSMS, RLSABOVETHEBARMAXPOOLSIZE=N
   This example disables the use of 64-bit data buffers.

---

Specifying the use of fixed real storage for 64-bit data buffers

You can use operator commands to specify the amount of fixed real storage to be used for VSAM RLS 31-bit or 64-bit data buffers.

1. Enter the SETSMS command.

   Example:
   SETSMS, RLSFIXEDPOOLSIZE=50
   This example reserves 50 megabytes of fixed real storage for the data buffers. It is the total of fixed real storage reserved for the 31-bit and 64-bit data buffers on this system.
Chapter 41. Using the SMS volume and ACS allocation test enhancements

SMS Volume enhancements for this release include Volume Status Change and Volume Selection Messages and Traces. There is also the addition of the ACS Allocation Test environment. These are described in the following list:

- In previous releases, if users want to change the status of a volume from NOTCON (not connected) to any other status (like QUIESCE, DISABLE etc.), they must use the ISMF panels to alter the SCDS and follow that with an activation of the SCDS. This enhancement allows users to use the VSMS,VOLUME command to alter the status of the volume from NOTCON to any of the others.

- As in previous releases, SMS issues summarized analysis messages when volume selection fails. These analysis messages provide diagnostic information by summarizing the number of volumes rejected for each failure reason during volume selection. In many cases, however, these analysis messages are not sufficient for the user to determine why allocation has failed. Also in some cases, a user may require analysis messages to determine why a data set was not allocated as expected even though the allocation was successful. When these situations occur, the user often needs to contact SMS support personnel for assistance and is usually advised to collect SMS traces for diagnosis. This process is time consuming not only to the users, but also to SMS support personnel. In this release the following new functions are added for SMS managed data sets to assist the user in performing problem diagnosis on volume selection with minimum assistance from IBM support personnel:
  1. Providing summarized and detailed analysis messages on request
  2. Externalizing DADSM failure reasons and diagnostic codes in summarized analysis messages
  3. Externalizing volume selection analysis data in SMS trace
  4. Adding new trace data for SMS and non-SMS managed VSAM allocations

- ACS Services can be invoked by various callers representing different environments. Frequently ACS Services needs to know who the caller is since their processing may differ for different callers or environments. In this release a new valid value is being provided for the ACS environment variable to enable the installation’s ACS routines to distinguish a new environment, the Allocation Test environment, from the environment that represents the system’s Allocation processing. For various supported environments such as ALLOC, RECALL, RECOVER, CONVERT, RENAME, CTRANS, STORE, CHANGE, RMMPOOL, and RMMVRS, ACS Services checks ACEROENV to determine which ACS read-only variables contain reliable information to be passed on to the Installation’s ACS routines, and will blank out the invalid ones. With this support, the two environments ALLOCTST and ALLOC will be treated similarly by ACS services but the ACS installation exits will be able to differentiate between them.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.
**Planning and installation**

This section describes the planning and installation tasks that you would perform to use Volume Selection Messages and Traces.

**Planning to use volume selection messages and traces**

**Before you begin:** To understand the concepts of Volume Selection Messages and Traces, read [z/OS Planning for Installation](#) and [z/OS DFSMSdfp Storage Administration](#). To understand how to use the VOLSELMGR and related parameters for Volume Selection Messages and Traces, see IGDSMSxx in [z/OS MVS Initialization and Tuning Reference](#).

**Software requirements:** The new function name requires z/OS V1R7 and the supplied PTFs.

**Setting up volume selection messages and traces**

Perform the following steps to set up the Volume Selection Messages and Traces:

Add the VOLSELMGR parameter and appropriate keyword to your IGDSMSxx member in SYS1.PARMLIB, or the SETSMS command.
Chapter 42. Using the Object Access Method Enhancements

New z/OS DFSMS V1R7 Object Access Method (OAM) enhancements provide you with the following benefits:

- **Immediate recall to DB2 DASD**
  In previous releases, OAM allowed for movement of objects within the storage hierarchy, however, this movement did not take place until the OAM Storage Management Component (OSMC) cycle ran for the storage group where the object resides. When objects have been written to removable media, the response time for reads is 30-60 seconds instead of milliseconds. In many cases, when an object has been referenced, the chances that it will be referenced again are very high. These subsequent references can be much faster if the object has already been recalled to a DB2 table residing on DASD. This release supports the immediate recall of objects, currently residing on removable media, to DB2, for a specified number of days. This allows subsequent requests to read the objects to be satisfied from DB2 DASD rather than another read from the tape or optical volume where the object resides. OSMC will restore the object to its original location after the specified number of days have passed.

- **New CLEAROLDLOC keyword**
  When an object is moved from an optical or tape volume to DB2 DASD during an OSMC cycle, the volser and sector location, or blockid, is retained in the object directory. This is because the transition to DB2 is usually temporary and the object will later move back to optical or tape. While an optical or tape volser is associated with an object, that volser cannot be expired, even if the object is currently residing in DB2. V1R7 adds a new CLEAROLDLOC keyword to instruct OAM to clear the original volser and sector location or blockid in the object directory for a given object when that object is moved by OSMC to DB2 DASD. This new keyword will be most useful to installations that do not normally transition objects back to tape or optical volumes once they have moved to DB2 DASD.

- **New Return to MVS Scratch exit routine**
  A new dynamic exit name, CBRUXTVS_EXIT, is invoked to inform the tape management system when a tape volume has been purged from the OAM inventory. This notification exit is patterned after the DFSMSrmm EDGTVEXT exit and the DFSMShsm ARCTVEXT exit.

- **Enhanced MOVEVOL utility**
  The MOVEVOL utility is enhanced to accommodate OAM scratch volumes. This allows MOVEVOL to be used with the DELETE option to remove scratch volumes from OAM’s inventory.

- **New TAPEDISPATCHERDELAY keyword**
  A new TAPEDISPATCHERDELAY keyword is added to the SETOAM statement in the CBROAMxx PARMLIB member to delay processing of certain requests and minimize demounting and remounting.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.
Application Programming

After installing z/OS DFSMS V1R7, you can take advantage of the new functions in the following ways.

