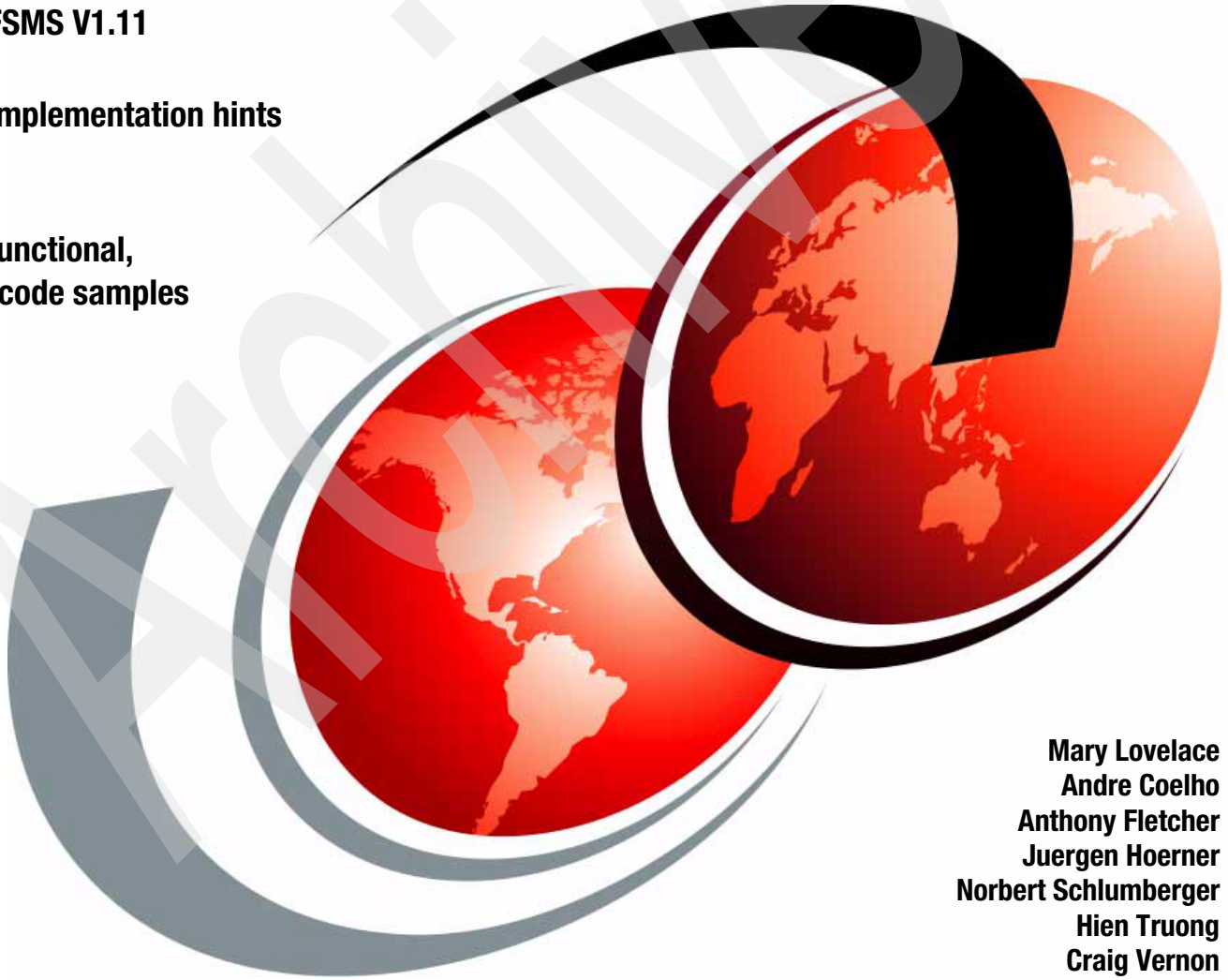
IBM

z/OS V1.11 DFSMS Release Guide

- Understand the features and functions of z/OS DFSMS V1.11
- Contains implementation hints and tips
- Provides functional, hands-on code samples



Mary Lovelace
Andre Coelho
Anthony Fletcher
Juergen Hoerner
Norbert Schlumberger
Hien Truong
Craig Vernon

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/OS V1.11 DFSMS Release Guide

November 2010

SG24-7768-00

Note: Before using this information and the product it supports, read the information in “Notices” on page xi.

First Edition (November 2010)

This edition applies to Version 1, Release 11 of z/OS V1.11 DFMS (product number 5694-A01).

Contents

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.

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.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

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. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US 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

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

AIX 5L™	IBM®	RETAIN®
AIX®	IMS™	RMF™
CICS®	Language Environment®	System Storage®
DB2®	MVS™	System z10®
DS8000®	OS/390®	System z®
Enterprise Storage Server®	POWER5™	Tivoli®
ESCON®	RACF®	VTF®
FICON®	Redbooks®	z/OS®
FlashCopy®	Redbooks (logo) ®	z10™

The following terms are trademarks of other companies:

Snapshot, and the NetApp logo are trademarks or registered trademarks of NetApp, Inc. in the U.S. and other countries.

SAP, and SAP logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries.

ACS, and the Shadowman logo are trademarks or registered trademarks of Red Hat, Inc. in the U.S. and other countries.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries 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.

Preface

Each release of data facility storage management subsystem (DFSMS) builds upon the previous version to provide enhanced storage management, data access, device support, program management, and distributed data access for the z/OS® platform in a system-managed storage environment.

This IBM® Redbooks® publication provides a summary of the functions and enhancements in z/OS V1R11 DFSMS. It provides you with the information that you need to understand and evaluate the content of this DFSMS release, along with practical implementation hints and tips. Also included are enhancements that were made available through an enabling PTF that has been integrated into z/OS DFSMS V1R11.

This book was written for storage professionals and system programmers who have experience with the components of DFSMS. It provides sufficient information so that you can start prioritizing the implementation of new functions and evaluating their applicability in your DFSMS environment.

The team who wrote this book

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Mary Lovelace is a Consulting IT Specialist at the International Technical Support Organization. She has more than 20 years of experience with IBM in large systems, storage, and storage networking product education, system engineering and consultancy, and systems support. She has written many IBM Redbooks publications about IBM Tivoli® Storage Productivity Center, IBM Tivoli Storage Manager, and z/OS storage products.

Andre Coelho is an IT Storage Specialist in Rio de Janeiro, Brazil, but working for IBM West Internal Accounts, Boulder, Colorado. He has 25 years of experience in z/OS and predecessors. His areas of expertise include DFSMSdfp, DFSMShsm, and DFSMSrmm.

Anthony Fletcher is an IT Specialist working with the Global Services Delivery (GSD) z/OS software platform of IBM Global Services OnDemand Infrastructure Services, based in New Zealand, but working for New Zealand and Australia. He has over 35 years of experience in z/OS, OS/390®, and their predecessors and related components, both as a customer of IBM and with IBM Global Services. He is a team leader for the mainframe operations of four diverse clients in the banking, airline, investment, and telecommunications industries. He holds a degree in Electrical Engineering from SALFORD University, Lancashire, UK. His main areas of expertise include DFSMS, DFSMSrmm, and DFSMShsm, and he has a working knowledge of RACF®. He has been involved with writing several IBM Redbooks publications. He also has experience in installing non-IBM products for the GSD platform.

Juergen Hoerner is a is an IT Specialist working for IBM TSS Mainz, Germany. He has 19 years of experience in supporting IBM OS/390 Software. His areas of expertise include z/OS DFP, DFSMS, NFS, UNIX® System Services, and DFS/SMB. In this book, he wrote Chapter 11, “Network File System enhancements” on page 443.

Norbert Schlumberger is an IT Architect with IBM Germany. He has 33 years of experience in storage software and storage management for IBM and customer systems, including 21 years of experience in DFSMSrmm. He also has experience in DFSMShsm and a good knowledge of RACF. Norbert's areas of expertise include performing conversions from vendor tape management products to DFSMSrmm, new DFSMSrmm implementations, and marketing support for DFSMSrmm, including IBM 3494/IBM 3495 ATLs, VTSs, TS7700, and vendor robotics. He wrote a tool called “Tape Copy Tool: Enhancing DFSMSrmm” that is used to copy data sets residing on tape from one media to another media, including the update of the ICF user catalog and DFSMSrmm after all data is successfully copied. He also has experience in installing and using the IBM System Storage® VTF® Mainframe. He has worked at IBM for 36 years.

Hien Troung is a Storage Specialist in Australia. He has 21 years of experience in MVS™, OS/390, and z/OS. He holds a degree in Computer Science from La Trobe University, Melbourne, Australia. His areas of expertise include DFSMS, DFSMShsm, ICF Catalogs, and he has a good knowledge of RACF.

Craig Vernon is a Senior IT Specialist in IBM Australia. He holds a Bachelor of Applied Science in Computing Sciences from the University of Technology, Sydney, and a Bachelor of Laws from Macquarie University, also in Sydney. He has 30 years of experience with z/OS and IBM large systems. He has worked at IBM for over 22 years and his areas of expertise include z/OS software problem diagnosis, z/OS installation, maintenance and customization, Linux® on z/Series installation and problem diagnosis, and RACF and DFSMSrmm conversions. He has written about FICON® implementation, Linux on z/Series hardware, and software copyright issues. Craig has also written several z/OS DFSMS courses.

Thanks to the following people for their contributions to this project:

Rich Conway and Bob Haimowitz
International Technical Support Organization

Elcana Cohen and Doron Tal
IBM Israel

Mike Wood
IBM UK

Guilhermo Gil Carral and Luiz Carlos Bastos de Amorim
IBM Brazil

Mike Bull and Mike Madrid
IBM San Jose, California

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author - all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

New and enhanced functions in z/OS V1.11 DFSMS

Each release of DFSMS builds upon the previous version to provide enhanced storage management, data access, device support, program management, and distributed data access for the z/OS platform in a system-managed storage environment.

In this chapter, we review the enhancements in z/OS V1.9 DFSMS and z/OS V1.10 DFSMS. We then provide an overview of the function and enhancements in z/OS V1.11 DFSMS. This chapter provides an introduction to the DFSMSV1.11 release content and is a summary of the content in this book.

1.1 DFSMS enhancement summaries

This chapter provides the following summaries:

- Summary of the enhancements in DFSMS V1.9 in “DFSMS V1R9 summary” on page 2
- Summary of the enhancements in DFSMS DFSMS V1.10 in “DFSMS V1R10 enhancement summary” on page 4
- Summary of the enhancements in DFSMS V1R11 that are presented in this book in “DFSMS V1R10 and EAV summary” on page 7.

1.2 DFSMS V1R9 summary

A description of the significant DFSMS V1R9 enhancements are provided in chapters 2 through 8 of *DFSMS V1R10 and EAV Technical Guide*, SG24-7617. In this section, we provide a summary of the enhancements.

1.2.1 DFSMSdfp enhancement summary

The following DFSMSdfp enhancements are delivered in z/OS V1.9 DFSMS:

- Basic Access Methods (BAM) performance enhancements
- VSAM System-Managed Buffering (SMB) enhancements
- Pre-DFSMS V1.9 changes to page data sets
- IDCAMS LISTCAT
- TSO TRANSMIT/RECEIVE - RECEIVE error (SMS related)
- Large format data set support for TSO commands and facilities
- Large format data set support for z/OS FTP

1.2.2 DFSMSdss enhancement summary

The following DFSMSdss enhancements are delivered in z/OS V1.9 DFSMS:

- Fast Replication use of new Storage Facility Image (SFI) support

1.2.3 DFSMShsm enhancement summary

The following DFSMShsm enhancements are delivered in z/OS V1.9 DFSMS:

- New cross-memory service invoking DSS
- Functional statistics record (FSR) improvements
- Return priority exit ARCRPEXT changes

1.2.4 ISPF enhancements supporting DFSMS

The following ISPF enhancements are delivered in z/OS V1.9 DFSMS:

- Instead of having LPAR-specific profile data sets, ISPF V1.9 now enables support for concurrent sharing of ISPF profile data sets across a sysplex.
- ISPF V1.9 enables editing, browsing, and viewing of z/OS UNIX files.

- System symbols can be specified within data set names and volumes entered on ISPF panels.
- The ISPF Data Set List Utility now totals the number of tracks and selected data sets at top of the list.

1.2.5 DFSMS PDS and PDSE enhancements

The following PDS and PDSE enhancements are delivered in z/OS V1.9 DFSMS:

- A binder change to prevent the usage of PDS that is not specified as RECFM=U.
- The volcount restriction introduced in DFSMS V1 R8 has been removed by a DFSMS V1R9 PTF.

1.2.6 DFSMSdfp V1R9 OAM enhancements

The Object Access Method (OAM) storage hierarchy consists of three levels (disk, optical, and tape). To offer further granularity at the tape level, two new sublevels within the tape level of the OAM storage hierarchy are introduced with DFSMS V1R9. This effectively expands OAM's storage hierarchy to four levels:

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

1.2.7 RLS dumpspace enhancements

Before z/OS V1R9, when a VSAM RLS abend occurred, an abend dump was taken only on the originating system where the abend occurred. In z/OS V1R9, new sysplex-wide dumping support causes systems running VSAM RLS in the sysplex to take dumps (the originating system takes a dump as well).

1.2.8 DFSMSrmm V1R9 enhancements

The DFSMSrmm enhancements in DFSMS V1.9 provide improvements in the areas of new hardware support, checking of long running requests, improved interaction with system managed libraries, unqualified data set name support, and subcommand enhancements.

The following topics are covered:

- Task management support
- Multitasking of utilities
- Control data set (CDS) serialization
- JCL data set names
- Data set names in RMM subcommands
- Shared parmlib support
- TSO subcommands
- 3592 Model E05 software support
- REPORT17 of EDGRRPTE REXX exec
- Migration and coexistence considerations
- Common Information Model (CIM) provider

1.2.9 DFSMS V1R9 Network File Systems enhancements

Network File System (NFS) is a base element of z/OS that allows remote access to z/OS host processor data from workstations, personal computers, or any other system on a TCP/IP network that is using client software for the Network File System protocol.

The following enhancements are delivered in z/OS V1.9:

- 24-bit addressing relief
- Multiple TCP/IP stack support
- AddDS Operator Command
- RACF Data Labeling
- NFS V4 client support
- Client Attribute syntax
- Server CTrace upgrade
- Terminal ID based restricted MVSLOGIN

1.3 DFSMS V1R10 enhancement summary

The DFSMS V1R10 enhancements in general are described in chapters 10 through 17 of *DFSMS V1R10 and EAV Technical Guide*, SG24-7617. A summary of the enhancements is provided here.

1.3.1 DFSMSdfp V1R10 enhancements

The following DFSMSdfp enhancements are delivered in z/OS V1.10 DFSMS:

- DFSMS data class space override
- System determined block size set by DATACLASS
- New read only variables in DATACLASS
- Message enhancements

1.3.2 DFSMS V1R10 PDSE

The following PDSE enhancements are delivered in z/OS V1.10 DFSMS:

- BMF DISPLAY HSPSTATS command

  There are several options of the DISPLAY SMS,PDSE,HSPSTATS or DISPLAY SMS,PDSE1,HSPSTATS command that can be selected depending on the granularity required in the output.

- BMF DISPLAY VSTOR command

  The DISPLAY SMS,PDSE,VSTOR command output shows the current PDSE 64-bit directory buffer virtual storage utilization.

- Significance of the HSPSTATS

  The performance of access to PDSE data sets depends on the cache tuning, which depends on the types of PDSE data sets that are being used. The characteristics of DATA members and MODULE members differ. The DISPLAY HSPSTATS commands provide information about the LRUtime, LRUcycles, and Hiper Space size that work together. Managing the values for these values is a balance between performance and CPU consumption.

1.3.3 DFSMShsm V1R10

The following enhancements are delivered in z/OS V1.10 DFSMShsm:

- NEWNAME parameter for data set backup

  In z/OS V1.10, DFSMShsm has new keywords added to the BACKDS command to provide more flexibility when backing up individual data sets.

- CDS Backup Improvements

  In z/OS V1.10, DFSMShsm starts using XCF to request exclusive enqueue on the ARCGPA/ARCCAT resource needed for backing up the control data sets. Other (long running) DFSMShsm functions in the HSMplex might have enqueued this resource, even if they release their enqueues on certain intervals. Refer to *DFSMShsm Implementation and Customization Guide*, SC35-0418 for more details about these intervals.

- Concurrent Copy enhancements

  As of DFSMShsm V1.10, support for the different concurrent copy techniques has been added. The options are made available through a management class setting (Backup Copy Technique). This setting is used when invoking DFSMSdss during automatic backup, volume backup, or data set command backup.

- RAS: Tape copy enhancement

  The unit name in Tape Copy Needed (TCN) records for internally generated tape copies is the generic unit name generated for input use instead of the original unit name specified for output. This can cause problems in a mixed tape technology environment because esoteric unit names can be used to control allocations to the different technology tape drives. The 'SETSYS unit for TCN unit name' RAS item satisfies the requirement to use the specified SETSYS unit name in the tape copy MWE built from the TCN record.

1.3.4 DFSMSdss V1R10

The following topics are covered:

- DUMP using NEWNAMEUNCONDITIONAL

  Currently, a user of DFSMSdss can only rename a data set during COPY DATASET or RESTORE DATASET processing, because the new name of the data set applies to a data set that is allocated or overwritten by DFSMSdss. This behavior and design of the RENAME or RENAMEUNCONDITIONAL keywords do not satisfy user requirements. Therefore, DFSMSdss adds the NEWNAMEUNCONDITIONAL keyword, which can be specified during logical data set DUMP processing.

  For this support, DFSMSdss chooses a data set name to be associated with the dumped data from a source data set based on rename filtering criteria. Using the NEWNAMEUNCONDITIONAL keyword in dump processing, a user can specify source data sets along with corresponding new names to control the assigning of a new name.