Implementing Immediate Recall to DB2 DASD

For each OSREQ RETRIEVE request for an object residing on optical or tape media, OAM determines whether or not a recall to DB2 DASD is required, either explicitly through the RECALL keyword on the OSREQ RETRIEVE, or implicitly through SETOSMC keywords in the CBROAMxx PARMLIB member. If a recall is specified, OAM initiates a recall request to the OSMC component to write a full copy of the object to DB2 DASD at the same time the OSR component of OAM services the object retrieval. Because these OSMC recall tasks can consume considerable resources, you may wish to use the MAXRECALLTASKS keyword on the SETOSMC statement to limit the number of recalls that can run simultaneously.

Note: MAXRECALLTASKS must be set to a non-zero value to activate explicit or implicit recall to DB2 DASD.

Specifying immediate recall explicitly in OSREQ RETRIEVE invocation

Use the new RECALL keyword to recall an object to DB2 DASD, for example:

```
OSREQ RETRIEVE, MF=(E, PARM_LIST),
   RECALL=60,
   BUFLIST=(R10)
```

In this example, RECALL=60 specifies that the objects are to be recalled to DB2 DASD and kept there for 60 days.

Note: Explicit recalls can be affected by the MAXRECALLTASKS and RECALLOFF keywords described in Table 4.

Specifying immediate recall implicitly through SETOSMC values

You can code immediate recall parameters directly in a SETOSMC statement in the CBROAMxx PARMLIB member or update them in a MODIFY OAM command (F OAM command).

<table>
<thead>
<tr>
<th>SETOSMC keyword</th>
<th>F OAM,UPDATE keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXRECALLTASKS(nnn)</td>
<td>MAXRECALL,n.nn</td>
<td>Specifies the maximum number of RECALL tasks that can be run concurrently. Valid values are 0–255. The default is 0. A value of 0 indicates that no RECALL operations are to be run. This applies to both implicit and explicit recalls.</td>
</tr>
</tbody>
</table>
### Table 4. New SETOSMC Keywords and Functions (continued)

<table>
<thead>
<tr>
<th>SETOSMC keyword</th>
<th>F OAM,UPDATE keyword</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECALLOPTICAL((nnn))</td>
<td>RECALLO,(nnn)</td>
<td>Specifies that objects residing on optical devices are recalled to DB2 for (nnn) days when retrieved. Valid values are 0–255. The default is 0. The object’s pending action date in the object directory is set to today’s date + the number of days specified in (nnn). A value of 0 indicates that the RECALL is for the current day only and then the object is restored to its original location. This applies to implicit recalls only. This keyword can be specified at the global level or at the storage group level.</td>
</tr>
<tr>
<td>RECALLTAPE((nnn))</td>
<td>RECALLT,(nnn)</td>
<td>Specifies that objects residing on tape devices are recalled to DB2 for (nnn) days when retrieved. Valid values are 0–255. The default is 0. The object’s pending action date in the object directory is set to today’s date + the number of days specified in (nnn). A value of 0 indicates that the RECALL is for the current day only and then the object is restored to its original location. This applies to implicit recalls only. This keyword can be specified at the global level or at the storage group level.</td>
</tr>
<tr>
<td>RECALLALL((nnn))</td>
<td>RECALLA,(nnn)</td>
<td>Specifies that objects residing on optical or tape devices are recalled to DB2 for (nnn) days when retrieved. Valid values are 0–255. The default is 0. The object’s pending action date in the object directory is set to today’s date + the number of days specified in (nnn). A value of 0 indicates that the RECALL is for the current day only and then the object is restored to its original location. This applies to implicit recalls only. This keyword can be specified at the global level or at the storage group level.</td>
</tr>
<tr>
<td>RECALLNONE</td>
<td>RECALLN</td>
<td>Specifies that objects residing on optical or tape devices are not recalled to DB2 when retrieved. This applies to implicit recalls only. This keyword can be specified at the global level or at the storage group level.</td>
</tr>
<tr>
<td>RECALLOFF((mode))</td>
<td>RECALLF,ON</td>
<td>OFF</td>
</tr>
<tr>
<td></td>
<td></td>
<td>OFF: Explicit and implicit recalls are enabled. This is the default.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ON: Explicit and implicit recalls are disabled.</td>
</tr>
</tbody>
</table>

---

**Implementing the OAM Object Tape Volume Return to MVS Scratch Exit Routine**

The OAM Object Tape Volume Return to MVS Scratch exit routine, CBRUXTVS_EXIT, can be used to notify the installation’s tape management system that all knowledge of a given tape volume has been removed from OAM’s tape volume inventory. This is a notification only exit; OAM does not change its tape volume expiration processing regardless of the return code supplied by the user.
exit. The exit is invoked after OAM issues the CBR2165I message indicating that OAM has removed the tape volume from the OAM inventory and released it for return to the MVS scratch pool.

The exit point uses Dynamic Exit Services, CSVDYNEX, which allows multiple exit routines to be simultaneously associated with a single exit point. No additional steps are required for you to use this dynamic exit with DFSMSrmm. The DFSMSrmm tape exit (EDGTVEXT) is always invoked independently of the dynamic exit routines. If your installation manages tape volumes with any tape management systems other than DFSMSrmm, you can write the exit routine(s), or use the same load module used for the ARCTVEXT exit and add them to this exit using the MVS dynamic exits facility.

**Implementing the MOVEVOL utility enhancement**

OAM provides a Move Volume utility (MOVEVOL) capable of moving objects from a primary, backup, or scratch source volume (a tape volume or one side of an optical disk) to one or more target volumes. The MOVEVOL command also supports RECYCLE and DELETE keywords, which can optionally be used to return the volume to *like new* status, or to remove knowledge of the volume from the OAM inventory. Prior to V1R7, the MOVEVOL command was rejected if issued for an OAM scratch volume.

New to V1R7, the MOVEVOL command can now be issued for an OAM scratch volume. A successful MOVEVOL command issued with the DELETE option will result in the optical or tape volume being purged from OAM's volume inventory. For example,

```
MODIFY OAM,START,MOVEVOL,volser,DELETE
```

**Implementing the new TAPEDISPATCHERDELAY keyword**

The new SETOSMC TAPEDISPATCHERDELAY keyword specifies that OAM wait a specified number of seconds before demounting a tape volume, even if other work is available for this drive. This delay allows time for a new read request to come into OAM that requires the currently mounted tape volume. This delay can greatly reduce the number of mounts and demounts of volumes for certain applications. This keyword provides function similar to the OPTICALDISPATCHERDELAY keyword associated with the SETOPT statement in the CBROAMxx PARMLIB member.

The OAM tape dispatcher will delay processing of a unit of work for a specific period of time only when all of the following conditions are met:

- A nonzero tape dispatcher delay value has been specified with the TAPEDISPATCHERDELAY keyword on the SETOAM statement in the CBROAMxx PARMLIB member.
- A read request for an object on a currently mounted tape volume has just been completed.
- There is no request for the currently mounted tape volume waiting to be processed on the OAM tape dispatcher queue.
- The OAM tape dispatcher has found a request for another tape volume and is about to dispatch this unit of work.

If another read request for the currently mounted tape volume arrives within the delay interval, that unit of work will be dispatched immediately upon arrival. If no read request for the currently mounted volume arrives within the delay interval, another request for a different tape volume is dispatched.
Implementing the new CLEAROLDLOC keyword

The new SETOSMC CLEAROLDLOC keyword specifies whether OAM is to retain the original volume location information when OSMC processing moves objects from optical or tape media to DB2 DASD. Values for mode are:

OPT
   Specifies that previous volser and sector location values be cleared in the object directory when an object is moved from optical media to DB2

TAPE
   Specifies that previous volser and blockid location values be cleared in the object directory when an object is moved from tape media to DB2

BOTH
   Specifies that previous volser and sector or blockid location values be cleared in the object directory when an object is moved from either optical or tape media to DB2

NONE
   This is the default. Specifies that previous volser and sector or blockid location values be left unchanged in the object directory when an object is moved from optical or tape media to DB2
Chapter 43. Using DFSMSrmm enterprise enablement

In the past, you could use the high-level language application programming interface from C/C++ and Java (using the JNI) code running on the same z/OS system as the DFSMSrmm subsystem. A series of calls to the application programming interface were necessary to run the subcommand and to receive the output. Now, using the DFSMSrmm enterprise enablement enhancement, you can use the high-level language application programming interface as a web service. This enables the high-level language application programming interface to be used from any system or platform that can run Java, C++, or any language that supports the web services standards. Now, it is as if the high-level language application programming interface is available as a locally callable program. A single call to the application programming interface to run a subcommand and receive all the output is all that is needed.

In addition, a plug-in adapter created for the SNIA CIM environment supports removable media. This Java class maps DFSMSrmm resources into those defined in the CIM object model. This plug-in adapter uses the CIM provider interface to provide real-time information about storage resources.

The following table lists the types of tasks and associated procedures that you must complete to fully use this enhancement.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Planning and Installation</td>
<td>• “Planning to use DFSMSrmm enterprise enablement”</td>
</tr>
<tr>
<td></td>
<td>• “Setting up DFSMSrmm enterprise enablement” on page 182</td>
</tr>
<tr>
<td>Operating</td>
<td>• “Operating” on page 182</td>
</tr>
<tr>
<td>Diagnosis</td>
<td>• “Diagnosing” on page 182</td>
</tr>
</tbody>
</table>

Planning and Installation

This section describes the planning and installation tasks that you would perform to use DFSMSrmm enterprise enablement.

Planning to use DFSMSrmm enterprise enablement

Before you begin: To understand the concepts of DFSMSrmm enterprise enablement, read z/OS DFSMSrmm Application Programming Interface.

Software requirements: The enterprise enablement feature requires z/OS V1R7 and the supplied PTFs. WebSphere Application Server for z/OS V5.0.2 and later, or an equivalent, is also required. A fully functional web service as well as an xmlCIM compliant product that supports Java, such as the CIMOM from The Open Group, is required to use the DFSMSrmm CIM provider.

Coexistence requirements: This enhancement is an optional, compatible change. There are no coexistence actions required. The DFSMSrmm control data set can be shared with lower-level systems that have all available toleration maintenance installed.
Perform the following steps to plan for using DFSMSrmm enterprise enablement:

1. Decide on the high-level language to use. C/C++ or any other high-level language is required to exploit the DFSMSrmm class library.

2. Determine the XML parser to use in order to process the XML output from the DFSMSrmm application programming interface.

3. Decide whether you will use a local or a more general UDDI registry to publish your DFSMSrmm application programming interface web service, or if you will directly code the web service location into your application rather than use a UDDI.

4. Decide how and when authentication will be done. Authentication is not done by the DFSMSrmm application programming interface web service.

5. Decide on the xmlCIM compliant product to use.

Now you are ready to set up DFSMSrmm enterprise enablement.

**Setting up DFSMSrmm enterprise enablement**

**Before you begin:** The infrastructure to support the use of web services must be implemented and available on both the application system and the target z/OS system running DFSMSrmm.

**Related reading:** For more information about setting up the DFSMSrmm application programming interface via web services, see [z/OS DFSMSrmm Application Programming Interface](#) and [z/OS DFSMSrmm Implementation and Customization Guide](#). For more information about setting up the DFSMSrmm Common Information Model (CIM) provider, see [z/OS DFSMSrmm Implementation and Customization Guide](#).

**Operating**

To run the DFSMSrmm application programming interface via web services, you must be running WebSphere Application Server or an equivalent (see [181](#)). To run the CIM provider, you must be running a CIM server such as the SNIA Java CIMOM (see [181](#)).

**Before you begin:** For more information, see [z/OS DFSMSrmm Implementation and Customization Guide](#).

**Diagnosing**

You can diagnose DFSMSrmm enterprise enablement problems.

**Before you begin:** For more information on diagnosing DFSMSrmm enterprise enablement problems, see [z/OS DFSMSrmm Diagnosis Guide](#).
Chapter 44. Using the DFSMShsm enhancements

The functional enhancements available with z/OS V1R7 DFSMShsm provide you with these benefits:

- **Extended tape table of contents (TTOCs)**
  To make better use of new high capacity tape volumes, DFSMShsm can write more than one million data sets to a migration tape or backup tape. In previous releases, this number was limited to 330,000 data sets.