- DEFRAG performance enhancement and CONSOLIDATE command

  As volumes have increased in size, the time that it takes to process them with DEFRAG has also increased. The larger volume size inherently requires more time to process the increased number and size of data sets. This increased processing time affects most customers because they have limited time in which to run maintenance tasks on their systems.

  CONSOLIDATE, being a function that is extracted from DEFRAG, shares the same performance problems. The changes introduced in DFSMSdss V1R10 begin to address this issue.

1.3.5 DFSMS V1R10 Virtual Concurrent Copy

The following enhancements to Virtual Concurrent Copy (VCC) are provided in DFSMS V1.10:

- Comparison of VCC implementation and Snapshot™
- Use of VCC in an EAV environment

  In an environment where Extended Address Volumes are present, the SDM VCC enhancement also allows the use of VSAM linear data sets (LDS) for Working Space Data Sets (WSDSs), so that the WSDSs can be allocated in the cylinder-managed space on an EAV.

1.3.6 DFSMS RLS Secondary Lock Structure support

In DFSMS V1.10, enhancements to RLS Secondary Lock Structure are provided.

In z/OS V1.10, you can define multiple, Secondary Lock Structures for VSAM RLS workloads, to reduce locking constraints. A new SMS Storage Class attribute called the *lock set* lets you specify a DFSMS lock structure to be used for VSAM records. An installation can define up to 256 lock sets per sysplex.

1.3.7 DFSMSdfp V1R10 OAM

The following enhancements to OAM are delivered in z/OS V1.10 DFSMS:

- An increased maximum size for objects that OAM can accept and manage.
- The OSREQ API is extended to include new store-related functions.
- You can permanently mark volumes FULL with the F OAM,UPDATE,VOLUME command.
- There is a new FORCE keyword to stop OSMC faster.
- There is an IEFSSN*xx* PARMLIB member keyword for changed OSREQ QUERY processing.
- A new value named LOST has been added for automatic backup copy retrieval of an object.
- Automatic access to backup can be configured in CBROAMxx PARMLIB.
- A new ONLYIF keyword in the CBROAMxx PARMLIB member allows statements in the PARMLIB member to be associated to a specific system.

1.3.8 DFSMSrmm V1R10

The DFSMSrmm enhancements in DFSMS V1R10 provide improvements in the areas of partitioning tape volumes and system managed tape libraries, EDGRMMnn parmlib commands and options, fast replication to copy the DFSMSrmm CDS, and enterprise system support. The enhancements include the following items:

- Enables use of DFSMSdss COPY Services
- SMF record changes
- Audit controls for release processing
- XREPTEXT tailoring via SYSIN
- Volume replacement policies
- Report generator enhancements

- System Managed Library and tape volume partitioning
- Common Information Model (CIM) Provider

1.3.9 DFSMS V1R10 and EAV summary

A significant part of DFSMS V1R10 provides enhancements to support Extended Addressing Volumes (EAV). The EAV enhancements are described in Part 3 of *DFSMS V1R10 and EAV Technical Guide*, SG24-7617. Here we provide a summary of the EAV content in that book.

Extended Address Volume introduction

This chapter provides information about the concepts and architectural overview of Extended Address Volume (EAV) architecture. The concept of Dynamic Volume Expansion introduced in the IBM System Storage DS8000® and how it can be used to migrate to EAV is also covered in this chapter.

Planning to use EAV

This chapter describes the planning that should be done in relation to software, hardware, and storage management disciplines before using EAV.

DS8000 EAV implementation

This chapter covers the necessary actions to actually implement EAV on a DS8000 from the hardware perspective.

DFSMShsm and EAV Volumes

This chapter covers various considerations related to the use of DFSMShsm when there are EAV volumes in the installation.

In particular, EAV volumes should not be used for DFSMShsm owned volumes (ML1, ML2, and Backup) because DFSMShsm cannot use the cylinder-managed space. The one exception is small data set packing (SDSP) data sets.

Migrated and backed up data sets cannot reside in cylinder-managed space because they are stored as non-VSAM data sets. The one exception is SDSP data sets on ML1, because they are VSAM data sets.

1.4 DFSMS V1.11 summary

The following areas of DFSMS have been enhanced in z/OS V1.11.

1.4.1 DFSMSdfp enhancements

Chapter 2, “DFSMSdfp enhancements in V1R11” on page 15 discusses new and changed functions in DFSMSdfp.

Enhancements to System z10 architecture and the FICON protocol

These enhancements optimize I/O for online transaction processing workloads. When used by the FICON channel, z/OS, and the IBM System Storage DS8000 control unit, High Performance FICON for System z® (zHPF) helps reduce processing impact and improves performance.

O/C/EOV: Performance and RAS improvement

A number of RAS improvements have been made in DFP along with two performance improvements.

Concatenation tape positioning

For IBM Tape Library 3590 and later, technology support was added to use fast positioning of SL and NL cartridges via Position Relative commands to the drive during EOV concatenation processing. EOV concatenation tape positioning involves moving the tape from tape mark to tape mark. This starts and stops tape motion at each tape mark. The measure of this start/stop time is referred to as *backhitch*. Backhitch time is longer for later technology tape drives due in part to the faster tape transport speed across the heads that the drive requires to process the tape. It takes more time to get to processing speed.

IFGSMS14/15 mapping macro DSECTs

Mapping macros for SMF types 14 and 15 have been enhanced to include DSECTs for sections and segments (where beneficial). Prior to z/OS V1.11, the mapping macro contains no DSECTs, making most of the provided mapping of the sections and segments difficult to use.

Automatic recovery for IEC708I

z/OS V1.11 provides automatic recovery for IEC708I to prevent a loop from reading the same volume or set of volumes when duplicate volume serial numbers are detected in the JFCB volume list.

Module updates to use branch relative addressing

Selected OCE CSETS have been converted to use Branch Relative addressing. This reduces the complexity in resolving CSECT addressability difficulties that were brought about by adding new functions.

Additional module entries for OCE component trace

The OCE component trace has been enhanced to include trace entries for IFG019RM, IFG019RN, IFG019RO, IFG019RF, IFG055ZZ, and IFG019RQ. In addition, the common UCB segment is included in all trace entries.

DASD data set expiration NEVEREXPIRE

The DASD data set output expiration date processing has been enhanced to externalize the expiration date in message IEC507D, where the expiration date never expires.

DADSM: Saving the job, step, and time at data set creation

With z/OS V1R11 DFSMS, DADSM records the job name, step name, and time of creation of a data set when a data set is created. This data is saved in Format 9 DSCB, which was introduced in z/OS V1R10 to provide EAV support.

1.4.2 Object Access Method

The enhancements to Object Access Method (OAM) in z/OS V1.11 are summarized here.

OAM 2 GB Object

Prior to z/OS V1.11, the maximum object size of an object on tape or optical media was 256 MB. In z/OS V1.10, OAM implemented 2 GB Object Support, which enabled applications to store objects up to 2 GB in size in DB2® DASD.

In z/OS V1.11, this enhancement extends this support to the tape tier of the OAM storage hierarchy (Stage 2). This support provides full functionality for objects up to 2 GB in size on DASD and tape. However, the maximum object size for objects residing on optical media still remains at 256 MB.

OAM archive retention enhancements

The OAM archive retention enhancements add the following four retention related enhancements to OAM:

- Deletion-hold: Prevents object deletion while an object is in deletion-hold mode.
- Retention-protection: Prevents object deletion prior to an object's expiration date. It does not allow an expiration date to be changed to an earlier date.
- Deletion-protection: Prevents object deletion prior to an object's expiration date.
- Event-based-retention: Sets the object expiration date so that it is dependent on external event notification.

Wildcard for OSREQ QUERY

Prior to V1.11, a generic search could be requested by putting a single asterisk (*) in the right most qualifier of the name. Now a percent sign (%) and underscore (_) wildcard are implemented. One or more percent signs (%) or underscores (_) can be used anywhere in the object name. A mix of asterisk and the new wildcard characters is not supported.

Utility used to modify collection defaults

Prior to V1.11, installations were required to execute a five step manual process, as documented in the OAM PISA, to modify existing Storage Class and Management Class defaults associated with a given OAM collection.

This enhancement provides a utility to modify the Storage Class and Management Class defaults associated with an OAM collection.

CBRUXSAE enhancement

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.

1.4.3 Extended Address Volume enhancements

Chapter 4, "Extended Address Volume enhancements" on page 83 describes the second wave of Extended Address Volume (EAV) enhancements delivered in z/OS V1R11. A summary of the first wave delivered in z/OS V1R10 is also given to provide a comprehensive understanding of EAV as it stands now.

EAV maximum size

The maximum EAV size remains at 262,668 cylinders or 223 GB, as delivered in EAV R1 in z/OS V1.10. The maximum size of an EAV might be expanded in subsequent releases.

Support for non-VSAM data set types

Support was added so that non-VSAM data sets are eligible to reside in Extended Addressing Space (EAS) (beyond cylinder 65519) on an EAV. However, this support has not been enabled, so these data sets cannot reside in the EAS except for extended-format sequential data sets (which by definition are SMS managed).

Support for non-VSAM data set types has only been enabled for extended-format sequential data sets.

EATTR data set level attribute

In z/OS V1.11, a new EATTR data set attribute for controlling the allocation of VSAM and non-VSAM data sets in regard to when extended attribute DSCBs and EAS can be used.

Dynamic Volume Expansion request interface

Dynamic Volume Expansion (DVE) is performed by the IBM System Storage DS8000 Storage Manager and can be requested through its web GUI. 3390 volumes may be increased in size, for example, from a 3390 model 3 to a model 9 or a model 9 to a model A (EAV).

z/OS V1.11 introduces an interface that can be used to make requests for dynamic volume expansion of a 3390 volume on a IBM System Storage DS8000 subsystem from that subsystem.

Automatic VTOC and index reformat after DVE

When a volume is dynamically expanded, the VTOC and VTOC index has to be reformatted to map the additional space. At z/OS V1.10 and earlier, this has to be done manually by the system programmer or storage administrator by submitting an ICKDSF REFVTOC job. With z/OS V1.11, when a volume is increased in size, this is detected by the system, which does an automatic VTOC and index rebuild. The system is informed by state change interrupts (SCIs), which is controlled with new DEVSUPxx parmlib member options.

Straddled extents in V1.11

In z/OS V1.10, there is a restriction that a given extent cannot start in track-managed space and end in cylinder-managed space. In z/OS V1.11, a given extent is now allowed to start in track-managed space and end in cylinder-managed space. You can only have one straddled extent per EAV. Essentially, the space is still contiguous, but starts at a low CCCCHHHH address, using a 16-bit cylinder number, and ends with a high CCCCcccH address, using a 28-bit cylinder number.

EAV migration issues

A number of migration issues are discussed:

- VSAM CA size compatibility.
- SMF records changes.
- VVDS access.
- What programs require migration actions.
- How to find affected programs.
- Programs that access VTOCs.
- Updates to LSPACE macro usage.
- Updates to DEVTYPE macro usage.

1.4.4 DFSMShsm enhancements

The following DFSMShsm enhancements are delivered in z/OS V1.11:

- DFSMShsm data set backup retention period

  In z/OS V1.11, a new keyword, RETAINDAYS, is introduced for the DFSMShsm (H)BACKDS command. It allows users to specify the number of days a backup version is retained. This RETAINDAYS value is applied when a backup copy is rolled off and during EXPIREBV processing.

- DFSMShsm ML1 enhancements

  In DFSMS in z/OS V1.11, migrate and backup data sets larger than 64 K tracks in size may reside on Migration Level 1 (ML1) DASD as large format sequential (LFS) data sets. OVERFLOW volumes can now be used for migration and backup copies for data sets of a installation specified size.

- DFSMShsm Fast Replication enhancements

  The following enhancements have been made in z/OS V1.11:

  – Allows FRRECOV DSNAME to delete or move data sets.

    Prior to V1.11, FRRECOV DSNAME relies upon the current catalog information for a data set to locate the backup copy within the volume copies of a copy pool backup version. Data sets that have moved or been deleted do have reliable catalog information to be able to locate the backup copy.

    If catalog information is captured at the time of an FRBACKUP request, that information can be used to locate the backup copy of the data set within a copy pool backup version and to recatalog the data set properly after recovery. The current catalog information for the data set is not required to be consistent with the information at the time of the FRBACKUP, which allows users to use FRRECOV for deleted or moved data sets.

  – Automates Catalog unallocation during volume FRRECOV.

    When a catalog is allocated by the catalog address space (CAS), the volume on which the catalog resides is also allocated. FRRECOV requires volumes that are to be restored to be unallocated. In this release, DFSMShsm automatically unallocates the catalogs.

  – ALLOWPPRCP settings can be used in the SMS copy pool definition.

    In pre-z/OS V1.11 DFSMS, you can select whether to allow Fast Replication backup and recover copy pool volumes to put PPRC primary volumes in a duplex pending state via the FRBACKUP and FRRECOV commands. The ALLOWPPRCP(YES|NO) keyword remains and is expanded to the ALLOWPPRCP(NO|PMNO|PMPREF|PMREQ) keyword so that the SMS copy pool PPRCP settings may be overridden.

  – Initializes volumes that have been withdrawn from a FlashCopy® relationship.

    DFSMShsm withdraws volume FlashCopy relationships automatically in a NOCOPY environment and also allows users to withdraw relationships for a copy pool version via the FRBACKUP command. DFSMShsm invokes SDM to complete this task. When a withdraw is invoked in this manner, the track by track copy is stopped. It is possible that the pointer to the VTOC starting location and the physical location of the VTOC itself might become out of sync. In this situation, the VTOC becomes inaccessible, resulting in an error when the VTOC is subsequently accessed. The volume must be re-initialized to create a new VTOC.

DFSMShsm and Extended Address Volume

With z/OS V1.10 and higher releases, z/OS has added support for DASD volumes with more than 65,520 cylinders.

The capacity of storage volumes has increased steadily over the years within the constraints of the 3390 storage device architecture. To expand the capacity of DASD storage volumes beyond 65520 cylinders, z/OS had to extend the track address format, which is why the term Extended Address Volume (EAV) denotes a volume of more than 65,520 cylinders.

The new DASD volume maximum for V1.10 and V1.11 is 262,668 cylinders. In z/OS 1.11, the EAV support in DFSMShsm is enhanced to include sequential extended format data sets.

1.4.5 VSAM and Record Level Sharing

Chapter 6, “VSAM and Record Level Sharing enhancements” on page 209 contains information about the enhancements to VSAM and Record Level Sharing.

These enhancements include:

- A new VSAM data trap that provides a RAS enhancement for VSAM record management.
- VSAM trace improvements.
- VSAM SMB performance improvement.
- A catalog health check feature to detect IMBED and REPLICATE.

1.4.6 SMS enhancements

The following SMS enhancements are delivered in z/OS V1.11:

DFSMS IDCAMS delete masking

The IDCAMS DELETE command can be used to delete multiple entries by using a wildcard character as part of the entry name. In z/OS V1R11, IDCAMS now provides more selective criteria on the DELETE command: A new MASK keyword now allows you to specify data set name selection criteria using a mask entry-name, or key filter, with the new keyword.

Data set separation by volume

In z/OS V1R11, SMS supports the allocation of critical data sets (such as DB2 partitions) on separate volumes to help reduce I/O contention. This new function is designed to expand the existing data set separation function, which allows you to specify that critical SMS-managed data sets be separated across extent pools and volumes that are not used by other data sets specified in the separation group.

Demand tape library allocation

In z/OS V1R11, there is a new JCL keyword on the DD statement, SMSHONOR. It is intended to be used to specify that allocation honor the device number or esoteric device type specified by UNIT. This change is intended to help improve serviceability by allowing you to specify a specific device within a tape library when gathering diagnostic information, such as when using GTF tracing.

Dynamic trace table

The new MODIFY DFS command is designed to allow you to format the DFS trace table, restart the trace, display the size and other information about the trace table, and change its size dynamically while the SMB server runs. These changes are intended to help improve serviceability for the SMB server.

1.4.7 Utilities

There are changes to the IDCAMS utility in DFSMS V1.11. IDCAMS, a utility for managing data sets in the system catalogs, has been enhanced to allow more complicated data set name specifications. Here we list the enhancements in z/OS V1.11:

- IDCAMS DELETE in NO MASK mode
- IDCAMS DELETE in MASK mode

1.4.8 DFSMSrmm

The DFSMSrmm enhancements in z/OS V1.11 DFSMS provide improvements in the areas of partitioning your tape volumes and system managed tape libraries, EDGRMMnn parmlib commands and options, fast replication to copy the DFSMSrmm CDS, and enterprise system support.

The following DFSMSrmm enhancements are provided in z/OS V1.11.

- EDGINERS Tape Label Scan

  The DFSMSrmm tape utility EDGINERS performs initialization and erasure of tapes. For tape label read and display, another utility, such as DITTO / File Manager, must be used.

  EDGINERS has been updated to support the reading and cross-verification of tape label information with the records defined in the DFSMSrmm control data set. The new SCAN function helps identify and manage tapes from other systems or that are in a problem state. The EDGINERS SCAN function reads the VOL1 and header labels for the first file on the specify volume.

- Dynamic Installation exits

  DFSMSrmm now uses the Dynamic Exit Services system to manage calls to the installation exits, determine if exit modules exist, and to provide error handling and recovery.

- Returning Volumes to the System-Managed Library

  The sample volume (CBRUXVNL) that is not in the library installation exit is enhanced so that fewer installations require customization. The idea is that volumes are entered into the tape library whenever possible.

- VRSEL GDG option

  There are new DFSMSrmm parmlib options that provide flexibility for how tape generation data sets are managed for cyclic retention.

- Report generator

  Extensive changes have been made to the DFSMSrmm report generator in z/OS V1.11, and these changes have improved usability, enabled more customization of reports, and simplified the way that selection information can be specified. The changes have been made by exploiting recent changes to DFSORT and ICETOOL, data typing, and report type inheritance, The changes further improve the reporting available for DFSMSrmm, DFSMShsm, and other DFSMS components.

- DFSMSrmm usability items

  The following DFSMSrmm usability items are provided in z/OS V1.11:

  - Use of MATCHVRS in the ISPF data set dialog
  - ADDVOLUME subcommand fields
  - Extended SEARCHVOLUME Subcommand

- VRSEL(OLD) Parmlib Option

  The VRSEL(OLD) option is no longer supported. All information related to the use of VRSEL(OLD) has been removed. If you have not yet migrated to the VRSEL(NEW) option, refer to the information in this book.

- Migration considerations
  - Report Generator
  - VRSEL GDG options
  - CBRUXVNL user exit
  - Rexx variables
  - Dynamic Installation Exits
  - VRSEL(OLD) Parmlib option
- DFSMSrmm VTFM setup

  The VTFM setup is not unique to DFSMSrmm or z/OS V1.11. It is included for completeness.

1.4.9 Network File System

In z/OS V1.11, there are enhancements to the z/OS NFS server and z/OS NFS client.

These V1.11 enhancements focus on delivering several line items for adding remaining NFSv4 functionality, performance improvements in the NFS server and client, and reliability, availability, and serviceability (RAS) cleanup in the NFS client. The z/OS V1.11 enhancements continue to support the DFSMS enhancements.

2

DFSMSdfp enhancements in V1R11

In this chapter, we discuss new and changed functions in DFSMSdfp. The following topics are covered:

- High Performance FICON
- O/C/EOV performance and RAS improvement
- DADSM: Save job, step, and time at data set creation

2.1 High Performance FICON for System z

Enhancements have been made to the System z10® architecture and the FICON protocol. These enhancements optimize I/O for online transaction processing workloads. When exploited by the FICON channel, z/OS, and the IBM System Storage DS8000 control unit, High Performance FICON for System z (zHPF) helps reduce processing impact and improve performance in the following ways:

- The maximum number of I/Os can be improved up to 100% for small data transfers that are able to exploit zHPF.
- Production workloads with a mix of data transfer sizes might see up to 30 to 70% of FICON I/Os utilizing zHPF, saving 10 to 30% channel utilization.
- Sequential I/O transferring is less than a single track size (48 KB per I/O).
- Data accessed by DB2, Partition Data Set Extended (PDSE), Virtual Storage Access Method (VSAM), and extended format SAM can provide a benefit to zHPF and the new channel programs built by Media Manager.

2.1.1 Requirements

The requirements are as follows:

- Only available on System z10 and FICON Express2 and Express4.
- z/OS V1R8 and higher (or V1R7 with life cycle extension 5637-A01).
- IBM System Storage DS8000 Release 4.1 (LMC level 5.4.1.xx [bundle version 64.1.x.x]) or later).
- Priced license feature with a monthly maintenance charge.
- Support for zHPF is an optional feature of the DS8000 series Turbo Models (931, 932, and 9B2) and is available with the High Performance FICON licensed feature indicator.

2.2 O/C/EOV: Performance and RAS improvement

A number of RAS improvements have been made in DFP along with two performance improvements.

2.2.1 Performance improvements

The performance improvements are listed below,

Concatenation tape positioning

For IBM Tape Library 3590 and later, technology support was added to use fast positioning of SL and NL cartridges via Position Relative commands to the drive during EOV concatenation processing. EOV concatenation tape positioning involves moving the tape from tape mark to tape mark. This starts and stops tape motion at each tape mark. The measure of this start/stop time is referred to as *backhitch*. Backhitch time is longer for later technology tape drives due in part to the faster tape transport speed across the heads that the drive requires to process the tape. It takes more time to get to processing speed. With these improvements, the impact of backhitch time is reduced.

DSAB chain searching performance

DSAB chain searching performance has been improved by exploiting the better GETDSAB hash algorithm.

2.2.2 RAS improvements

This section describes the RAS improvements in z/OS V1.11 DFSMSdfp.

IFGSMS14/15 mapping macro DSECTs

Mapping macros for SMF types 14 and 15 have been enhanced to include DSECTs for sections and segments where beneficial. Prior to z/OS V1.11, the mapping macro contains no DSECTs, making most of the provided mapping of the sections and segments difficult to use.

DSECT=YES|NO will be added. DSECT=NO will be the default, which results in the same macro generation that is currently available. DSECT=YES will cause a DSECT before each of the following sections (&namexxx is the DSECT name, where &name is the user specified name on the IFGSMF14 macro invocation, otherwise, it will default to IFGSMF):

- Main (&name)
- DCB/DEB segment (&nameTDDEX)
- UCB segment (&nameUCB)
- Hiperbatch section (&nameHBTCH)
- ISAM extension segment (&nameISAMX)
- Extended information segment (&nameXIX)
- Header for each extended information segment (&nameXIC)
- Each of the current seven extended information segments (&nameCSB, &nameSCI, &nameSPI, &nameA4I, &nameADI, &namePSI, and &nameENC)

Automatic recovery for IEC708I

z/OS V1.11 provides automatic recovery for IEC708I to prevent a loop from reading the same volume or set of volumes when duplicate volume serial numbers are detected in the JFCB volume list.

Additionally, an abend option is provided via the Label Anomaly Installation Exit for tape multivolume conditions that currently result in IEC709I, IEC710I, IEC711I, and IEC712I error codes. Four new bits will be defined for entry into the Label Anomaly exit. They will represent the following four new anomalies:

- Unexpected tape volume sequence number (TEPAOUTSEQ). The next volume that will be processed has a sequence number that is not what is expected based on the current volume's sequence number.

  Return code C will result in an abend, IEC709I, and no further action.

- Last tape volume missing (TEPARDFMLV). The last processed volume was specified, but it has an EOV label, not an EOF label.

  Return code C will result in an abend, IEC710I, and no further action.

- First volume missing when reading backward (TEPARDBMLV). This is the last volume in the list. It is standard label but does not have an EOF label.

  Return code C will result in an abend, IEC711I, and no further action.

- First volume missing when reading forward (TEPARDFMFV). The first volume in the list does not begin with logical volume sequence number 1.

  Return code C will result in an abend, IEC712I, and no further action.

The new LABAN entry indicator flags in the IFGTEP macro are shown in Figure 2-1.

```
*        2 TEPAFLAG3  BIT(8),    ! Tape problem flag 3
*        3 TEPABYRACF BIT(1),    ! Bypass RACDELETE option active.
*        3 TEPAEXSKIP BIT(1),    ! Exit requests no RACDELETE
*        3 TEPAABEND  BIT(1),    ! Exit requests abend when RC08
*        3 TEPAOUTSEQ BIT(1),    ! Iec709i vols out of sequence .....NEW
*        3 TEPARDFMLV BIT(1),    ! Iec710i read forwd miss last vol..NEW
*        3 TEPARDBMLV BIT(1),    ! Iec711i read bckwd miss last vol..NEW
*        3 TEPARDFMFV BIT(1),    ! Iec712i read forwd miss 1st vol...NEW
*        3 *          BIT(1),    ! Reserved.....................CHANGED
```

Figure 2-1 LABAN entry indicator flags in the IFGTEP macro

Return Code C will be recognized only for the above new anomalies and results in either abend 413 rc08 for OPEN or abend 637 rc2C for EOV. These are the same existing abends used for other LABAN abend conditions

Module updates to use branch relative addressing

Selected OCE CSETS have been converted to use Branch Relative addressing. This reduces complexity in resolving CSECT addressability difficulties that were brought about by adding new functions.

Additional module entries for OCE component trace

The OCE component trace has been enhanced to include trace entries for IFG019RM, IFG019RN, IFG019RO, IFG019RF, IFG055ZZ, and IFG019RQ. In addition, the common UCB segment is included in all trace entries.

DASD data set expiration NEVEREXPIRE

The DASD data set output expiration date processing has been enhanced to externalize the expiration date in message IEC507D, where the expiration date never expires. This provides an installation automation option to enforce dates that never expire by automating the reply to IEC507D. The new text variation is:

```
IEC507D  E dev,ser,jjj,sss[,dsn][,NEVEREXPIRE]
```

2.3 DADSM: Saving job, step, and time at data set creation

With z/OS V1R11 DFSMS, DADSM records the job name, step name, and time of creation of a data set when a data set is created. This data is saved in Format 9 DSCB, which was introduced in z/OS V1R10 to provide EAV support.

Currently, only data sets that are EAS eligible can have a Format 9 DSCB. With z/OS V1R11, this means only VSAM and extended format sequential access method (EFSAM) type data sets are eligible. For more details about the new extended attribute DCSBs, see 4.2.7, “New format DSCBs for EAV volumes” on page 92.

2.3.1 Format 9 DSCB

The Format 9 DSCB was introduced in V1R10 and has all the information z/OS needs to record the attributes of a data set in the EAS of an EAV. The Format 9 DSCB can point to one or more Format 3 DSCBs, as shown in Example 2-1. The Format 8 DSCB is used in this scenario strictly as an indication that a Format 9 DSCB with all the EAS information has been inserted between the Format 1 (Format 8) DSCB and the Format 3 DSCB chain.

Note: In z/OS V1R11,data set attributes are recorded in the Format 9 DSCB. These fields indicate the job and step name and time since midnight of the day the data set was created. The creation date is still kept in the Format 1 DSCB.

In future releases, additional Format 9 DSCBs may be chained between the subtype 1 and any Format 3 DSCBs.

The Format 9 DSCB exists only for EAS-eligible data sets (VSAM (V1R10) and extended-format sequential data sets (V1R11)), and it contains the following EAV information:

- The format identifier is x'F9'.
- A subtype field.
- In the first EAV release, the subtype is 1.
- In future releases, additional subtypes may be added.
- Track addresses, which point directly to up to ten Format 3 DSCBs.
- All the Format 3 DSCBs can be read with one channel program.
- A 20-byte field, DS9ATRV1, that IBM is reserving for vendors. IBM will not specify or monitor its content.

2.3.2 Format 9 DSCB field layout

The fields shown in Example 2-1 are specifically used to record the job name, step name, and time of the data set creation. The mapping macro IECSDSL1 maps the Format 9 DSCB, along with the others.

Example 2-1 Format 9 DSCB mapping

```
OFFSET
DEC    HEX   TYPE        LENGTH  NAME (DIM)     DESCRIPTION
====== ===== =========  =======  ============== ================
  0      (0) STRUCTURE      140  IECSDSL9       Format 9 DSCB Mapping
  0      (0) CHARACTER        1   DS9KEYID      Key identifier
  1      (1) CHARACTER        1   DS9SUBTY      Subtype
  2      (2) UNSIGNED         1   DS9NUMF9      Number of Format 9 DSCBs for
                                                 this data set. Valid only in
                                                 the first Format 9 DSCB
  3      (3) CHARACTER       41   DS9ATRI1      Reserved for IBM use only
  3      (3) CHARACTER        1    DS9FLAG1     Format 9 DSCB flag byte 1
             1... ....              DS9CREAT    Format 9 DSCB build by create

   The following three fields valid when DS9CREAT is set
  3      (3) CHARACTER        8    DS9JOBNAME   Job name used to create the
                                                 data set described by its
                                                 Format 8 DSCB
```

```
 11      (B) CHARACTER        8    DS9STEPNAME  Step name used to create the
                                                data set described by its
                                                Format 8 DSCB
 19     (13) CHARACTER        6    DS9TIME      Number of microseconds
                                                since midnight that the data
                                                set described by its Format 8
                                                DSCB was created. See creation
                                                date field, DS1CREDT, for the date
 25     (19) CHARACTER       18    *            Not used
 44     (2C) CHARACTER        1   DS9FMTID      Format identifier x'F9'       . .
.
. . .
```

Figure 2-2 shows the job name, step name, and time for extended format sequential data. It is not necessary for the actual data set to reside in the EAS portion of an EAV, just that it is EAS eligible. The figure shows two data sets, but of those two, only CVERNON.EAVTEST.EFSAM1 resides in EAS, yet both have an associated Format 9 DSCB in which the job name, step name, and time are saved.

```
 SYSTEMS SUPPORT UTILITIES---IEHLIST                              PAGE    1
DATE: 2009.219  TIME: 18.19.56
               CONTENTS OF VTOC ON VOL GKDD65  <THIS IS AN SMS MANAGED VOLUME>
---------------DATA SET NAME----------------  SER NO  SEQNO  DATE.CRE  DATE.EXP  DATE.REF  EXT DSORG RECFM OPTCD BLKSIZE
CVERNON.EAVTEST.EFSAM0                        GKDD65      1  2009.219    00.000    00.000    1  PS   FB     00    27920
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST BLK(TTTT-R) DIR.REM  F2 OR F3(C-H-R)  DSCB(C-H-R)
S  BE        80           CYLS             1                                                                  5   0  15
EATTR   JOB        STEP      CREATE TIME
OPT     CVERNON4   ALXFS     18:08:58.494928
            EXTENTS  NO  LOW(C-H)   HIGH(C-H)
                      0    21  0       21 14
                           ----UNABLE TO CALCULATE EMPTY SPACE.

---------------DATA SET NAME----------------  SER NO  SEQNO  DATE.CRE  DATE.EXP  DATE.REF  EXT DSORG RECFM OPTCD BLKSIZE
CVERNON.EAVTEST.EFSAM1                        GKDD65      1  2009.219    00.000    00.000    1  PS   FB     00    27920
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST BLK(TTTT-R) DIR.REM  F2 OR F3(C-H-R)  DSCB(C-H-R)
S  BE        80           CYLS             21                                                                 5   0  17
EATTR   JOB        STEP      CREATE TIME
OPT     CVERNON4   ALXFS     18:08:58.538451
EXTENTS  NO       LOW(C-H)      HIGH(C-H)
          0      65562  0       65582 14
                           ----UNABLE TO CALCULATE EMPTY SPACE.
```

Figure 2-2 Formatted LISTVTOC extract showing the creation job name, step name, and time

Figure 2-3 shows a hex dump of the Format 8 and Format 9 DSCBs for the same data sets in Figure 2-2 on page 20. Notice that the hexadecimal dump Format 9 DSCB shows the same job name and step name in EBCDIC character codes used in Figure 2-2 on page 20. Both the job name and creation time are highlighted in bold font and the step name is the field in between those two elements.

```
 SYSTEMS SUPPORT UTILITIES---IEHLIST                              PAGE    3
DATE: 2009.219  TIME: 18.19.56
               CONTENTS OF VTOC ON VOL GKDD65  <THIS IS AN SMS MANAGED VOLUME>
 LINE 1 = DSNAME               0 . . . 4 . . . . 9 . . . .14 . . . .19 . . . .24 . . . .29 . . . .34 . . . .39 . . . .
 LINE 2 = 44 . . . .49 . . . .54 . . . .59 . . . .64 . . . .69 . . . .74 . . . .79 . . . .84 . . . .89 . . . .94 . . . .
 LINE 3 = 99 . . . 104 . . . 109 . . . 114 . . . 119 . . . 124 . . . 129 . . . 134 . . . 139          DSCB ADDR (CCHHR)
CVERNON.EAVTEST.EFSAMO
          F8C7D2C4C4F6F500016D00DB000000010002C9C2D4D6E2E5E2F2404040404000000094000000400090006D10005000000080C000000100
          00000000000081000015000001500OE000000000000000000000000000000000000000000005000010          000500000F
 FORMAT 9 DSCB                 09010180C3E5C5D9D5D6D5F4C1D3E7C6E2404040000F367A4FD00000000000000000000000000000000000
          F900000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
          00000000000000000000000000000000000000000000000000000000000000000000000000000000000000          0005000010
CVERNON.EAVTEST.EFSAM1
          F8C7D2C4C4F6F500016D00DB000000010002C9C2D4D6E2E5E2F2404040404000000094000000400090006D10005000000080C000001500
          00000000000081000001A0010002E001E000000000000000000000000000000000000000005000012          0005000011
 FORMAT 9 DSCB                 09010180C3E5C5D9D5D6D5F4C1D3E7C6E2404040000F367AF9D300000000000000000000000000000000000
          F900000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
          00000000000000000000000000000000000000000000000000000000000000000000000000000000000000          0005000012
```

Figure 2-3 Hex dump extract of Format 9 DSCB showing job, step, and time

3

DFSMS Object Access Method enhancements

In this chapter, we discuss the following DFSMS Object Access Method (OAM) enhancements:

- OAM 2 GB Object Support
- OAM archive retention enhancements
- A wildcard for OSREQ QUERY
- A utility used to modify collection defaults
- CBRUXSAE enhancement

3.1 OAM 2 GB object support

Prior to z/OS V1.11, the maximum object size of an object on tape or optical media was 256 MB. In z/OS V1.10, OAM implemented 2 GB Object Support, which enabled applications to store objects up to 2 GB in size in DB2 DASD.

Note: 2 GB is technically 2^{31}, which is 2,147,483,648 bytes.

In z/OS V1.11, this enhancement extend this support to the tape tier of the OAM storage hierarchy (Stage 2). This support provides full functionality for objects up to 2 GB in size on DASD and tape. However, the maximum object size for objects residing on optical media still remains at 256 MB.

3.1.1 Overview

In a z/OS 31-bit (also referred to as 2 GB) address space, the maximum value that can be addressed is 2,147,483,647. Originally, it was the designer's intent to support an actual 2 GB object size in OAM, but to better accommodate the DB2 buffer limitation and other constraints in the OAM storage hierarchy, the maximum value was reduced slightly below 2 GB.

So, although the support is still referred to as OAM 2 GB object support, the actual OAM maximum object size in z/OS V1.11 is 2000 MB (2 x 1000 x 1024 x 10024 = 2,097,152,000 bytes).