- **Simplified recycle criteria for connected sets**
  To allow connected sets to be recycled sooner, and free up more volumes to scratch or tape pools, DFSMShsm allows you to specify that the entire connected set’s average percentage of valid data is to be used to determine whether to recycle a connected set. In previous releases, DFSMShsm required the first volume in the connected set to meet the percent valid criterion before determining the connected set’s average percentage of valid data.

- **INCLUDE statement not required for ABARS processing**
  With this change, ABARS processing no longer requires you to specify an INCLUDE statement in the data set selection list; you need only specify allocation information (an ALLOCATE statement) or tape catalog information (an ACCOMPANY statement).

- **LRECL and DS Empty indicator can be queried**
  For easier analysis of migrated data sets, you can determine the logical record length (LRECL) of a data set that has been migrated, by using the IDCAMS DCOLLECT function and specifying the MIGRATEDDATA option. Additionally, you can determine which migrated data sets are empty through the DFSMShsm LIST command. The LRECL and an “empty data set” flag are now collected in the MCD record for the data set in the MCDS. In previous releases, you needed to recall a migrated data set to learn this information.

- **HMIGRATE processing improved for wildcard data set names**
  For easier data set migrations, DFSMShsm bypasses already-migrated data sets when you specify the HMIGRATE command with a wildcard filter (*). In previous releases, HMIGRATE processing resulted in numerous error messages if your datasetname specification included migrated data sets.

- **Fast subsequent migration processing improved**
  To increase the number of migrated data sets eligible for fast subsequent migration, this function has been changed to use new indicators for reconnection eligibility. This more robust indication of reconnection eligibility can increase your use of fast subsequent migration in two ways. First, the new indication allows you to use fast subsequent migration even if you use a product other than DFSMShsm to back up your data sets. Second, it allows DFSMShsm to reconnect data sets originally migrated to ML2 tape without a valid backup copy (such Tape Mount Management data). In previous releases, these data sets were not eligible for reconnection.

The following table lists the types of tasks and associated procedures that you must complete to fully use these enhancements.

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Procedure that you must perform:</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Planning and Installation” on page 184</td>
<td>• “Planning and Installation” on page 184</td>
</tr>
</tbody>
</table>
Planning and Installation

This section describes the planning and installation tasks that you would perform to use the DFSMShsm enhancements.

Planning to use the DFSMShsm enhancements

**Before you begin:** Be familiar with information about the following topics:

- Offline control data set (OCDS); see z/OS DFSMShsm Implementation and Customization Guide
- The Interactive Storage Management Facility (ISMF) interface; see z/OS DFSMS Using the Interactive Storage Management Facility
- Recycle processing for connected sets; see z/OS DFSMShsm Storage Administration
- Aggregate groups; see z/OS DFSMShsm Storage Administration
- HMIGRATE command; see z/OS DFSMShsm Managing Your Own Data
- Fast subsequent migration; see z/OS DFSMShsm Storage Administration

**Software requirements:** DFSMShsm enhancements require z/OS V1R7 and the supplied PTFs.

**Coexistence requirements:** To use the DFSMShsm enhancements, observe the following coexistence requirements:

- For extended TTOCs in a sysplex, a coexistence PTF is required on lower level systems. It is highly recommended, however, that you do not use extended TTOCs in a sysplex until all systems are upgraded to z/OS V1R7 to avoid inhibiting tape operations on the pre-V1R7 systems.
- For LRECL and DS Empty indicators, this information is available to pre-V1R7 systems only through the FIXCDS command.
- For fast subsequent migration improvements, a coexistence PTF is required on lower level systems in the sysplex.
- For large format journal support, a coexistence PTF is required so lower level releases will fail if they attempt to use a large format journal.
- For large format data sets, a coexistence PTF is required so lower level releases of DFSMShsm will fail migration/recall, backup/recover or ABACKUP/ARECOVER of large format data sets.
Administering

You can implement the functions associated with the DFSMShsm enhancements by completing the following tasks:

- Specify the use of extended TTOCs
- Use simplified recycle criteria for connected sets
- Define an aggregate group without specifying a data set to back-up
- Query the LRECL or DS Empty status for migrated data sets
- Migrate multiple data sets with H迁移
- Specify the use of reconnection.

Specifying the use of extended TTOCs

To allow your installation to use extended TTOCs, do the following:

1. Backup your existing OCDS.
2. Shutdown DFSMShsm.
3. Define a new OCDS with a maximum record size of 6144.
4. Use the REPRO command to copy the old OCDS into the newly defined OCDS.
5. Ensure that the DFSMShsm startup procedure OFFCAT DD statement points to the new OCDS name.
6. Mark all backup and migration tapes that are partials as full by issuing DELVOL volser BACKUP|MIGRATION MARKFULL before restarting DFSMShsm.
   If you do not mark all partial tapes full, when DFSMShsm processing selects volumes, these partial tapes will be selected, the message ARC0309I TAPE VOLUME volser REJECTED, TTOC TYPE CONFLICT will be issued, and the tape will be marked full. DFSMShsm will process all partials, and mark them full before selecting a new scratch tape.
7. Restart DFSMShsm.
8. Enter the SETSYS command with the new EXTENDED TTOC(Y) parameter specified.
   
   Example:
   
   SETSYS EXTTC(Y)

Using simplified recycle criteria for connected sets

You can specify that the RECYCLE PERCENTVALID command is not to use the first volume’s percentage of valid data as a criterion for recycling volumes in a connected set. To do so, enter the RECYCLE PERCENTVALID command with the new CHECKFIRST parameter specified as N (no):

   Example:
   
   RECYCLE PERCENTVALID CHECKFIRST(N)

In this example, recycle processing is reset to use the entire connected set’s percentage of valid data to determine its eligibility for reconnection. The
CHECKFIRST(N) specification is in effect until the next use of the RECYCLE PERCENTVALID command. By default, recycle processing continues to check the first volume, as in previous releases.

If you enter the RECYCLE command with the VOLSER parameter, the system ignores the CHECKFIRST parameter.