By handling object sizes greater than 256 MB in z/OS V1.11, there are a number of potential benefits:

- New applications that could not have been possibly or easily enabled because of the 256 MB maximum object size restriction now can be enabled.
- Existing applications can be enhanced:
  - Some applications bundle multiple entities together in a single OAM object. A common example is a set of scanned check images. Applications maintain metadata that identifies where each unique entity is within the OAM object; this metadata could be kept in a separate database outside of OAM or it could be incorporated into the object itself. For example, the first *n* bytes of the object could be a directory or index into the content within the object (OAM has no knowledge of the structure or content of the object and just views it as a byte stream, which allows for many possibilities from an application perspective). It should be noted that after the object has been stored, the OSREQ RETRIEVE function can be used to read just a portion of the object, which allow applications to use their metadata to identify where within the OAM object the entity of interest resides and then only retrieve that portion. With object sizes greater than 256 MB, applications can bundle even more entities into a single OAM object, which overall might result in less total objects that the application tracks normally.
  - Some applications might have entities that exceed 256 MB in size and might have had to artificially divide these entities into chunks of up to 256 MB each so that they can be stored in OAM. This implies that the application must maintain metadata about these individual chunks in order to recombine them to reform the entity when needed. With object sizes greater than 256 MB, applications might not need to artificially divide large objects and can instead store them as a single OAM object, thus reducing the metadata that the application must maintain.

Since 2000 MB is practically 2 GB, it is fairly clear that an application would not be able to acquire 2000 MB of virtual storage for an OAM buffer in a 31-bit address space. This means that either the application and OAM needs to exploit 64-bit addressing capabilities in z/OS or a new mechanism must be devised.

The OSREQ application programming interface was enhanced in V1.10 so that objects greater than 256 MB and up to 2000 MB in size can be stored, in parts, sequentially to the “DASD” level of the OAM storage hierarchy.

Objects greater than 256 MB are only supported on the DB2/DASD level of the OAM storage hierarchy; there is no support for objects greater than 256 MB in the remainder of the OAM storage hierarchy (for example, optical or tape sublevels).

This implies that:

- The OAM Storage Management Component (OSMC) cannot transition objects greater than 256 MB from the DB2/DASD level of the OAM storage hierarchy to any other level of the OAM storage hierarchy, and because objects greater than 256 MB cannot exist on any other level of the OAM storage hierarchy, there is no OSMC transition to the DB2/DASD level of the OAM storage hierarchy.
- No backups can be made of objects greater than 256 MB to optical or tape.

In z/OS V1.11, 2 GB Object Support (Phase 2) extends the 2000 MB object size to tape. This support provides full functionality (OSMC transitions, backups, recovery, and movevol) for objects up to 2000 MB in size on DASD and tape. using the new STOREBEG, STOREPRT, and STOREEND functions that were added to the OSREQ API in z/OS V1.10 and the new optional STIMEOUT keyword added to the STOREBEG function.

This enhancement provides the following functionality for objects up to 2000 MB in size:

- Applications can store and retrieve objects directly to and from DASD or tape via OSREQ.
- OSMC can transition these objects between DASD and tape.
- OSMC can expire these objects from DASD or tape.
- OSMC can create backup copies of these objects on tape media.
- OSMC Move Volume Utility can be used to move these objects from a tape volume.
- OSMC Object Recovery Utility can be used to recover these objects if lost or damaged.
- OSMC Volume Recovery Utility can be used to recover these objects on a tape volume.
- OSMC can recall these objects from tape to DASD for a specified number of days.
- OSMC RECYCLE command can be used to move these objects from tape volumes.

3.1.2 OSREQ macro Application Program Interface

A new capability to store an object greater than 256 MB in a series of parts was implemented by making a change to the OAM OSREQ application programming interface. OAM operations using OSREQ are bounded by the ACCESS and UNACCESS functions; within the bounds of an ACCESS and UNACCESS, an application can use existing STORE, RETRIEVE, QUERY, CHANGE, and DELETE functions.

The OSREQ macro provides the ACCESS, CHANGE, DELETE, QUERY, RETRIEVE, STORE, and UNACCESS functions. The following functions were added to the OSREQ macro in V1.10 to store objects greater than 256 MB and up to 2000 MB in size:

- STOREBEG: Begin a store operation for an object.
- STOREPRT: Store an individual part of an object.
- STOREEND: End a store operation for an object.

In z/OS V1.11, a new optional STIMEOUT keyword is provided for the OSREQ STOREBEG request to specify an interval after which OAM can free resources being held for the store sequence on behalf of the application. This keyword would be used primarily by applications that anticipate a long delay between individual store sequence requests as a mechanism to provide a “keep alive” capability for the resources. This keyword does not apply to the DASD level of the OAM storage hierarchy, and is therefore ignored when the store request is directed to DASD.

Note:

- The STOREBEG, STOREPRT, and STOREEND functions cannot be used to store objects less than or equal to 256 MB.
- The existing OSREQ STORE function can only be used for objects less than or equal to 256 MB.
- As in z/OS V1.10, you cannot retrieve an entire object at once through the OSREQ programming interface when the object size is greater than 256 MB, but you can use the existing OSREQ RETRIEVE function to retrieve one or more or all of the parts of the object with each part being up to a maximum of 256 MB in size.

OSREQ STOREBEG syntax

Figure 3-1 shows the OSRES STOREBEG macro syntax.

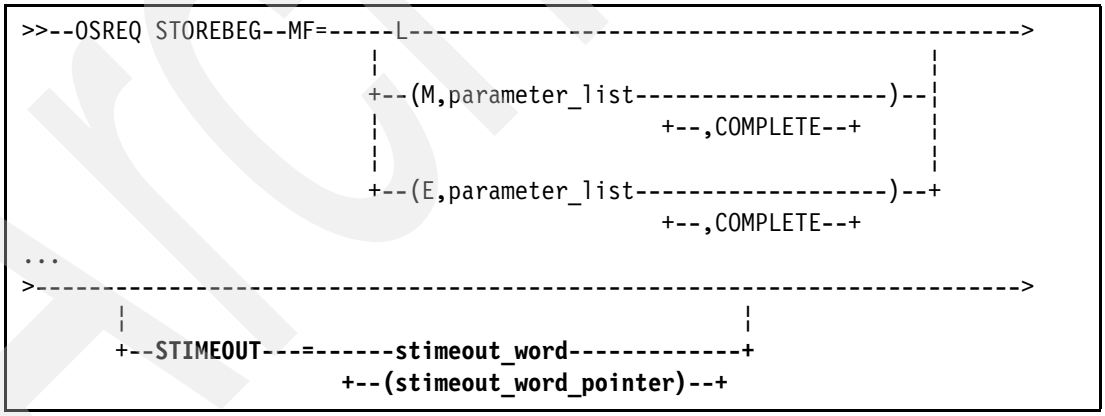


Figure 3-1 STOREBEG STIMEOUT syntax

The STIMEOUT keyword is only used by a STOREBEG request and is ignored by all other requests.

The stimeout_word specifies the maximum interval in seconds between STOREBEG, STOREPRT, and STOREEND requests that OAM should wait before OAM will assume that there will be no more activity for this store sequence and will free resources held on behalf of this store sequence. OAM will normally attempt to detect cases when there has been no activity from the application during a store sequence in progress and free limited resources that are being held on behalf of the application. This can occur if the application abnormally ends, encounters an error, or otherwise does not normally complete the individual function calls in a store sequence. Specify a value if there will be an unusually long delay between the requests in a store sequence to ensure that OAM does not free resources used for the store sequence. Note that this interval does not apply to the DASD level of the OAM storage hierarchy.

Valid values for the number of seconds that can be specified are 0-9999. If the STIMEOUT keyword is not specified (or if the STIMEOUT value is specified as zero), then the STIMEOUT value defaults to 300 seconds (5 minutes).

Note:

- These keywords are required and therefore they must be specified on the MF=E form if it indicates COMPLETE or they must be specified on at least one of the forms if the MF=E does not indicate COMPLETE.
- The size specified must be the exact total size of the object.

3.1.3 OSREQ TSO/E command processor

The TSO/E OSREQ command provides the STORE, RETRIEVE, QUERY, CHANGE, and DELETE functions, and for each of these functions, the required OSREQ ACCESS and OSREQ UNACCESS is done implicitly by the TSO/E OSREQ command. The TSO/E OSREQ command also provides a COMPARE function. The OSREQ TSO/E command can be used to verify your OAM installation.

In z/OS V1.10 the STOREBEG, STOREPRT, and STOREEND functions are not externalized to the OSREQ TSO/E command. The reason is that these functions all need to be performed within the bounds of a single OSREQ ACCESS and UNACCESS. These functions also require the new “store sequence” token from the STOREBEG to be provided for subsequent STOREPRT and STOREEND invocations, which would not be possible through the structure of the OSREQ TSO/E command. Instead the OSREQ TSO/E command’s existing STORE function has been changed to internally invoke the STOREBEG, STOREPRT, and STOREEND functions of the OSREQ macro. As a result, the OSREQ TSO/E command STORE function can be used to store objects greater than 256 MB (but this is only “dummy data”).

The OSREQ TSO/E command’s RETRIEVE and COMPARE functions have been changed to process objects greater than 256 MB. Furthermore, an optional DATA keyword on the OSREQ TSO/E RETRIEVE and COMPARE functions can be used to display the actual object data on the terminal. For objects greater than 256 MB, using the DATA keyword requires the specification of an offset and length to identify a “part” of the object to display, because it is not practical to attempt to display the entire contents of an object greater than 256 MB.

3.1.4 OSREQ application interface return codes and reason codes

New return and reason code combinations are provided to identify errors such as programming or environmental conditions that might occur as a result of the implementation of the OAM 2 GB Object Support (Stage 2).

The new and changed reason codes shown in Table 3-1 are documented in the *z/OS V1R11.0 Object Access Method Application Programmer's Reference,* SC35-0425*,* and in *z/OS V1.11 DFSMSdfp Diagnosis,* GY27-7618.

Table 3-1 New return codes and reason codes

Return codes	Reason codes	Descriptions
8	24 xx 15 01	STIMEOUT in unusable storage.
	24 xx 15 02	STIMEOUT value invalid.
	38 xx 0A zz	Buffer unavailable for STOREPRT or STOREEND. A possible cause could be that the STIMEOUT value specified on the OSREQ STOREBEG is too low.
16	E4 xx 02 zz	Buffer not acquired for STOREBEG. Ensure that the system is configured to make sufficient 64-bit addressable virtual memory available to the OAM address space.

3.1.5 RETCODE2 for OSREQ for STOREEND

return_code2_word is an optional keyword that can be used to determine if OAM has scheduled additional processing for an OSREQ request. The information returned in RETCODE2 is dependent on the OSREQ function (RETRIEVE, STORE, or STOREEND) requested.

For an OSREQ STORE or STOREEND request, RETCODE2 specifies whether this STORE or STOREEND request resulted in scheduling an immediate backup copy to be written for this object.

return_code2_word is only valid when the STORE or STOREEND is successful.

A RETCODE2 value of 16 is modified to only apply to immediate backup copies directed to optical.

The RETCODE2 codes are shown in Table 3-2.

Table 3-2 RETCODE2 table

RETCODE2	Description
0	An immediate backup copy request was successfully scheduled.
4	An immediate backup copy request is not required.
8	An attempt to schedule an immediate backup for this object was not successful because OSMC is not up and running.
14	An attempt to schedule an immediate backup for this object was not successful due to an unexpected scheduling error.
16	An immediate backup to optical is not supported for STOREEND.

3.1.6 OAM SMF record type 85

The fields shown in Table 3-3 are new or changed in the SMF record.

Table 3-3 New or changed fields in SMF type 85

Offset	Name	Type	Descriptions
A0	ST2FLGS	Binary	Flag for OSREQ STORE 12: Unused. 13: Deletion-hold on. 14: Retention-protect on. 15: Deletion-protect on 16: Event-based-retention on.
A0	ST5FLGS	Binary	Flag for OSREQ CHANGE 3: RETPD = -1. 4: RETPD = -2. 5: RETPD = x'7FFFFFFF'. 6: EVENTEXP. 7:DELETEHOLD = HOLD. 8:DELETEHOLD = NOHOLD.
A0	ST10FLGS	Binary	Flags for OSREQ STOREEND 1: Unused 2: When on, the object is stored to Tape. 3: Unused. 4: Unused. 5: Shelf-resident tape volume mounted. 6: Library-resident tape volume mounted. 7: Tape already mounted. 8: Immediate backup copy is scheduled. 9: Stored to LOB storage structure. 10: Stored on sublevel 1 volume. 11: Stored on sublevel 2 volume. 12: CANCEL=YES specified. 13: Deletion-hold. 14: Retention-protected. 15: Deletion-protected. 16: Event-based-retention.
B8	ST1STOUT	Binary	STIMEOUT value. Specifies the maximum interval in seconds between STOREBEG, STOREPRT, and STOREEND requests that OAM should wait before OAM assumes that there will be no more activity for this store sequence. Valid for subtype 8.
BC	ST1OLRD	EBCDIC	Old Last Ref Date prior to OSREQ CHANGE or RETRIEVE.
C6	ST1NLRD	EBCDIC	New Last Ref Date after OSREQ CHANGE or RETRIEVE.

3.1.7 Sysplex

An OAMPLEX has existing restrictions for transaction shipping of requests from one OAM member of the SYSPLEX to another OAM member of the SYSPLEX. The restrictions are mainly intended to avoid sending large amounts of data between systems so that no shipping of requests will occur for objects greater than 50 MB. For objects less than or equal to 50 MB, only the following requests will be shipped:

- Read from tape when the volume is currently mounted on another OAM member of the SYSPLEX
- Read from optical on another OAM member of the SYSPLEX to which the optical device is physically attached
- Write to optical on another OAM member of the SYSPLEX to which the optical device is physically attached

With 2 GB Object Support (Stage 2) where even larger object sizes (greater than 256 MB and up to 2000 MB in size) can be processed, these existing restrictions will continue to apply.

3.1.8 How to implement

If you review Chapter 3. “Migrating, Installing, and Customizing OAM”, in *z/OS V1R11.0 DFSMS Object Access Method Planning, Installation, and Storage Administration Guide for Object Support*, SC35-0426, you will find the information that is necessary to create the structure that you will need for OAM and DB2. SYS1.SAMPLIB contains the jobs.

In our test, as we were using OAM to write objects of 2000 MB in size, it was necessary to change the MOS= parameter. This is set in SYS1.PARMLIB(IEFSSNxx), and you need to load the initial program to validate this configuration. Figure 3-2 shows the values we have used. Change those values to ones for your specific policies.

```
SUBSYS SUBNAME(OAM1)
  INITRTN(CBRINIT)
  INITPARM('MSG=EM,MOS=2000,LOB=A,DP=N,QB=Y')
```

Figure 3-2 Parameters used in our test

We customize SYS1.PARMLIB(CBROAMxx) as shown in Figure 3-3. We use PARMLIB member xx="70", so you see this number in our examples.

```
 OAMXCF OAMGROUPNAME(OAMTEST)                    /* OAM XCF         */
 ONLYIF SYSNAME(SC70)
        OAMXCF OAMMEMBERNAME(OAM70)              /* MEMBER NAME     */
 ONLYIF SYSNAME(SC70)
        SETOPT ABALL(BACKUP1)                    /* ACC BKP1 IF UNAV*/
 SETOAM DEMOUNTWAITTIME(90)                      /* GLOBAL LEVEL    */
        MAXRECYCLETASKS(1)
        MAXTAPERETRIEVETASKS(2)
        MAXTAPESTORETASKS(2)
        MOUNTWAITTIME(10)
        TAPEDISPATCHERDELAY(40)
        TAPEFULLTHRESHOLD(50000)
        DSNWITHSGNAME
        OAMSCRATCHSYNCH(ENABLED)
        TAPERECYCLEMODE(MVSSCRATCH)
 SETOAM STORAGEGROUP(TOAMBK1                     /* DEFAULT OBJ BKP */
         TAPEUNITNAME(ATL3)
        SGMAXTAPERETRIEVETASKS(1)
        SGMAXTAPESTORETASKS(1)
        TAPEDRIVESTARTUP(5000)
        TAPEPERCENTFULL(98))
SETOAM STORAGEGROUP(TOGRP1
        TAPEUNITNAME(ATL3)
        L2TAPEUNITNAME(ATL3)
        SGMAXTAPERETRIEVETASKS(1)
        SGMAXTAPESTORETASKS(2)
        TAPEDRIVESTARTUP(5000)
        TAPEPERCENTFULL(98))
       STORAGEGROUP(TOGRP1B1
        TAPEUNITNAME(ATL3)
        SGMAXTAPERETRIEVETASKS(1)
        SGMAXTAPESTORETASKS(1)
        TAPEDRIVESTARTUP(5000)
        TAPEPERCENTFULL(98))
       STORAGEGROUP(TOGRP1B2
         TAPEUNITNAME(ATL3)
         SGMAXTAPERETRIEVETASKS(1)
         SGMAXTAPESTORETASKS(1)
         TAPEDRIVESTARTUP(5000)
         TAPEPERCENTFULL(98))
 SETOSMC FIRSTBACKUPGROUP(TOAMBK1)
         MAXRECALLTASKS(2)
         RECALLALL(1)
        STORAGEGROUP(TOGRP1
         FIRSTBACKUPGROUP(TOGRP1B1)
         SECONDBACKUPGROUP(TOGRP1B2)
         RECALLOFF(ON))
```

Figure 3-3 CBROAMxx customization for our scenario

After you load the initial program, you should issue the D SMS,OAM command. You will see the information shown in Figure 3-4.

```
 DB2 SSID:  D9DG
  XCF GROUP NAME: OAMTEST
  XCF MEMBER NAME: OAM70
  CBROAM: 70
  OAM1 Parms: TIME=LOC  MSG=EM  UPD=Y  QB=Y
              MOS=2000  OTIS=N  LOB=A  DP=N
******************************** BOTTOM OF DATA *
```

Figure 3-4 Partial result of the D SMS,OAM command

It is important the value of MOS be 2000.

In regards to location, the letter provides you with information about where your object is. For example:

- The D object resides on DASD.
- The R object resides on DASD in recalled mode.
- The T object resides on a tape sublevel 1 volume.
- The U object resides on a tape sublevel 2 volume.
- The blank object resides on an optical volume.

For more details, see 3.6.4, “CBR0443I” on page 78.

3.1.9 Storing a 2 GB object to a tape: Sample output

We run job CBRSAMIV to create an object of 2 GB on tape. The JCL we used is shown in Example 3-1.

Example 3-1 Sample JCL writing to tape

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 OSREQ STORE    TOGRP1.OBJTEST TOGRP1.OBJTEST LENGTH(2097152000)
 OSREQ QUERY    TOGRP1.OBJTEST TOGRP1.OBJTEST
 LISTCAT ENTRIES('TOGRP1.OBJTEST') ALL
 OSREQ CHANGE   TOGRP1.OBJTEST TOGRP1.OBJTEST RP(365)
 OSREQ QUERY    TOGRP1.OBJTEST TOGRP1.OBJTEST
 OSREQ RETRIEVE TOGRP1.OBJTEST TOGRP1.OBJTEST COMPARE VIEW(PRIMARY)
 OSREQ DELETE   TOGRP1.OBJTEST TOGRP1.OBJTEST
/*
***************************** Bottom of Data ****************************
```

The output is shown in Example 3-2.

Example 3-2 CBRSAMIV sample output

```
******************************** Top of Data ********************************
1READY
  OSREQ STORE    TOGRP1.OBJTEST TOGRP1.OBJTEST LENGTH(2097152000) ❶
 OSREQ STORE successful. Return code = 00000000, reason code = 00000000.
 OSREQ STORE response time is 76946 milliseconds.
 OSREQ STORE data rate is 26615 kilobytes/second.
 READY
  OSREQ QUERY    TOGRP1.OBJTEST TOGRP1.OBJTEST
 OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
 OSREQ QUERY response time is 5 milliseconds.
 ------------------------------------------------------------------------
 Collection name         = TOGRP1.OBJTEST
 Object name             = TOGRP1.OBJTEST
 Object size             = 2097152000
 Creation date           = 2009-08-28
 Creation timestamp      = 17.21.12.294304
 Last referenced date    = 0001-01-01
 Expiration date         = 0001-01-01
 Management class        = OBJTAPE
 Storage class           = OBJTAPE
 Primary retrieve key    = 'E3C8E2F0F0F100000004'X
 Backup retrieve key     = '00000000000000000000'X
 Backup2 retrieve key    = '00000000000000000000'X
 Estimated retrieve time = 60000
 Location                = T ❷
 Pending action date     = 2009-08-28
 Status flags            = 0
 Retention protect date  = 0001-01-01
 Deletion-protected      = N
 READY
  LISTCAT ENTRIES('TOGRP1.OBJTEST') ALL
 NONVSAM ------- TOGRP1.OBJTEST
      IN-CAT --- UCAT.VSBOX01
      HISTORY
        DATASET-OWNER-----(NULL)     CREATION--------2009.240
        RELEASE----------------2     EXPIRATION------9999.999
        ACCOUNT-INFO-----------------------------------(NULL)
      OAMDATA
        DIRECTORYTOKEN----TOGRP1
      SMSDATA
        STORAGECLASS ----OBJTAPE     MANAGEMENTCLASS--OBJTAPE
        DATACLASS --------(NULL)     LBACKUP ---XXXX.XXX.XXXX
      ASSOCIATIONS--------(NULL)
      ATTRIBUTES
 READY
  OSREQ CHANGE   TOGRP1.OBJTEST TOGRP1.OBJTEST RP(365)
 OSREQ CHANGE successful. Return code = 00000000, reason code = 00000000.
 OSREQ CHANGE response time is 4 milliseconds.
 READY
  OSREQ QUERY    TOGRP1.OBJTEST TOGRP1.OBJTEST
 OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
 OSREQ QUERY response time is 5 milliseconds.
```

```
 ************************************************************************
 Collection name         = TOGRP1.OBJTEST
 Object name             = TOGRP1.OBJTEST
 Object size             = 2097152000
 Creation date           = 2009-08-28
1Creation timestamp      = 17.21.12.294304
 Last referenced date    = 0001-01-01
 Expiration date         = 2010-08-28
 Management class        = OBJTAPE
 Storage class           = OBJTAPE
 Primary retrieve key    = 'E3C8E2F0F0F100000004'X
 Backup retrieve key     = '00000000000000000000'X
 Backup2 retrieve key    = '00000000000000000000'X
 Estimated retrieve time = 60000
 Location                = T ❷
 Pending action date     = 2009-08-28
 Status flags            = 0
 Retention protect date  = 0001-01-01
 Deletion-protected      = N
 READY
  OSREQ RETRIEVE TOGRP1.OBJTEST TOGRP1.OBJTEST COMPARE VIEW(PRIMARY)
 OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
 OSREQ QUERY response time is 5 milliseconds.
 OSREQ RETRIEVE successful. Return code = 00000000, reason code = 00000000.
 OSREQ RETRIEVE response time is 108213 milliseconds.
 OSREQ RETRIEVE data rate is 18924 kilobytes/second.
 Data comparison for object TOGRP1.OBJTEST TOGRP1.OBJTEST successful.
 READY
  OSREQ DELETE   TOGRP1.OBJTEST TOGRP1.OBJTEST
 OSREQ DELETE successful. Return code = 00000000, reason code = 00000000.
 OSREQ DELETE response time is 20 milliseconds.
 READY
 END
****************************** Bottom of Data ******************************
```

Note the value of 2000 MB (❶) in Example 3-2 on page 33. We have a “Location” as “T” (❷), which means that the location is a tape. You can chain commands together to accomplish your goals.

3.1.10 Storing a 2 GB object to a DASD: Sample output

We use the JCL shown in Example 3-3.

Example 3-3 Sample JCL writing on DASD

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
 OSREQ STORE    TOGRP1.TESTXXX TOGRP1.TESTXXX LENGTH(2097152000)
 OSREQ QUERY    TOGRP1.TESTXXX TOGRP1.TESTXXX
 LISTCAT ENTRIES('TOGRP1.TESTXXX') ALL
 OSREQ CHANGE   TOGRP1.TESTXXX TOGRP1.TESTXXX RP(400)
```

```
 OSREQ QUERY    TOGRP1.TESTXXX TOGRP1.TESTXXX
 OSREQ RETRIEVE TOGRP1.TESTXXX TOGRP1.TESTXXX COMPARE VIEW(PRIMARY)
 OSREQ DELETE   TOGRP1.TESTXXX TOGRP1.TESTXXX
/*
*************************** Bottom of Data ************************
```

The result of this allocation is shown in Example 3-4. Note that we have stored 2000 MB on DASD (❶) “D” means that the object resides on a DASD volume (❷).

Example 3-4 Result of JCL using DASD

```
******************************** TOP OF DATA ********************************
READY
 OSREQ STORE    TOGRP1.TESTXXX TOGRP1.TESTXXX LENGTH(2097152000) ❶
CBRO400I OSREQ STORE successful. Return code = 00000000, reason code = 00000000.
CBRO404I OSREQ STORE response time is 17012 milliseconds.
CBRO405I OSREQ STORE data rate is 120384 kilobytes/second.
READY
 OSREQ QUERY    TOGRP1.TESTXXX TOGRP1.TESTXXX
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
CBRO404I OSREQ QUERY response time is 5 milliseconds.
CBRO419I ************************************************************************
CBRO410I Collection name         = TOGRP1.TESTXXX
CBRO411I Object name             = TOGRP1.TESTXXX
CBRO412I Object size             = 2097152000
CBRO413I Creation date           = 2009-08-31
CBRO414I Creation timestamp      = 22.07.50.474029
CBRO415I Last referenced date    = 0001-01-01
CBRO416I Expiration date         = 0001-01-01
CBRO417I Management class        = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                = D ❷
CBRO444I Pending action date     = 2009-08-31
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = N
READY
 LISTCAT ENTRIES('TOGRP1.TESTXXX') ALL
NONVSAM ------- TOGRP1.TESTXXX
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.243
       RELEASE----------------2     EXPIRATION------9999.999
       ACCOUNT-INFO-----------------------------------(NULL)
     OAMDATA
       DIRECTORYTOKEN----TOGRP1
     SMSDATA
       STORAGECLASS ----OBJDASD     MANAGEMENTCLASS-TOGRP1D1
       DATACLASS --------(NULL)     LBACKUP ---XXXX.XXX.XXXX
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
READY
```

```
 OSREQ CHANGE   TOGRP1.TESTXXX TOGRP1.TESTXXX RP(400)
CBRO400I OSREQ CHANGE successful. Return code = 00000000, reason code = 00000000
CBRO404I OSREQ CHANGE response time is 4 milliseconds.
READY
 OSREQ QUERY    TOGRP1.TESTXXX TOGRP1.TESTXXX
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
CBRO404I OSREQ QUERY response time is 4 milliseconds.
CBRO419I ************************************************************************
CBRO410I Collection name         = TOGRP1.TESTXXX
CBRO411I Object name             = TOGRP1.TESTXXX
CBRO412I Object size             = 2097152000
CBRO413I Creation date           = 2009-08-31
CBRO414I Creation timestamp      = 22.07.50.474029
CBRO415I Last referenced date    = 0001-01-01
CBRO416I Expiration date         = 2010-10-05
CBRO417I Management class        = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                = D
CBRO444I Pending action date     = 2009-08-31
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = N
READY
 OSREQ RETRIEVE TOGRP1.TESTXXX TOGRP1.TESTXXX COMPARE VIEW(PRIMARY)
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
CBRO404I OSREQ QUERY response time is 3 milliseconds.
CBRO400I OSREQ RETRIEVE successful. Return code = 00000000, reason code = 000000
CBRO404I OSREQ RETRIEVE response time is 15603 milliseconds.
CBRO405I OSREQ RETRIEVE data rate is 131255 kilobytes/second.
CBRO420I Data comparison for object TOGRP1.TESTXXX TOGRP1.TESTXXX successful.
READY
 OSREQ DELETE   TOGRP1.TESTXXX TOGRP1.TESTXXX
CBRO400I OSREQ DELETE successful. Return code = 00000000, reason code = 00000000
CBRO404I OSREQ DELETE response time is 99 milliseconds.
READY
END
```

3.1.11 Migration considerations

There are no migration considerations.

3.1.12 Coexistence considerations

Coexistence APAR OA26334 must be installed on any pre-V1.11 level systems prior to starting OAM for the first time on V1.11. This PTF must be installed even if this new function introduced in V1.11 is not exploited.

Table 3-4 outlines the expected behavior on pre-V1.11 level systems for functions initiated against objects that are greater than 256 MB in size that reside on, or are targeted for, tape media.

Table 3-4 Coexistence OAM functions behavior in V1.10 and V1.9

Function	V1.10 behavior	V1.9 behavior
OSREQ STORE	Fail	Fail
OSREQ RETRIEVE	Allowed	Fail
OSREQ DELETE	Allowed	Fail
OSREQ QUERY	Allowed	Allowed
OSREQ CHANGE	Allowed	Allowed
OSMC transition	Objects skipped and CBR9226I message occurs on V1.10.	Objects skipped and CBR9225I message occurs on V1.9.
MOVEVOL	Objects skipped and CBR9226I message occurs on V1.10.	Objects skipped and CBR9225I message occurs on V1.9.
Volume recovery	Objects skipped and CBR9226I message occurs on V1.10.	Objects skipped and CBR9225I message occurs on V1.9.
Object recovery	Fail	Fail
OSMC Expiration	Allowed	Objects skipped and message CBR9225I occurs.
OSMC DASD space manager	Allowed	Objects skipped and message CBR9225I occurs.

3.2 OAM archive retention enhancements

The OAM archive retention enhancements add the following four retention related enhancements to OAM:

- Deletion-hold: Prevents object deletion while an object is in deletion-hold mode.
- Retention-protection: Prevents object deletion prior to an object's expiration date. Does not allow an expiration date to be changed to an earlier date.
- Deletion-protection: Prevents object deletion prior to an object's expiration date.
- Event-based-retention: Sets the object expiration date so that it is dependent on external event notification.

3.2.1 Overview

In pre-z/OS V1.11, OAM protects against inadvertent object deletion by making use of the following tools:

- A CBRUXSAE authorization user exit could optionally be used to verify that the caller has the authority to perform an OSREQ DELETE request.
- A CBRHADUX auto-deletion user exit could optionally be used to define a set of objects that are not to be deleted by OSMC expiration processing, even though the objects' expiration criteria had been met.

More control is required to ensure object data is not deleted or changed prior to the expiration criteria. In addition, a dynamic mechanism is needed to set expiration criteria.

The new values for Status flags are shown in Figure 3-5. The default status flag is zero.

```
QELQESF_________DS___XL2______ODSTATF_-_Status_flags_
QELSF_EBR_______EQU_X'0004'___Event-based_object_
QELSF_DELHOLD__EQU_X'0002'___Deletion-held_object_
QELSF_RETPROT__EQU_X'0001'___Retention-protected_object_
```

Figure 3-5 New values to Status flag

A given object could have any combination of the four new archive retention attributes. However, there are rules that dictate how the object can be deleted:

- An object in deletion-hold mode cannot be deleted (via OSMC or OSREQ) regardless of whether or not retention-protection, deletion-protection, or event-based-retention attributes are enabled.
- If an object has both retention-protection and deletion-protection attributes enabled, then it is treated as a retention-protected object.
- If an object is in event-based-retention mode and has either retention-protection or deletion-protection attributes enabled, then that object cannot be deleted.
- An object that does not have retention-protection, deletion-protection, or deletion-hold attributes enabled can be deleted at any time, even if that object is in event-based-retention mode.

3.2.2 Deletion logic

Figure 3-6 on page 39 outlines how the archive enhancement attributes are applied when an OSREQ DELETE request is received for a given object.

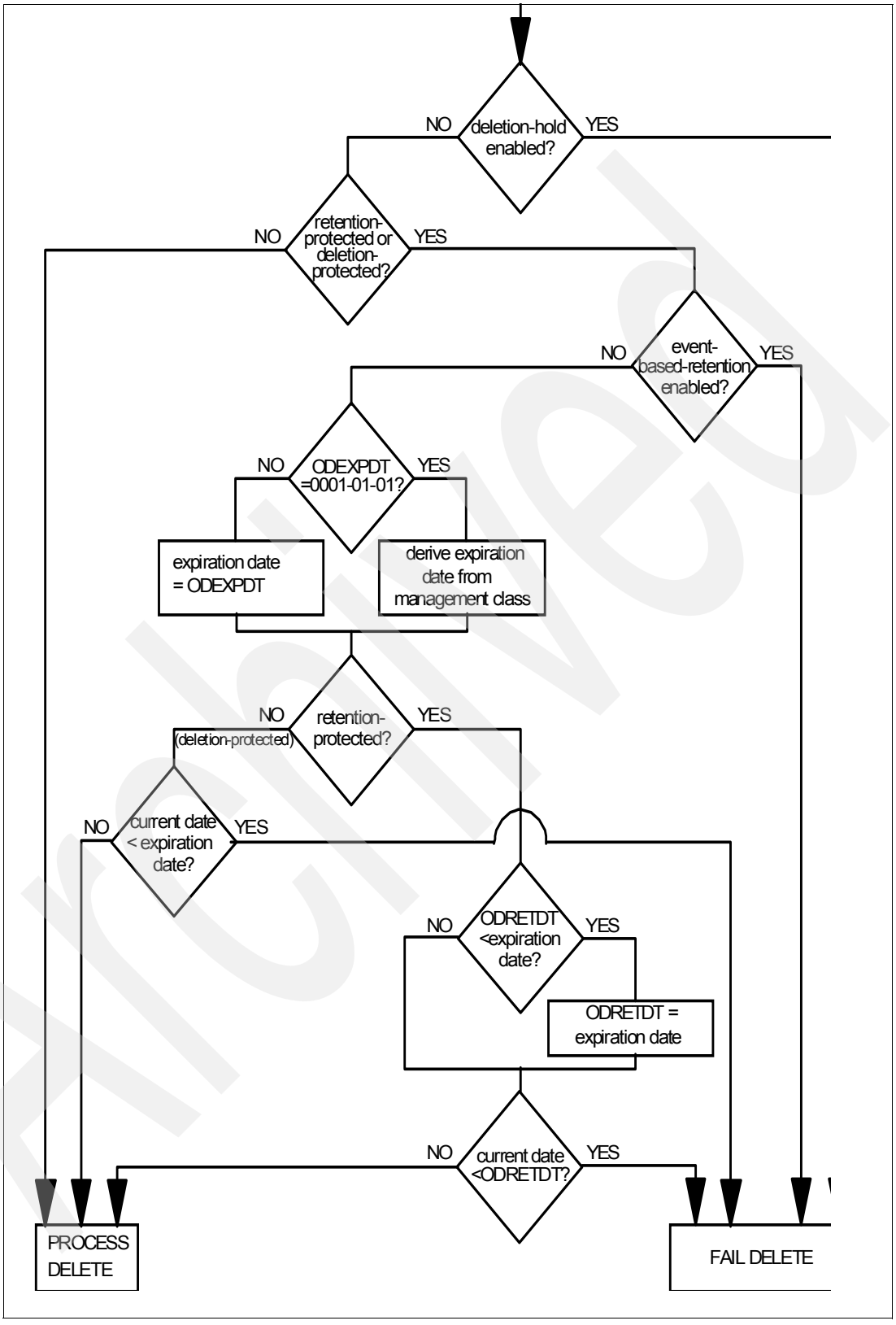


Figure 3-6 Deletion logic

3.2.3 Deletion-hold

Deletion-hold provides a mechanism for an application to place a hold on objects stored in OAM so that they cannot be deleted until the hold is released.