Be aware that CHECKFIRST(N) causes recycle processing to always determine the average percentage valid for a connected set, which can slow recycle performance if you have use many large connected sets.

**Defining an aggregate group without specifying a data set to back-up**

An aggregate group is a collection of user-defined data sets that are processed together. You define an aggregate group through Interactive Storage Management Facility (ISMF) panels, and perform backups through the ABACKUP command.

You can define an aggregate group without specifying a data set to back-up. If you omit the INCLUDE statement from the selection data set, you must specify allocation information (an ALLOCATE statement) or tape catalog information (an ACCOMPANY statement) for the aggregate group.

**Querying the LRECL or DS Empty status for migrated data sets**

To determine the LRECL of a migrated data set, or whether it is empty, you can use the following DFSMS commands or functions:

- LIST DSNAMES
- FIXCDS
- IDCAMS DCOLLECT

To display only those migrated data sets that were empty at the time of migration, enter the LIST DSNAMES command with the new SELECT(EMPTY) option:

- **Example:**
  
  LIST DSNAMES(datasetname) MCDS SELECT(EMPTY)

To use the FIXCDS command to query the LRECL of a migrated data set, or whether it is empty, enter the command with the DISPLAY, CREATE, VERIFY, or PATCH options.

- **Example:**
  
  FIXCDS D datasetname DISPLAY

**Note:** The LRECL of a migrated data set can be displayed only with the FIXCDS command or the IDCAMS DCOLLECT function with the MIGRATEDDATA option. For a detailed discussion of the DCOLLECT command, refer to z/OS DFSMS Access Method Services for Catalogs.

**Migrating multiple data sets with HMIGRATE**

To migrate more than one data set at a time, use the HMIGRATE command with a wildcard (*) in the datasetname specification.
In the following example, the HMIGRATE command migrates all level 0 data sets with the prefix and qualifier, GRPA.*.OUTLIST, to migration level 1 volumes.

- Example:
  
  HMIGRATE 'GRPA.*.OUTLIST'

  Here, HMIGRATE no longer attempts to migrate already-migrated data sets.

To migrate data sets from both level 0 and ML1 volumes directly to ML2 volumes, specify the ML2 parameter with the HMIGRATE command.

- Example:
  
  HMIGRATE 'GRPA.*.OUTLIST' ML2

  With this change DFSMShsm will skip data sets already migrated to ML2 tape, but will process data sets migrated to ML2 DASD.

### Specifying the use of reconnection

Determine whether to enable the fast subsequent migration function for potentially faster tape migration operations. This function allows unchanged data sets that are recalled from ML2 tape to be reconnected to the ML2 tape from which they were most recently recalled. You can use fast subsequent migration even if you use a product other than DFSMShsm to back up your data sets.

To enable fast subsequent migration, enter the SETSYS command, as follows:

- Example:
  
  SETSYS TAPEMIGRATION(RECONNECT(ALL))
Appendix. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to z/OS TSO/E Primer, z/OS TSO/E User's Guide, and z/OS ISPF User's Guide Vol I for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:
http://www.ibm.com/systems/z/os/zos/bkserv/
Notices

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:
Programming interface information

This publication documents information that is NOT intended to be used as Programming Interfaces of DFSMS.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at [http://www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.
Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual or view the Glossary of Computing Terms located at:

http://www.ibm.com/ibm/terminology/

This glossary includes terms and definitions from:

- The American National Standard Dictionary for Information Systems, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.

- The Information Technology Vocabulary developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.


The following cross-reference is used in this glossary:

See: This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

access method services. A multifunction service program that manages both VSAM and non-VSAM data sets and integrated catalog facility catalogs or VSAM catalogs. It defines data sets and allocates space for VSAM data sets, VSAM catalogues, and ICF catalogs. It converts indexed-sequential (ISAM) 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 and indexes, helps make inaccessible data sets accessible, lists the records of data sets and catalogs, defines and builds alternate indexes, and converts CVOLs and VSAM catalogs to integrated catalog facility catalogs.

access permission. A group of designations that determine who can access a particular AIX or UNIX file and how the user can access the file.

ACS. Automatic class selection.

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, communications data set.

alias. An alternative name for an ICF user catalog, a non-VSAM file, or a member of a partitioned data set (PDS) or PDSE.

automatic class selection (ACS). A mechanism for assigning Storage Management Subsystem classes and storage groups to data sets.

B

base addressing space. On an extended address volume, the cylinders with addresses below 65,536. These cylinder addresses are represented by 16-bit cylinder numbers or by 28-bit cylinder numbers with the high-order 12 bits equal to zero.

basic format. The format of a data set that has a data set name type (DSNTYPE) of BASIC. A basic format data set is a sequential data set that is specified to be neither large format nor extended format. The size of a basic format data set cannot exceed 65,535 tracks on each volume.

basic partition access method (BPAM). An access method that can be applied to create program libraries in direct access storage for convenient storage and retrieval of programs.

BPAM. See Basic partitioned access method.
catalog.  (1) A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors. (2) A data set that contains extensive information required to locate other data sets, to allocate and deallocate storage space, to verify the access authority of a program or operator, and to accumulate data set usage statistics. (A) (ISO) (3) To enter information about a file or a library into a catalog. (A) (ISO) (4) The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set. (5) To include the volume identification of a data set in the catalog. (6) See VSAM master catalog, VSAM user catalog.

CAS. catalog address space.

catalog address space. The area of virtual storage where catalog functions are performed. It contains tables with all user catalog names identified in the master catalog, their aliases, and their associated volume serial numbers. Any changes to the master catalog are automatically reflected in these tables.

coupling facility (CF). The hardware that provides high-speed caching, list processing, and locking functions in a Parallel Sysplex.

cylinder-managed space. The space on the volume that is managed only in multicylinder units. Cylinder-managed space begins at cylinder address 65 520. Each data set occupies an integral number of multicylinder units. Space requests targeted for the cylinder-managed space are rounded up to the size of a multicylinder unit. The cylinder-managed space exists only on extended address volumes.