If an object is held, it will not be deleted whether or not retention-protection or deletion-protection is active. If an object is not held, it is handled according to existing processing, such as normal expiration, retention-protection, deletion-protection, or event-based-retention.

An object is placed into deletion-hold status by specifying the DELHOLD=HOLD keyword with the OSREQ API. It can be released from deletion-hold status by specifying the DELHOLD=NOHOLD keyword with the OSREQ API.

The DELHOLD keyword is only valid in the OSREQ CHANGE, STORE, and STOREBEG functions.

An object's deletion-hold status is indicated by the ODSTATF_DELHOLD flag in the ODSTATF column in the row in the object directory table that corresponds with this object.

If an object with event-based-retention enabled is on hold, an event can still be signaled. The hold will not extend the expiration date for an object. If the expiration date specified for the object elapses while the object is on hold, the object will be eligible for deletion whenever the hold is released.

If OSMC expiration processing encounters an object that has a hold on it, and if the pending action date for that object is based off the expiration date, then the pending action date will be changed to “12-31-9999” to prevent it being selected on subsequent Storage Group cycles. The pending action date will be reset to current date when the hold is released via a DELHOLD=NOHOLD keyword on the OSREQ CHANGE request.

Note: An OSREQ CHANGE request normally results in the ACS routines being processed and the object's pending action date and last referenced date set to the current date; however, if the OSREQ CHANGE request only sets the DELHOLD value (it does not attempt to alter Storage Class, Management Class or retention period), then the ACS routines are not run and the pending action date and last referenced date are not altered.

When an object is in deletion-hold mode, it cannot be deleted from the OAM inventory (either via OSREQ DELETE or via OSMC expiration processing). An object can be put into deletion-hold mode via a new DELHOLD=HOLD parameter on the OSREQ API. It can be released from deletion-hold mode via a subsequent OSREQ API request with the DELHOLD=NOHOLD parameter specified, as shown in Example 3-5.

This job will STORE (❶) an object, QUERY it to see if it is okay (❷), change to DELHOLD(HOLD) to test the HOLD keyword *(*❸), QUERY it again to see the result (❹) (the Status Flag will change to 2 - DELHOLD is held (❺)), COMPARE and try to DELETE without success (❻*)* because, DELHOLD(HOLD) worked as expected.

Example 3-5 Using DELHOLD

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
```

```
1READY
 OSREQ STORE  TOGRP1.OAMTEST TOGRP1.OAMTEST LENGTH(209715)     ❶
OSREQ STORE successful. Return code = 00000000, reason code = 00000000.
OSREQ STORE response time is 7 milliseconds.
OSREQ STORE data rate is 29256 kilobytes/second.
READY
 OSREQ QUERY    TOGRP1.OAMTEST TOGRP1.OAMTEST ❷
OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
OSREQ QUERY response time is 4 milliseconds.
****************************************************************
Collection name         = TOGRP1.OAMTEST
Object name             = TOGRP1.OAMTEST
Object size             = 209715
Creation date           = 2009-09-13
Creation timestamp      = 18.14.08.737054
Last referenced date    = 0001-01-01
Expiration date         = 0001-01-01
Management class        = TOGRP1D1
Storage class           = OBJDASD
Primary retrieve key    = '00000000000000000000'X
Backup retrieve key     = '00000000000000000000'X
Backup2 retrieve key    = '00000000000000000000'X
Estimated retrieve time = 300
Location                = D
Pending action date     = 2009-09-13
Status flags            = 0
Retention protect date  = 0001-01-01
Deletion-protected      = N
READY
 LISTCAT ENTRIES('TOGRP1.OAMTEST') ALL
NONVSAM ------- TOGRP1.OAMTEST
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.256
       RELEASE----------------2     EXPIRATION------9999.999
       ACCOUNT-INFO-----------------------------------(NULL)
     OAMDATA
       DIRECTORYTOKEN----TOGRP1
     SMSDATA
       STORAGECLASS ----OBJDASD     MANAGEMENTCLASS-TOGRP1D1
       DATACLASS --------(NULL)     LBACKUP ---XXXX.XXX.XXXX
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
READY
 OSREQ CHANGE   TOGRP1.OAMTEST TOGRP1.OAMTEST DELHOLD(HOLD) ❸
OSREQ CHANGE successful. Return code = 00000000, reason code = 00000000.
OSREQ CHANGE response time is 4 milliseconds.
READY
 OSREQ QUERY    TOGRP1.OAMTEST TOGRP1.OAMTEST ❹
OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
OSREQ QUERY response time is 3 milliseconds.
****************************************************************
Collection name         = TOGRP1.OAMTEST
Object name             = TOGRP1.OAMTEST
Object size             = 209715
```

```
 Creation date           = 2009-09-13
1Creation timestamp      = 18.14.08.737054
 Last referenced date    = 0001-01-01
 Expiration date         = 0001-01-01
 Management class        = TOGRP1D1
 Storage class           = OBJDASD
 Primary retrieve key    = '00000000000000000000'X
 Backup retrieve key     = '00000000000000000000'X
 Backup2 retrieve key    = '00000000000000000000'X
 Estimated retrieve time = 300
 Location                = D
 Pending action date     = 2009-09-13
 Status flags            = 2 ❺
 Retention protect date  = 0001-01-01
 Deletion-protected      = N
 READY
  OSREQ RETRIEVE TOGRP1.OAMTEST TOGRP1.OAMTEST COMPARE VIEW(PRIMARY)
 OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
 OSREQ QUERY response time is 3 milliseconds.
 OSREQ RETRIEVE successful. Return code = 00000000, reason code = 00000000.
 OSREQ RETRIEVE response time is 7 milliseconds.
 OSREQ RETRIEVE data rate is 29256 kilobytes/second.
 Data comparison for object TOGRP1.OAMTEST TOGRP1.OAMTEST successful.
 READY
  OSREQ DELETE   TOGRP1.OAMTEST TOGRP1.OAMTEST ❻
 OSREQ DELETE unsuccessful. Return code = 00000008, reason code = 40060100.
 OSREQ DELETE response time is 3 milliseconds.
 READY
 END
```

We changed DELHOLD parameter to NOHOLD. See the result in Example 3-6.

Example 3-6 Final result when changing from HOLD to NOHOLD

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
1READY
  OSREQ CHANGE   TOGRP1.OAMTEST TOGRP1.OAMTEST DELHOLD(NOHOLD)
 OSREQ CHANGE successful. Return code = 00000000, reason code = 00000000.
 OSREQ CHANGE response time is 57 milliseconds.
 READY
  OSREQ QUERY    TOGRP1.OAMTEST TOGRP1.OAMTEST
 OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
 OSREQ QUERY response time is 4 milliseconds.
 ************************************************************************
 Collection name         = TOGRP1.OAMTEST
 Object name             = TOGRP1.OAMTEST
 Object size             = 209715
 Creation date           = 2009-09-13
 Creation timestamp      = 18.14.08.737054
 Last referenced date    = 2009-09-13
```

```
 Expiration date         = 0001-01-01
 Management class        = TOGRP1D1
 Storage class           = OBJDASD
 Primary retrieve key    = '00000000000000000000'X
 Backup retrieve key     = '00000000000000000000'X
 Backup2 retrieve key    = '00000000000000000000'X
 Estimated retrieve time = 300
Location                = D
Pending action date     = 2009-09-13
Status flags            = 0
Retention protect date  = 0001-01-01
Deletion-protected      = N
READY
 OSREQ RETRIEVE TOGRP1.OAMTEST TOGRP1.OAMTEST COMPARE VIEW(PRIMARY)
OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
OSREQ QUERY response time is 3 milliseconds.
OSREQ RETRIEVE successful. Return code = 00000000, reason code = 00000000.
OSREQ RETRIEVE response time is 6 milliseconds.
OSREQ RETRIEVE data rate is 34132 kilobytes/second.
Data comparison for object TOGRP1.OAMTEST TOGRP1.OAMTEST successful.
READY
 OSREQ DELETE   TOGRP1.OAMTEST TOGRP1.OAMTEST (1)
OSREQ DELETE successful. Return code = 00000000, reason code = 00000000.
OSREQ DELETE response time is 15 milliseconds.
READY
END
```

3.2.4 Retention-protection

Retention-protection's objective is to prevent an object from being deleted or modified prior to its expiration date and also ensures that the object's expiration date cannot be changed to an earlier date.

Retention-protection is enabled at the object Storage Group level via an OAM Retention Protection switch defined in the SMS object Storage Group construct. When an object is stored into an object Storage Group that has OAM Retention Protection enabled, that object is marked as retention-protected by setting the ODSTATF_RETPROT flag (see Figure 3-5 on page 38).

If an object Storage Group is retention-protected at the time, an object is stored into that group, and then that object is considered retention-protected for the life of the object. Turning retention-protection on/off for that object Storage Group will not change the Retention-Protection status of objects that were stored previously.

A retention-protected object cannot have its expiration date brought in to an earlier date, but the expiration date can be pushed out to a later date. Therefore, if an object has an expiration date of "forever:", then it will never be expired or deleted from OAM's object inventory.

Users can use ISMF to set the OAM Retention Protection parameter in the SMS Object Storage Group construct definition to enable retention-protection at an object Storage Group level, as shown in Figure 3-7.

```
Panel  Utilities  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                        OBJECT STORAGE GROUP ALTER            Page 2 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : TOGRP1

To ALTER Storage Group, Specify:
  Volume Full Threshold  . . . .       (0-9999)
  Drive Start Threshold  . . . .       (0-9999)
  Volume Full at Write Error . .       (Y or N)

  OAM Deletion Protection  . . . N     (Y=Enable or N=Disable)
  OAM Retention Protection . . . N     (Y=Enable or N=Disable)









Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 3-7 ISMF panel to enable or disable new OAM protection functions

Storing a retention-protection object

When an object is initially stored into an object Storage Group that has the OAM Retention-Protection attribute enabled, that object is stored as a retention-protected object.

Changing a retention-protection object

The OSREQ CHANGE function provides mechanisms to change the following attributes for a given object:

- Retention period.
- Management Class.
- Storage Class.
- Deletion-hold status.
- Event-based-retention status.
- Retention-protection: Provides OAM's most stringent protection to ensure that an object has not been modified or deleted prior to its expiration date. When retention-protection is enabled for a given object, OAM will not allow that object to be deleted prior to its expiration date. Additionally, OAM will not allow the expiration date to be changed to an earlier date. It will, however, allow the expiration date to be changed to a later date. Retention-protection is in effect for an object for the life of the object. Installations cannot turn it off.

See Figure 3-8 and perform the following steps:

a. Change OAM Retention Protection to Y.

b. Activate SMS.

c. As a test, run a job to have a RETENTION PROTECTION of 3 days and try to change it to 1. The result is shown in Example 3-7.

```
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
sss
                         OBJECT STORAGE GROUP ALTER           Page 2 of
2
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : TOGRP1

To ALTER Storage Group, Specify:
  Volume Full Threshold  . . . .      (0-9999)
  Drive Start Threshold  . . . .      (0-9999)
  Volume Full at Write Error . .      (Y or N)

  OAM Deletion Protection  . . . Y    (Y=Enable or N=Disable)
  OAM Retention Protection . . . Y    (Y=Enable or N=Disable)


Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to
Exit.
```

Figure 3-8 OAM Retention Protection changed to Y

- The job in Example 3-7 tests the retention change for a data set that was created with RP(3). See ❶ for the creation and expiration dates. An OSREQ CHANGE command is issued to change the Retention Protection value to RP(1) and a QUERY is issued. as shown by ❷. The Expiration date has not changed. The DELETE command indicated by ❸ fails because it runs prior to the specified expiration date.

Example 3-7 Test the retention change

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB9D9.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB9D9.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
1READY
 OSREQ STORE  TOGRP1.OAMTEST3 TOGRP1.OAMTEST3 LENGTH(209715) RP(3)
OSREQ STORE successful. Return code = 00000004, reason code = 04020480.
OSREQ STORE response time is 50 milliseconds.
OSREQ STORE data rate is 4095 kilobytes/second.
READY
 OSREQ QUERY    TOGRP1.OAMTEST3 TOGRP1.OAMTEST3
OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
OSREQ QUERY response time is 4 milliseconds.
********************************************************************
```

```
Collection name         = TOGRP1.OAMTEST3
Object name             = TOGRP1.OAMTEST3
Object size             = 209715
Creation date           = 2009-09-13 ❶
Creation timestamp      = 19.46.00.027802
Last referenced date    = 0001-01-01
Expiration date         = 2009-09-16 ❶
Management class        = TOGRP1D1
Storage class           = OBJDASD
Primary retrieve key    = '00000000000000000000'X
Backup retrieve key     = '00000000000000000000'X
Backup2 retrieve key    = '00000000000000000000'X
Estimated retrieve time = 300
Location                = D
Pending action date     = 2009-09-13
Status flags            = 1
Retention protect date  = 2009-09-16
Deletion-protected      = N
READY
 LISTCAT ENTRIES('TOGRP1.OAMTEST3') ALL
NONVSAM ------- TOGRP1.OAMTEST3
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.256
       RELEASE----------------2     EXPIRATION------9999.999
       ACCOUNT-INFO-----------------------------------(NULL)
     OAMDATA
       DIRECTORYTOKEN----TOGRP1
     SMSDATA
       STORAGECLASS ----OBJDASD     MANAGEMENTCLASS-TOGRP1D1
       DATACLASS --------(NULL)     LBACKUP ---XXXX.XXX.XXXX
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
READY
 OSREQ CHANGE   TOGRP1.OAMTEST3 TOGRP1.OAMTEST3 RP(1) ❷
OSREQ CHANGE successful. Return code = 00000004, reason code = 04050448.
OSREQ CHANGE response time is 4 milliseconds.
READY
 OSREQ QUERY    TOGRP1.OAMTEST3 TOGRP1.OAMTEST3 ❷
OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
OSREQ QUERY response time is 3 milliseconds.
******************************************************************
Collection name         = TOGRP1.OAMTEST3
Object name             = TOGRP1.OAMTEST3
Object size             = 209715
Creation date           = 2009-09-13 ❷
1Creation timestamp     = 19.46.00.027802
Last referenced date    = 0001-01-01
Expiration date         = 2009-09-16 ❷
Management class        = TOGRP1D1
Storage class           = OBJDASD
Primary retrieve key    = '00000000000000000000'X
Backup retrieve key     = '00000000000000000000'X
Backup2 retrieve key    = '00000000000000000000'X
Estimated retrieve time = 300
```

```
Location                = D
Pending action date     = 2009-09-13
Status flags            = 1
Retention protect date  = 2009-09-16 ❷
Deletion-protected      = N
READY
 OSREQ RETRIEVE TOGRP1.OAMTEST3 TOGRP1.OAMTEST3 COMPARE VIEW(PRIMARY)
OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
OSREQ QUERY response time is 3 milliseconds.
OSREQ RETRIEVE successful. Return code = 00000000, reason code = 00000000.
OSREQ RETRIEVE response time is 4 milliseconds.
OSREQ RETRIEVE data rate is 51199 kilobytes/second.
Data comparison for object TOGRP1.OAMTEST3 TOGRP1.OAMTEST3 successful.
READY
 OSREQ DELETE   TOGRP1.OAMTEST3 TOGRP1.OAMTEST3 ❸
OSREQ DELETE unsuccessful. Return code = 00000008, reason code = 40060303.
OSREQ DELETE response time is 3 milliseconds.
READY
END
```

- You may run a test to extend the date. The result is shown in Example 3-8.

Example 3-8 Change RP to 5 to set the new data to 2009-09-18

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
1READY
 OSREQ CHANGE   TOGRP1.OAMTEST3 TOGRP1.OAMTEST3 RP(5)
OSREQ CHANGE successful. Return code = 00000000, reason code = 00000000.
OSREQ CHANGE response time is 4 milliseconds.
READY
 OSREQ QUERY    TOGRP1.OAMTEST3 TOGRP1.OAMTEST3
OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
OSREQ QUERY response time is 4 milliseconds.
**************************************************************************
Collection name         = TOGRP1.OAMTEST3
Object name             = TOGRP1.OAMTEST3
Object size             = 209715
Creation date           = 2009-09-13
Creation timestamp      = 19.46.00.027802
Last referenced date    = 2009-09-13
Expiration date         = 2009-09-18
Management class        = TOGRP1D1
Storage class           = OBJDASD
Primary retrieve key    = '00000000000000000000'X
Backup retrieve key     = '00000000000000000000'X
Backup2 retrieve key    = '00000000000000000000'X
Estimated retrieve time = 300
Location                = D
Pending action date     = 2009-09-13
Status flags            = 1
Retention protect date  = 2009-09-18 ❶
```

```
Deletion-protected      = N
READY
END
```

3.2.5 Deletion-protection

Deletion-protection provides a mechanism to prevent an object from being deleted prior to its expiration date. However, it does not protect an object's expiration date from being changed either by a OSREQ CHANGE function or by a change in the SMS Management Class associated with this object.

The OAM Deletion Protection attribute in the SMS object storage group definition works in combination with the DP=P keyword in IEFSSNxx PARMLIB member to determine the current deletion-protection mode (enabled or disabled) for all objects in this object storage group. The OAM Deletion Protection attribute is ignored when DP=A or DP=N. When deletion-protection is enabled, objects in this object storage group cannot be deleted prior to their expiration date. Deletion-protection does not restrict any changes to an object's expiration date.

In order to enable deletion-protection for a given object storage group, use ISMF to set the OAM Deletion Protection attribute to Enabled in the SMS object storage group definition, and then activate the SCDS.

Deletion-protection is similar to retention-protection, in that the objective is to prevent deliberate or accidental deletion of data until its specified retention criterion is met. However, there are a number of key differences, as shown in Table 3-5.

Table 3-5 Differences between deletion-protection and retention-protection

	Deletion-protection	**Retention-protection**
Enablement	The DP=x switch in IEFSSNxx PARMLIB member is used in conjunction with the OAM Deletion Protection switch in the SMS object storage group construct to enable deletion-protection at a global or object storage group level.	Turned on at the object storage group level via the OAM Retention Protection switch in SMS object storage group construct.
Protection status of object	Not associated with specific object.	Never changes. If an object was initially stored in an OAM Retention Protection enabled storage group, then the object is retention-protected for the life of the object.
Expiration date protection	None.	Expiration dates can be pushed out to a later date, but cannot be changed to an earlier date. This setting is set either via OSREQ CHANGE or via transition to a new management class. The latest expiration date is in ODRETDT.

When deletion-protection is enabled (at a global or Storage Group level), OAM will not allow an object to be deleted prior to its expiration date. Deletion-protection differs from retention-protection in that deletion-protection can be turned on and off by the installation, and deletion-protection has no restrictions on the expiration date changing.

Note: If you refer to *z/OS V1R11 DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426, you will discover that the definition of DP=N indicates that deletion-protection is *not* enabled for any object Storage Groups. This is the default.

3.2.6 Implementing through parmlib changes

A new DP=x parameter is added to the OAM1 entry in the IEFSSNxx member of SYS1.PARMLIB to indicate when deletion-protection is enabled or disabled, as shown in Figure 3-9. When deletion-protection is enabled, OAM will not allow an object to be deleted prior to its expiration date.