D

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.

device. This term is used interchangeably with unit. For a disk or tape, a unit on which a volume may be mounted. For example, a tape drive is a device; a tape cartridge is a volume. Device also applies to other types of equipment, such as a card reader or a channel-to-channel (CTC) adapter.

Device Support Facilities (ICKDSF). A program used for initialization of DASD volumes and track recovery.

DFSMS. See Data Facility Storage Management Subsystem.

DFSMSdfp. A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss. A DFSMS functional component or base element of z/OS, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm. A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

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, this function is provided by DFSMS, DFSORT, and RACF. See also system-managed storage.

direct access device space management (DADSM). A collection of subroutines that manages space on disk volumes. The subroutines are: Create, Scratch, Extend, and Partial Release.

dump. A capture of valuable storage information at the time of an error.

E

ECS. Enhanced Catalog Sharing.

extended address volume (EAV). A volume with more than 65 520 cylinders.

extended addressing space (EAS). On an extended address volume, the cylinders with addresses that are equal to or greater than 65 536. These cylinder addresses require more than 16 bits to represent.

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. Data sets in extended format can be striped or compressed. Data in an extended format VSAM KSDS can be compressed. See also striped data set, compressed format.

extent. A file extent is a storage area for records allocated to a file by the server. Extents are not formally architected in DDM.

G

GDG. See generation data group.

GDS. See generation data set.
**generation data group (GDG).** A collection of historically related non-VSAM data sets that are arranged in chronological order; each data set is a generation data set.

**generation data set.** One generation of a generation data group.

**ICKDSF.** See *Device Support Facilities program.*

**initial program load (IPL).** (1) The initialization procedure that causes an operating system to commence operation. (2) The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction. (3) The process of leading system programs and preparing a system to run jobs. (4) Synonymous with system restart, system startup.

**Interactive Storage Management Facility (ISMF).** The interactive interface of DFSMS/MVS that allows users and storage administrators access to the storage management functions.

**Interactive System Productivity Facility (ISPF).** An interactive base for ISMF.

**IPL.** See *initial program load.*

**ISMF.** See *Interactive Storage Management Facility.*

**ISPF.** See *Interactive System Productivity Facility.*

**JCL.** See *Job control language.*

**Job control language (JCL).** A problem-oriented language used to identify the job or describe its requirements to an operating system.

**large format.** The format of a data set that has a data set name type (DSNTYPE) of LARGE. A large format data set has the same characteristics as a sequential (non-extended format) data set, but its size on each volume can exceed 65,535 tracks. There is no minimum size requirement for a large format data set.

**Logical partition (LPAR).** An LPAR uses software and firmware to logically partition the resources on a system. An LPAR consists of processors, memory, and I/O slots available in one processor complex.

**master catalog.** A key-sequenced data set or file with an index containing extensive data set and volume information that VSAM requires to locate data sets or files, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set or file, and to accumulate usage statistics for data sets or files.

**multicylinder unit.** A fixed unit of disk space that is larger than a cylinder. For example, a multicylinder unit might be 21 cylinders; in this case, the number of the first cylinder in each multicylinder unit would be a multiple of 21.

**multilevel security.** A security policy that allows the classification of data and users based on a system of hierarchical security levels (for example: unclassified, secret, top secret) combined with a system of non-hierarchical security categories (for example: Project A, Project B, Project C). In order to access data, a user must have a security level greater than or equal to that of the data, and be authorized to all of the categories assigned to the data.

**name hiding.** Prevents unauthorized users from obtaining names about data sets.

**non-VSAM data set.** A data set allocated and accessed using one of the following methods: BDAM, BPAM, BISAM, BSAM, QSAM, QISAM.

**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.

**PDS.** See *Partitioned data set.*

**PDSE.** See *partitioned data set extended.*

**performance.** (1) A measurement of the amount of work a product can produce with a given amount of resources. (2) In a system-managed storage 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.

**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.*
RACF. See Resource access control facility.

Resource Access Control Facility (RACF). An IBM licensed program that is included in z/OS Security Server and is also available as a separate program for the z/OS and VM environments. RACF provides access control by identifying and verifying the users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

SCDS. See Source control data set.

sequential data set. A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. Contrast with direct data set.

SMS. Storage Management Subsystem.

SMS class. A list of attributes that SMS applies to data sets having similar allocation (data class), performance (storage class), or backup and retention (management class) needs.

SMS-managed data set. A data set that has been assigned a storage class.

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, communications data set.

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 facility. The physical components that comprise a single storage server (DS8000 or DS6000) including the base frame and the optional expansion frames. A storage facility is composed of two processor complexes (servers) and some number of storage devices that are packaged in one or more enclosures with associated power supplies and cooling.

storage facility image. For hosts that use FICON/ESCON I/O commands, a storage facility image contains one or more ESCON or Fibre Channel (FICON) I/O interfaces (ports) that can access one or more control-unit images. Each control-unit image has an associated set of devices. Each device is assigned a unique device address on the control-unit image. Depending upon the model, more than one storage facility image can be configured on a storage facility. (For DS8000, the storage facility can support more than one storage facility image.) A storage facility image might also be referred to as a storage image.

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, dummy storage group.

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 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.

stripe. In DFSMS, the portion of a striped data set, such as an extended format 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.

sysplex. A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

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 system-managed storage environment.

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.
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.

Track address. A 32-bit number that identifies each track within a volume. A track address is in the format hex CCCCcccH, where CCCC is the low-order 16 bits of the cylinder number, ccc is the high-order 12 bits of the cylinder number, and H is the four-bit track number. For compatibility with older programs, the ccc portion is hex 000 for tracks in the base addressing space.

Track-managed space. The space on a volume that is managed in tracks and cylinders. For an extended address volume, track-managed space ends at cylinder address 65 519. Each data set occupies an integral number of tracks.

UNIX. A highly portable operating system originally developed by Bell Laboratories that features multiprogramming in a multi-user environment. UNIX is implemented in the C language. UNIX was originally developed for use on minicomputers but has been adapted on mainframes and microcomputers. It is especially suitable for multiprocessor, graphics, and vector-processing systems.

VLF. Virtual lookaside facility.

Virtual storage access method (VSAM). An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by the relative-record number.

Volume. The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also DASD volume, optical volume, tape volume.