```
SUBSYS SUBNAME(OAM1) INITRTN(CBRINIT)
INITPARM('[TIME=GMT][,MSG=x][,OTIS=x][,UPD=x][,MOS=nnn][,LOB=x][,QB=x][,DP=x]')
```

Figure 3-9 New parameter DP in IEFSSNxx

When deletion-protection is enabled, OAM will not allow an object to be deleted prior to its expiration date. The DP=x keyword can be used to set the deletion-protection scope as follows:

- DP=A indicates that deletion-protection is enabled for all object Storage Groups.
- DP=P indicates that deletion-protection is partially enabled. Specifically, deletion-protection is enabled only for object Storage Groups that have OAM Deletion Protection set to ENABLED in ISMF for the object Storage Group SMS construct. Remember that retention-protection takes precedence over deletion-protection, meaning that an object cannot be deleted prior its expiration date.
- DP=N indicates that deletion-protection is not enabled for any object Storage Groups. This is the default.

3.2.7 Verifying changes

To verify your changes, perform the following steps:

1. Issue the D SMS,OAM command. You will see the result shown in Figure 3-10.

```
.
.
.
DB2 SSID:  D9DG
XCF GROUP NAME: OAMTEST
XCF MEMBER NAME: OAM70
CBROAM: 70
OAM1 Parms: TIME=LOC  MSG=EM  UPD=Y  QB=Y
            MOS=2000  OTIS=N  LOB=A  DP=N
****************************** BOTTOM OF DATA
```

Figure 3-10 Display OAM and checking DP=N was implemented

2. To test our changes, we change our IEFSSNxx Parmlib member, as shown in Figure 3-11.

```
VIEW       SYS1.PARMLIB(IEFSSNR7) - 01.04
Command ===>
000219 SUBSYS SUBNAME(OAM1)
000220   INITRTN(CBRINIT)
000221   INITPARM('MSG=EM,MOS=2000,LOB=A,DP=P,QB=Y')
```

Figure 3-11 Changed to DP=P to test deletion-protection

a. Perform an IPL and then issue the D SMS,OAM command and you will see the panel shown in Figure 3-12.

```
XCF GROUP NAME: OAMTEST
XCF MEMBER NAME: OAM70
CBROAM: 70
OAM1 Parms: TIME=LOC  MSG=EM  UPD=Y  QB=Y
            MOS=2000  OTIS=N  LOB=A  DP=P
```

Figure 3-12 Changes to DP=P after loading the initial program

b. Change our SMS definition to follow just DP=P by using Example 3-9 as a reference. Be sure to create the Storage Group name first.

Example 3-9 Storage Group Application Selection: SMS Deletion with N

```
STORAGE GROUP APPLICATION SELECTION
 Command ===>

 To perform Storage Group Operations, Specify:
   CDS Name  . . . . . . SYS1.SMS.SCDS
(1 to 44 character data set name or 'Active' )
   Storage Group Name    TOGRP*           (For Storage Group List, fully or
                                          partially specified or * for all)
   Storage Group Type                     (VIO, POOL, DUMMY, COPY POOL BACKUP,
```

```
                                          OBJECT, OBJECT BACKUP, or TAPE)

 Select one of the following options  :
   1  1. List    - Generate a list of Storage Groups
      2. Define  - Define a Storage Group
      3. Alter   - Alter a Storage Group
      4. Volume  - Display, Define, Alter or Delete Volume Information

 If List Option is chosen,
    Enter "/" to select option      Respecify View Criteria
                                    Respecify Sort Criteria
 Use ENTER to Perform Selection;
```

c. In the STORAGE GROUP APPLICATION SELECTION panel, specify the Storage Group Name and specify 1 to generate a list of Storage Groups. Press Enter to go to the STORAGE GROUP LIST panel shown in Example 3-10.

Example 3-10 STORAGE GROUP LIST panel

```
STORAGE GROUP LIST
Command ===>


CDS Name : SYS1.SMS.SCDS

Enter Line Operators below:

     LINE       STORGRP  SG               VIO     VIO   AUTO      MIGRATE SYSTEM
     OPERATOR   NAME     TYPE             MAXSIZE  UNIT  MIGRATE   OR SYS GROUP
    ---(1)----  --(2)--- -------(3)------  --(4)--  (5)-  --(6)---  -----(7)------
    *ALTER      TOGRP1   OBJECT           -------  ----  --------  --------
                TOGRP1B1 OBJECT BACKUP    -------  ----  --------  --------
                TOGRP1B2 OBJECT BACKUP    -------  ----  --------  --------
    ----------  -------- --------  ----------  ----------  BOTTOM  OF  DATA -------
```

d. Specifying a LINE OPERATOR of ALTER and pressing Enter takes you to the OBJECT STORAGE GROUP ALTER panel shown in Example 3-11.

Example 3-11 OBJECT STORAGE GROUP ALTER panel

```
OBJECT STORAGE GROUP ALTER           Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : TOGRP1

To ALTER Storage Group, Specify:
  Description ==> OBJECT STORAGE GROUP FOR TOGRP1
              ==>

  Qualifier  . . . . . TOGRP1    (1 to 8 character qualifier)
  Cycle Start Time . . NONE  (0-23 or NONE)  End Time . .     (0-23 or blank
  OSMC Processing System . . . . SC70      (? for list of OSMC System names)

  Library Names   (1 to 8 Characters each):
   ===>           ===>           ===>           ===>
   ===>           ===>           ===>           ===>

   ALTER     SMS Storage Group Status . . N  (DEFINE - Y, ALTER - Y or N)

Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
```

e. Pressing Enter in the OBJECT STORAGE GROUP ALTER panel after specifying N in the SMS Storage Group Status takes you to the OBJECT STORAGE GROUP ALTER panel shown in Example 3-12.

Example 3-12 OBJECT STORAGE GROUP ALTER panel

```
Panel 4:
OBJECT STORAGE GROUP ALTER             Page 2 of 2
 Command ===>

 SCDS Name . . . . . : SYS1.SMS.SCDS
 Storage Group Name  : TOGRP1

 To ALTER Storage Group, Specify:
   Volume Full Threshold  . . . .       (0-9999)
   Drive Start Threshold  . . . .       (0-9999)
   Volume Full at Write Error . .       (Y or N)

   OAM Deletion Protection  . . . Y     (Y=Enable or N=Disable)
   OAM Retention Protection . . . N     (Y=Enable or N=Disable)
Use ENTER to Perform Verification; Use UP Command to View previous
Panel;
```

f. Validate and activate your SMS and then test.

3. We submitted two jobs, one with no RP defined (see ❶ in Example 3-13) and the next one with RP(1). We tried to delete both objects on the same day, which is not allowed, as shown in Example 3-13.

Example 3-13 Job without an RP defined

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
IEF375I JOB/MHLRES2F/START 2009260.2346
IEF376I JOB/MHLRES2F/STOP  2009260.2346 CPU    0MIN 00.05SEC SRB    0MIN
00.00SEC
READY
  OSREQ STORE  TOGRP1.TESTXP1 TOGRP1.TESTXP0 LENGTH(2097) ❶
CBR0400I OSREQ STORE successful. Return code = 00000000, reason code =
00000000.
CBR0404I OSREQ STORE response time is 28 milliseconds.
CBR0405I OSREQ STORE data rate is 72 kilobytes/second.
READY
  OSREQ QUERY  TOGRP1.TESTXP1 TOGRP1.TESTXP0
CBR0400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBR0404I OSREQ QUERY response time is 4 milliseconds.
CBR0419I ************************************************************************
CBR0410I Collection name         = TOGRP1.TESTXP1
CBR0411I Object name             = TOGRP1.TESTXP0
CBR0412I Object size             = 2097
CBR0413I Creation date           = 2009-09-17
CBR0414I Creation timestamp      = 23.46.12.744264
CBR0415I Last referenced date    = 0001-01-01
CBR0416I Expiration date         = 0001-01-01 ❹
```

```
CBRO417I Management class        = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                = D
CBRO444I Pending action date     = 2009-09-17
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = Y
READY
READY
  OSREQ DELETE TOGRP1.TESTXP1 TOGRP1.TESTXP0
CBRO401I OSREQ DELETE unsuccessful. Return code = 00000008, reason code =
40060202.
CBRO404I OSREQ DELETE response time is 4 milliseconds.
```

4. We change the RP parameter (retention period) and get the result shown in (❶) Example 3-14. Example 3-13 on page 52, at ❹, shows that the RP had no regular expiration date.

Example 3-14 Add an RP to this object

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
READY
  OSREQ CHANGE TOGRP1.TESTXP1 TOGRP1.TESTXP0 RP(1) ❶
CBRO400I OSREQ CHANGE successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ CHANGE response time is 33 milliseconds.
READY
  OSREQ QUERY  TOGRP1.TESTXP1 TOGRP1.TESTXP0
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ QUERY response time is 4 milliseconds.
CBRO419I ************************************************************************
CBRO410I Collection name         = TOGRP1.TESTXP1
CBRO411I Object name             = TOGRP1.TESTXP0
CBRO412I Object size             = 2097
CBRO413I Creation date           = 2009-09-17
CBRO414I Creation timestamp      = 23.46.12.744264
CBRO415I Last referenced date    = 0001-01-01
CBRO416I Expiration date         = 2009-09-18 ❶
CBRO417I Management class        = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                = D
CBRO444I Pending action date     = 2009-09-18
```

```
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = Y
READY
END
READY
  OSREQ DELETE TOGRP1.TESTXP1 TOGRP1.TESTXP0
CBRO400I OSREQ DELETE successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ DELETE response time is 6 milliseconds.
READY
END
```

5. The result of the job with a retention period of 1 (RP(1)) is shown in Example 3-15. When creating your objects, pay attention to the creation date and expiration date, and see if they are what you have planned.

Example 3-15 Job with RP(1)

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
IEF375I  JOB/MHLRES2E/START 2009260.1938
IEF376I  JOB/MHLRES2E/STOP  2009260.1938 CPU    0MIN 00.06SEC SRB    0MIN
00.00SEC
READY
  OSREQ STORE  TOGRP1.TESTXP1 TOGRP1.TESTXP1 LENGTH(2097) RP(1)
CBRO400I OSREQ STORE successful. Return code = 00000004, reason code =
04020480.
CBRO404I OSREQ STORE response time is 27 milliseconds.
CBRO405I OSREQ STORE data rate is 75 kilobytes/second.
READY
  OSREQ QUERY  TOGRP1.TESTXP1 TOGRP1.TESTXP1
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ QUERY response time is 4 milliseconds.
CBRO419I********************************************************************
CBRO410I Collection name         = TOGRP1.TESTXP1
CBRO411I Object name             = TOGRP1.TESTXP1
CBRO412I Object size             = 2097
CBRO413I Creation date           = 2009-09-17
CBRO414I Creation timestamp       = 19.38.44.411613
CBRO415I Last referenced date     = 0001-01-01
CBRO416I Expiration date          = 2009-09-18
CBRO417I Management class         = TOGRP1D1
CBRO418I Storage class            = OBJDASD
CBRO427I Primary retrieve key     = '00000000000000000000'X
CBRO428I Backup retrieve key      = '00000000000000000000'X
CBRO429I Backup2 retrieve key     = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                 = D
CBRO444I Pending action date      = 2009-09-17
CBRO445I Status flags             = 0
```

```
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = N
READY
  LISTCAT ENTRIES('TOGRP1.TESTXP1') ALL
NONVSAM ------- TOGRP1.TESTXP1
     IN-CAT --- UCAT.VSBOXO1
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.260
       RELEASE----------------2     EXPIRATION------9999.999
       ACCOUNT-INFO-----------------------------------(NULL)
     OAMDATA
       DIRECTORYTOKEN----TOGRP1
     SMSDATA
       STORAGECLASS ----OBJDASD     MANAGEMENTCLASS-TOGRP1D1
       DATACLASS --------(NULL)     LBACKUP ---XXXX.XXX.XXXX
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
READY
END
```

6. We ran a job several minutes later, when it was already the next day, so we have a new date, as shown in Example 3-16.

Example 3-16 The second job to delete the object

```
Note that after 23:59, a second job to delete was excuted, deleting the
dataset:
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
IEF375I  JOB/MHLRES2F/START 2009261.0003
IEF376I  JOB/MHLRES2F/STOP  2009261.0003 CPU    0MIN 00.03SEC SRB    0MIN
00.00SEC
READY
  OSREQ DELETE TOGRP1.TESTXPT TOGRP1.TESTXP1
CBRO400I OSREQ DELETE successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ DELETE response time is 7 milliseconds.
READY
END
```

7. Now we perform a test where we use the DP=A setting, which means that we have set global protection (see 3.2.6, “Implementing through parmlib changes” on page 49 for more details):

   a. Change the IEFSSNxx Parmlib member, as shown in Figure 3-13.

```
VIEW       SYS1.PARMLIB(IEFSSNR7) - 01.04
Command ===>
000219 SUBSYS SUBNAME(OAM1)
000220   INITRTN(CBRINIT)
000221   INITPARM('MSG=EM,MOS=2000,LOB=A,DP=A,QB=Y')
```

Figure 3-13 Changed to DP=A to test deletion-protection

   b. Perform an IPL and issue the D SMS,OAM command and you will see the result shown in Figure 3-14.

```
XCF GROUP NAME: OAMTEST
XCF MEMBER NAME: OAM70
CBROAM: 70
OAM1 Parms: TIME=LOC  MSG=EM  UPD=Y  QB=Y
            MOS=2000  OTIS=N  LOB=A  DP=A
```

Figure 3-14 The change to DP=A was reflected after IPL

   c. Change the SMS definition, as shown in Figure 3-15.

```
 OBJECT STORAGE GROUP ALTER            Page 2 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : TOGRP1

To ALTER Storage Group, Specify:
  Volume Full Threshold  . . . .       (0-9999)
  Drive Start Threshold  . . . .       (0-9999)
  Volume Full at Write Error . .       (Y or N)

  OAM Deletion Protection  . . . N     (Y=Enable or N=Disable)
  OAM Retention Protection . . . N     (Y=Enable or N=Disable)




Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to
Exit.
```

Figure 3-15 Change OAM Protection to N

8. Validate and activate your SMS and test it.
   a. Set the first job with RP(3) and review the result shown in Example 3-17.