VSAM. See virtual storage access method.
Index

A
ABARS processing enhancement 183
accessibility 189
ACS Allocation Test 173
aggregate groups without backups administration 186

B
basic format data sets creating 156
BSAM page fixed data buffers 83 overview 83

C
CANCEL command using 113
catalog auto-tuning 165 overview 165
catalog enhancements check 11 enabling JOBCAT and STEPCAT DD statements 165 using catalog auto-tuning 12, 166 catalog enhancements for R8 overview 121 catalog enhancements, R11 description 11 catalog enhancements, R7 description 165 catalog healthcheck 11 CIM Provider overview 95 CIM provider enhancements planning tasks 31, 76, 96 COPY DATASET command using 115 copy pool altering 104 COPYSCDS keyword on SETSMS command 129 creating an ACDS directly from a SCDS 129

D
data set migration
HMIGRATE command 186
DEBUG trace event for SMS overview 125
DEFINE DUMPCLASS command using 107
device support address space 159
DFSMs (continued)
OAM enhancements 15, 63, 91, 133, 175 PDSE 64-bit directory buffers 117 PDSE caching statistics 55 REPROMERGECAT FromKey/ToKey 163 tape data set authorization 141 VSAM RLS 64-bit data buffers 169 VSAM RLS secondary lock structures 49 VSAM SMB enhancements 87 DFSMS subchannel set support 161 DFSMSd fp
ACS Allocation Test 173 AMS enhancements 7 auto-tuning of VSAM catalogs 165 catalog healthcheck 11 creating an ACDS directly from a SCDS 129
DEBUG trace event for SMS 125 device support address space 159
DISPLAY SMS,SMSVSAM, DIAG(CONTENTIONS) 125 extended address volumes 3, 35 implicit VVDS space quantity enhancement 165 large format data sets 153 MODIFY CATALOG,VDUMPOFF command 125 MODIFY CATALOG,VDUMPON command 125 SMS enhancements 9, 59 SMS volume selection 131 Volume Selection Messages and Traces 173 Volume Status Change 173 DFSMSdss
dump, new name 72 enhancements 69 DFSMSdss enhancements 69 NEWNAMEUNCONDITIONAL keyword 69 DFSMSshm DFSMSshm fast replication, tape support 103 enhancements 23, 69, 183 fast replication, data set recovery 109 NEWNAME keyword 69 RETAINDAYS keyword 23 usability enhancements 145 DFSMSshm enhancements 23, 69 administration 185 overview 183 planning tasks 184 updates to the FSR 99 DFSMSshm fast replication, data set recovery 109 administration 111 canceling recovery requests 113 copy pool information 114 DFSMSshm fast replication, data set recovery (continued) defining dump classes 107 DFSMSshsm support 109 holding recovery requests 113 limiting recovery requests 114 overview 109 physical data set copy 115 planning tasks 110 releasing recovery requests 113 selecting recovery method 114 setting up tasks 111 DFSMSshm fast replication, tape support 103 administration 106 deleting unneeded dump copies 106 dumping backups to tape 106 overview 103 planning tasks 104 recovering a volume from a dump tape 106 setting up tasks 104 DFSMSshm usability enhancements 145 administration 25, 72, 147 operations 27, 73, 149 overview 23, 69, 145 planning tasks 25, 71, 146 DFSMSrmm CIM Provider 95 ease of use items 30, 76 enterprise enablement 31, 75, 181 enterprise level interface 137 operating 30 optimization and management 75 programming enhancements 30 RAS enhancements 95 report generator 29, 75 tape data set authorization 137 usability items 95, 137 UTC implementation 137 vital record specification policy management simplification 137 DFSMSrmm API enhancements planning tasks 32 DFSMSrmm control data set serialization planning tasks 98 DFSMSrmm ease of use items planning tasks 79 DFSMSrmm enhancements planning tasks 31, 76, 96, 138 DFSMSrmm enterprise enablement enhancements planning tasks 77 DFSMSrmm migration planning tasks 31 DFSMSrmm operational enhancements planning tasks 32 DFSMSrmm task management enhancements planning tasks 97

© Copyright IBM Corp. 2004, 2009
DFSMUsmm usability items
planning tasks 98, 140
DFSMUsmm UTC implementation
planning tasks 139
DFSMUsmm utility and inventory
management enhancements
planning tasks 78, 97
disability 189
DISPLAY SMS,
SMSVSAM,DIAG(CONTENTION) 125
DISPLAY SMS,SMSVSAM,
DIAG(CONTENTION) overview 125
DSNTYPE
values of LARGE, BASIC, EXTREQ,
and EXTPREF 156
dynamic service task count
planning tasks 121
f2 setting up tasks 122

e
enterprise enablement 181
diagnosing problems 182
operating 182
overview 181
planning tasks 181
setting up tasks 182
diagnosing problems 80, 98, 140
overview 137
planning tasks 138
setting up tasks 96, 139
extended address volume
diagnosing problems with 6, 46
extended address volumes 35
application programming 5, 39
overview 35
planning tasks 3, 36
setting up tasks 37
SMF record changes 41
supported data sets 4, 37
extended address volumes enhancement
enhancements 3
extended address volumes enhancement
settings 4
extended TTOCs
administration 185
extent constraint removal
overview 167
EXTPREF
specifying as DSNTYPE 156
EXTRQ
specifying as DSNTYPE 156

f
FACILITY class resource names
defining 111
fast replication
copy pool information 114
data set recovery 112, 113
defining dump classes 107
limiting recovery requests 114
physical data set copy 115
selecting recovery method 114

i
Implicit VVDS space quantity
enhancement 165
overview 165
ISMF
copy pool application 104

k
keyboard 189

l
large format data sets 153
application programming 156
creating 155
diagnosing problems with 158
overview 153
planning tasks 153
setting up tasks 155
SMF record changes 158
supported environments 154
LIST COPYPOOL command
using 114
LRECL or DS Empty status
querying 186

m
mainframe
education x
MODIFY CATALOG,VDUMPPOOL
command 125
overview 125
MODIFY CATALOG,VDUMPPON
command 125
overview 125
MODIFY DEVMAN command 127
overview 127
MVS
MODIFY DEVMAN command 127

n
new function
setting up tasks 174
new function name
application programming 134, 176
NEWNAME keyword 69
NEWNAMEUNCONDITIONAL
keyword 69

o
OAM enhancements 15, 63, 91, 133, 175
overview 15, 63, 91, 133, 175
system programming 16, 64
OAM tape sublevels
system programming 91
OCE enhancements 13
planning tasks 13

p
PDSE 64-bit directory buffers 117
changing the size 119
overview 117
planning tasks 117
PDSE caching
displaying statistics for 56
PDSE caching data in memory after close
changing by command 93, 120
PDSE caching statistics 55
overview 55

q
QUERY command
using 113

r
RACF
protecting commands 111
recycle criteria for connected sets
administration 185
RELEASE FRRECOV command
using 113
REPRO MERGECAT
FromKey/ToKey 163
RETAINDAYS keyword 23

s
SETSMS command
COPYSCDS keyword 129
DEBUG trace option for 125
SETSYS DUPLEX
ERRORALTERNATE parameter 149
SETSYS FASTREPLICATION command
using 114
SETSYS MAXDSRECOVERTASKS(50)
command using 114

z/OS V1R11.0 DFSMS Using the New Functions
200
SETSYS
TAPEMIGRATION(RECONNECT(ALL))
command using shortcut keys 189
SMF records
changes for extended address volumes 41
changes for large format data sets 158
SMS
DEBUG trace event for 125
SMS volume selection overview 131
SMS volume selection enhancements 89
SnapShot 103
sublevels, tape 91

T
tape data set authorization 137, 141
administration 143
overview 137, 141
planning tasks 141
setting up tasks 142
tape sublevels
OAM 91
the DFSMShsm enhancements 183

U
usability items 95, 137
overview 95, 137
UTC implementation 137
overview 137

V
vital record specification policy management simplification 137
overview 137
planning tasks 139
Volume Selection Messages and Traces
overview 173
Volume Status Change 173
overview 173
VSAM
extent constraint removal 167
VSAM extent constraint removal planning tasks 167
setting up tasks 168
VSAM RLS 64-bit data buffers 169
administration 119, 171
overview 169
planning tasks 169
setting up tasks 118, 170
task name 171
VSAM RLS secondary lock structures 49
administration 51
deleting 53
displaying 52
overview 49
planning tasks 50
VSAM secondary lock structures
setting up tasks 50
VSAM SMB enhancements 87
overview 87
VSAM SMBVSP data class keyword planning tasks 87

W
wildcard filter
HMI MIGRATE command 186

Z
z/OS Basic Skills information center x
z/OS V1R10 updates
DFSMShsm
OAM enhancements 63
PDSE enhancements 55
VSAM RLS secondary lock structures 49
DFSMShsm enhancements 69
DFSMShsm enhancements 23
DFSMShsm enhancements 23
ease of use items 76
enterprise enablement 75
optimization and management 75
z/OS V1R11 updates
DFSM
OAM enhancements 15
DFSMShsm
AMS enhancements 7
catalog healthcheck 11
extended address volumes 3
SMS enhancements 9
DFSMShsm enhancements 23
DFSMShsm enhancements 23
ease of use items 30
enterprise enablement 31
operating 30
programming enhancements 30
OCE enhancements 13
z/OS V1R7 updates
DFSM
DFSM subchannel set support 161
OAM enhancements 175
REPRO MEGECAT FromKey/ToKey 163
VSAM RLS 64-bit data buffers 169
DFSMShsm
ACS Allocation Test 173
device support address space 159
implicit VVDS space quantity enhancement 165
large format data sets 153
Volume Selection Messages and Traces 173
Volume Status Change 173
VSAM catalog, auto-tuning 165

z/OS V1R7 updates (continued)
DFSMShsm enhancements 183
DFSMShsm enhancements 23
enterprise enablement 181
VSAM extent constraint removal 167
z/OS V1R8 updates
DFSM
catalog enhancements 121
OAM enhancements 133
PDSE enhancements 117
tape data set authorization 141
DFSMShsm
DFSMShsm fast replication, tape support 103
enhancements 145
fast replication, data set recovery 109
DFSMShsm
enterprise level interface 137
tape data set authorization 137
usability items 137
UTC implementation 137
vital record specification policy management simplification 137
MVS
MODIFY DEVMAN command 127

z/OS V1R9 updates
DFSM
BSAM page fixed data buffers 83
OAM enhancements 91
SMS volume selection enhancements 89
VSAM SMB enhancements 87
DFSMShsm
CIM Provider 95
RAS enhancements 95
usability items 95

z/OS V1R7 updates (continued)
DFSM
OAM enhancements 175
REPRO MEGECAT FromKey/ToKey 163
VSAM RLS 64-bit data buffers 169
DFSMShsm
ACS Allocation Test 173
device support address space 159
implicit VVDS space quantity enhancement 165
large format data sets 153
Volume Selection Messages and Traces 173
Volume Status Change 173
VSAM catalog, auto-tuning 165

z/OS V1R8 updates
DFSM
catalog enhancements 121
OAM enhancements 133
PDSE enhancements 117
tape data set authorization 141
DFSMShsm
DFSMShsm fast replication, tape support 103
enhancements 145
fast replication, data set recovery 109
DFSMShsm
enterprise level interface 137
tape data set authorization 137
usability items 137
UTC implementation 137
vital record specification policy management simplification 137
MVS
MODIFY DEVMAN command 127

z/OS V1R9 updates
DFSM
BSAM page fixed data buffers 83
OAM enhancements 91
SMS volume selection enhancements 89
VSAM SMB enhancements 87
DFSMShsm
CIM Provider 95
RAS enhancements 95
usability items 95
Readers’ Comments — We’d Like to Hear from You

z/OS
DFSMS Using the New Functions
Publication No. SC26-7473-06

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

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. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.
Submit your comments using one of these channels:
• Send your comments to the address on the reverse side of this form.
• Send your comments via e-mail to: mhvrdfs@us.ibm.com

If you would like a response from IBM, please fill in the following information:

Name

Address

Company or Organization

Phone No.

E-mail address