Example 3-17 Job with RP(3) and running under DP=A

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB9D9.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB9D9.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
READY
  OSREQ STORE  TOGRP1.TESTXPT TOGRP1.TESTXP2 LENGTH(2097) RP(3)
CBRO400I OSREQ STORE successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ STORE response time is 240 milliseconds.
CBRO405I OSREQ STORE data rate is 7 kilobytes/second.
READY
  OSREQ QUERY  TOGRP1.TESTXPT TOGRP1.TESTXP2
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ QUERY response time is 61 milliseconds.
CBRO419I
*------------------------------------------------------------------------
CBRO410I Collection name        = TOGRP1.TESTXPT
CBRO411I Object name           = TOGRP1.TESTXP2
CBRO412I Object size            = 2097
CBRO413I Creation date          = 2009-09-18
CBRO414I Creation timestamp     = 23.30.54.539723
CBRO415I Last referenced date    = 0001-01-01
CBRO416I Expiration date         = 2009-09-21 ❶
CBRO417I Management class       = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                = D
CBRO444I Pending action date     = 2009-09-18
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = Y
READY
  LISTCAT ENTRIES('TOGRP1.TESTXP2') ALL
NONVSAM ------- TOGRP1.TESTXP2
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.260
       RELEASE----------------2     EXPIRATION------9999.999
       ACCOUNT-INFO-----------------------------------(NULL)
     OAMDATA
       DIRECTORYTOKEN----TOGRP1
     SMSDATA
       STORAGECLASS ----OBJDASD     MANAGEMENTCLASS-TOGRP1D1
       DATACLASS --------(NULL)     LBACKUP ---XXXX.XXX.XXXX
```

```
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
READY
END
```

b. The expiration date was 2009-09-21. (See (❶) in Example 3-17 on page 57). We did our deletion test on 2009-09-21 and it worked, as shown in Example 3-18.

Example 3-18 Deletion job

```
---- MONDAY,    21 SEP 2009 ----
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
IEF376I  JOB/MHLRES2I/STOP  2009264.1431 CPU    0MIN 00.04SEC SRB    0MIN
00.00SEC
READY
  OSREQ QUERY  TOGRP1.TESTXPT TOGRP1.TESTXP2
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ QUERY response time is 5 milliseconds.
CBRO419I
************************************************************************
CBRO410I Collection name        = TOGRP1.TESTXPT
CBRO411I Object name           = TOGRP1.TESTXP2
CBRO412I Object size            = 2097
CBRO413I Creation date          = 2009-09-18
CBRO414I Creation timestamp     = 23.30.54.539723
CBRO415I Last referenced date    = 0001-01-01
CBRO416I Expiration date        = 2009-09-21
CBRO417I Management class       = TOGRP1D1
CBRO418I Storage class          = OBJDASD
CBRO427I Primary retrieve key   = '00000000000000000000'X
CBRO428I Backup retrieve key    = '00000000000000000000'X
CBRO429I Backup2 retrieve key   = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location               = D
CBRO444I Pending action date     = 2009-09-18
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = Y
READY
  LISTCAT ENTRIES('TOGRP1.TESTXP2') ALL
NONVSAM ------- TOGRP1.TESTXP2
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.260
       RELEASE----------------2     EXPIRATION------9999.999
       ACCOUNT-INFO-----------------------------------(NULL)
     OAMDATA
       DIRECTORYTOKEN----TOGRP1
     SMSDATA
```

```
       STORAGECLASS ----OBJDASD     MANAGEMENTCLASS-TOGRP1D1
       DATACLASS --------(NULL)     LBACKUP ---XXXX.XXX.XXXX
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
READY
END
READY
  OSREQ DELETE TOGRP1.TESTXPT TOGRP1.TESTXP2
CBRO400I OSREQ DELETE successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ DELETE response time is 50 milliseconds.
```

c. Set the second job with RP(4) and review the result shown in Example 3-19.

Example 3-19 Job with RP(4) and running under DP=A

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB9D9.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
READY
  OSREQ STORE  TOGRP1.TESTXPT TOGRP1.TESTXP3 LENGTH(2097) RP(4)
CBRO400I OSREQ STORE successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ STORE response time is 8 milliseconds.
CBRO405I OSREQ STORE data rate is 255 kilobytes/second.
READY
  OSREQ QUERY  TOGRP1.TESTXPT TOGRP1.TESTXP3
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBRO404I OSREQ QUERY response time is 3 milliseconds.
CBRO419I
********************************************************************************
CBRO410I Collection name        = TOGRP1.TESTXPT
CBRO411I Object name           = TOGRP1.TESTXP3
CBRO412I Object size            = 2097
CBRO413I Creation date          = 2009-09-18
CBRO414I Creation timestamp     = 23.32.20.979505
CBRO415I Last referenced date    = 0001-01-01
CBRO416I Expiration date         = 2009-09-22 ❶
CBRO417I Management class       = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                = D
CBRO444I Pending action date     = 2009-09-18
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = Y
READY
```

d. Perform the same deletion job with the object defined as RP(4). Review the date at ❶ in Example 3-19 on page 59 and the results in Example 3-20.

Example 3-20 Result of deletion job

```
---- MONDAY,    21 SEP 2009 ----
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB9D9.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB9D9.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
IEF376I  JOB/MHLRES2J/STOP  2009264.1449 CPU    0MIN 00.04SEC SRB    0MIN
00.00SEC
READY
  OSREQ QUERY  TOGRP1.TESTXPT TOGRP1.TESTXP3
CBR0400I OSREQ QUERY successful. Return code = 00000000, reason code =
00000000.
CBR0404I OSREQ QUERY response time is 5 milliseconds.
CBR0419I
********************************************************************
CBR0410I Collection name        = TOGRP1.TESTXPT
CBR0411I Object name           = TOGRP1.TESTXP3
CBR0412I Object size            = 2097
CBR0413I Creation date          = 2009-09-18
CBR0414I Creation timestamp      = 23.32.20.979505
CBR0415I Last referenced date    = 0001-01-01
CBR0416I Expiration date        = 2009-09-22
CBR0417I Management class       = TOGRP1D1
CBR0418I Storage class          = OBJDASD
CBR0427I Primary retrieve key    = '00000000000000000000'X
CBR0428I Backup retrieve key     = '00000000000000000000'X
CBR0429I Backup2 retrieve key    = '00000000000000000000'X
CBR0430I Estimated retrieve time = 300
CBR0443I Location               = D
CBR0444I Pending action date     = 2009-09-18
CBR0445I Status flags            = 0
CBR0446I Retention protect date  = 0001-01-01
CBR0447I Deletion-protected      = Y
READY
READY
  OSREQ DELETE TOGRP1.TESTXPT TOGRP1.TESTXP3
CBR0401I OSREQ DELETE unsuccessful. Return code = 00000008, reason code =
40060202.
CBR0404I OSREQ DELETE response time is 3 milliseconds.
READY
```

Figure 3-16 explains the `CBRO401I OSREQ DELETE unsuccessful. Return code = 00000008, reason code = 40060202` message. The left column (Return Code) is 8, and after that column, you can see that the number 40060202 should be read as 40 06 02 02, and that each one is below columns 0, 1, 2, 3 respectively.

Return Code	Reason Code (Bytes)				Description
	0	1	2	3	
8 (X'08')	X'40'	X'06'	X'01'	X'00'	OSREQ DELETE: Deletion is not allowed because the object is in DELHOLD=HOLD state.
	X'40'	X'06'	X'02'	X'01'	OSREQ DELETE: Deletion is not allowed because the object is under deletion-protection and is still in event-based-retention state.
	X'40'	X'06'	X'02'	X'02'	OSREQ DELETE: Deletion is not allowed because the object is under deletion-protection and the object's expiration date has not yet been reached.
	X'40'	X'06'	X'03'	X'01'	OSREQ DELETE: Deletion is not allowed because the object is under retention-protection and is still in event-based-retention state.
	X'40'	X'06'	X'03'	X'03'	OSREQ DELETE: Deletion is not allowed because the object is under retention-protection and the object's retention date has not yet been reached.

Figure 3-16 Message explanation

Another test we did was to use the keyword DELHOLD(NOHOLD) to change the protected object TOGRP1.TESTXP3 and performing a deletion in advance in the sequence. The results of DELHOLD(NOHOLD) are shown in Example 3-21.

Example 3-21 Changing the object using DELHOLD(NOHOLD)

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
READY
  OSREQ CHANGE   TOGRP1.TESTXPT TOGRP1.TESTXP3 DELHOLD(NOHOLD)
CBRO400I OSREQ CHANGE successful. Return code = 00000000, reason code = 00000000.
CBRO404I OSREQ CHANGE response time is 143 milliseconds.
READY
  OSREQ QUERY  TOGRP1.TESTXPT TOGRP1.TESTXP3
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
CBRO404I OSREQ QUERY response time is 4 milliseconds.
CBRO419I ************************************************************************
CBRO410I Collection name         = TOGRP1.TESTXPT
CBRO411I Object name             = TOGRP1.TESTXP3
CBRO412I Object size             = 2097
CBRO413I Creation date           = 2009-09-18
CBRO414I Creation timestamp      = 23.32.20.979505
CBRO415I Last referenced date    = 0001-01-01
CBRO416I Expiration date         = 2009-09-22
CBRO417I Management class        = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
```

```
CBRO443I Location                = D
CBRO444I Pending action date     = 2009-09-21
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = Y
READY
```

Run the job to try delete the object in advance and review the result shown in Example 3-22.

Example 3-22 OSREQ DELETE job

```
---- MONDAY,    21 SEP 2009 ----
...
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DB2xx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DB2xx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
READY
  OSREQ DELETE   TOGRP1.TESTXPT TOGRP1.TESTXP3
CBRO401I OSREQ DELETE unsuccessful. Return code = 00000008, reason code = 40060202.
CBRO404I OSREQ DELETE response time is 4 milliseconds.
READY
  OSREQ QUERY  TOGRP1.TESTXPT TOGRP1.TESTXP3
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
CBRO404I OSREQ QUERY response time is 4 milliseconds.
CBRO419I ************************************************************************
CBRO410I Collection name         = TOGRP1.TESTXPT
CBRO411I Object name             = TOGRP1.TESTXP3
CBRO412I Object size             = 2097
CBRO413I Creation date           = 2009-09-18
CBRO414I Creation timestamp      = 23.32.20.979505
CBRO415I Last referenced date    = 0001-01-01
CBRO416I Expiration date         = 2009-09-22
CBRO417I Management class        = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                = D
CBRO444I Pending action date     = 2009-09-21
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = Y
READY
END
```

3.2.8 Event-based-retention

Traditionally, an OAM object can expire automatically based on its age, its usage, or a specific date (derived from its Management Class or a management-class-approved object-specific retention period).

Event-based retention provides an alternate mechanism whereby retention time for an object can be initiated by an external event.

The RETPD keyword can be used optionally with OSREQ STORE, STOREBEG, or CHANGE requests to explicitly set an expiration date for the object, or indicate to OAM to derive the expiration date dynamically based on the SMS Management Class associated with this object.

An object with RETPD = -2 (negative 2) indicates that this object is in event-based-retention status and is waiting until an external event trigger is received before setting the object's expiration date and setting the ODSTATF_EBR flag to ON in the ODSTATF.

An object that is in event-based-retention mode is waiting for notification (via the EVENTEXP keyword in the OSREQ API) to set its expiration date. The user's application would then have to keep track of any event that would initiate the retention time and keep track of what data is associated with the event.

Upon the occurrence of such an event, the user's application would use the EVENTEXP=number_of_days keyword on an OSREQ CHANGE request to notify OAM of the event. At this point, the object's expiration date (ODEXPDT) is set as follows:

- If the EVENTEXP value specified is greater than the object's Management Class retention limit, the expiration date (ODEXPDT) is set to the creation date of the object plus the object's Management Class retention limit. Otherwise, the ODEXPDT is set to the sum of today's date plus the EVENTEXP value.
- For retention-protection objects:
  - The ODRETDT is set to whichever is later, the newly calculated ODEXPDT or the current ODRETDT.
  - The ODEXPDT is set to whichever is later, the ODRETDT or the ODEXPDT.

The object is no longer waiting for an EVENTEXP notification, and has no knowledge that it was previously in an event-based-retention mode. After the expiration criteria has elapsed, OAM deletes the data during OSMC expiration processing.

Note: An object in event-based-retention mode is waiting for its expiration date to be set; therefore, it cannot be deleted if either deletion-protection or retention-protection is enabled.

3.2.9 OSREQ macro application programming interface

The OSREQ macro provides the ACCESS, CHANGE, DELETE, QUERY, RETRIEVE, STORE, STOREBEG, STOREPRT, STOREEND, and UNACCESS functions. The OSREQ macro is changed to provide new retention functions.

OSREQ CHANGE function

The CHANGE function is used to alter the Storage Class, Management Class, or retention period for previously stored objects. It can also be used to put an object into (or out of) deletion-hold status, as well as notify an event-based-retention object that an expiration event has occurred. The specified change is made to the object directory table immediately.

The following new keywords have been added to OSREQ CHANGE function:

- DELHOLD = HOLD | NOHOLD
  - 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 to be released from a deletion-hold status. This object can be deleted by OSREQ DELETE or by OSMC expiration processing.

- EVENTEXP=nnnn indicates that the object's expiration date should be set to current date + nnnn days. If the specified object is not currently waiting for an event, then this change request fails.
- Add -2 (negative 2) as valid value for RETPD to indicate that event-based-retention is enabled for this object.

The new OSREQ CHANGE command syntax is shown in Figure 3-17.

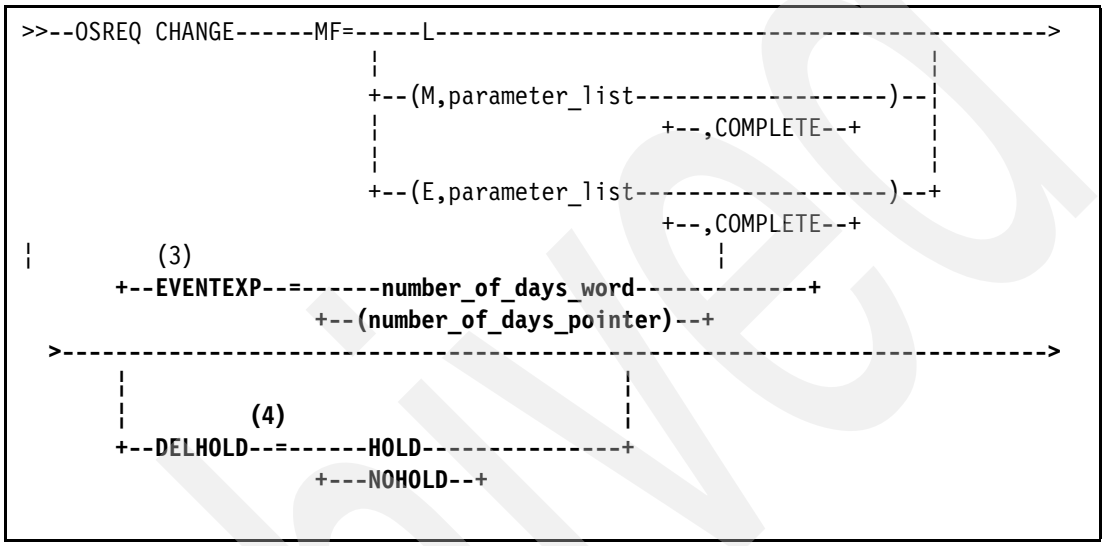


Figure 3-17 OSREQ CHANGE command syntax

Note:

- These keyword parameters must be specified on at least one of the forms if the MF=E entry does not indicate COMPLETE.
- These keyword parameters set the object's pending action date to the current date.
- The EVENTEXP and RETPD keywords cannot be issued in the same statement.
- If the DELHOLD keyword is issued without any type (2) keywords, ACS routines will not run or and pending action date will not be set.

OSREQ STORE and OSREQ STOREBEG functions

The following new keywords and values are added to the OSREQ STORE and STOREBEG functions:

- DELHOLD = HOLD | NOHOLD
  - HOLD indicates that an object will 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 will not be placed in a deletion-hold status. This object can be deleted by OSREQ DELETE or by OSMC expiration processing. This is the default value if DELHOLD is not specified for a STORE or STOREBEG request.
- Add -2 (negative 2) as valid value for RETPD to indicate that event-based-retention is enabled for this object.

OSREQ STORE syntax

The new OSREQ STORE syntax is shown in Figure 3-18.

```
 >>--OSREQ STORE--MF=-----L-------------------------------------------->
                        +--(M,parameter_list------------------)--+
                        |                     +--,COMPLETE--+     |
                        |                                   |     |
                        +--(E,parameter_list------------------)--+
                                              +--,COMPLETE--+
>-------------------------------------------------------------------------->
       |              +---NOHOLD--+          |
       +--DELHOLD--=------HOLD---------------+
```

Figure 3-18 New OSREQ STORE syntax

Note:

- These keywords are required and must be specified on the MF=E form if it indicates COMPLETE or they must be specified on at least one of the forms if the MF=E does not indicate COMPLETE.
- The size specified must be the exact total size of the object.

OSREQ STOREBEG syntax

The new syntax for OSREQ STOREBEG is shown in Figure 3-19.

```
>>--OSREQ STOREBEG--MF=-----L-------------------------------------------->
                        +--(M,parameter_list------------------)--+
                        |                     +--,COMPLETE--+     |
                        |                                   |     |
                        +--(E,parameter_list------------------)--+

>-------------------------------------------------------------------------->
       |              +---NOHOLD--+          |
       +--DELHOLD--=------HOLD---------------+
```

Figure 3-19 New OSREQ STOREBEG syntax

Note:

- These keywords are required and must be specified on the MF=E form if it indicates COMPLETE or they must be specified on at least one of the forms if the MF=E does not indicate COMPLETE.
- The size specified must be the exact total size of the object.

3.2.10 OSREQ TSO/E command processor

The OSREQ command is a TSO/E command processor that can be used to verify object support after installing OAM.

Changes have been made to the OSREQ TSO/E Command Processor to accommodate the new EVENTEXP and DELHOLD keywords, and the new (-2) value for RETPD. See 3.2.1, “Overview” on page 37, bullet “Event-based-retention” for more details about RETPD (-2).

Additionally, changes have been made to the OSREQ QUERY output messages to include the ODPENDDT, ODLOCFL, ODSTATF, and ODRETDT keywords for object pending action date, location flag, status flags and retention date, as well as deletion-protection status.

Note: A TSO/E command interface (OSREQ TSO/E Command Processor) provides a simple mechanism to invoke individual OSREQ API functions. The usually required OSREQ ACCESS and UNACCESS functions are performed automatically within the OSREQ TSO/E command processor around the individual function to be performed.

The OSREQ TSO/E command processor does not provide the full functionality of the OSREQ API, and you cannot store real data using the OSREQ TSO/E command processor STORE function; otherwise, dummy data is created for the object. You can use the OSREQ TSO/E command processor as a simple mechanism to perform some other operations on real object data without having to write an application program.

You can use the OSREQ TSO/E command processor to compare the primary copy of an object with a backup copy of the object, for example, to ensure that both copies match in order to meet audit requirements. The OSREQ TSO/E command processor can be invoked in a native TSO environment at the READY prompt or within a CLIST, from ISPF, or by invoking the TSO/E program IKJEFT01 in the background through a batch job and providing OSREQ TSO/E commands as an input.

For more details, see Appendix B, “Sample Library Members”, in *z/OS V1R11 DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426.

The new OSREQ TSO/E command syntax is shown in Figure 3-20.

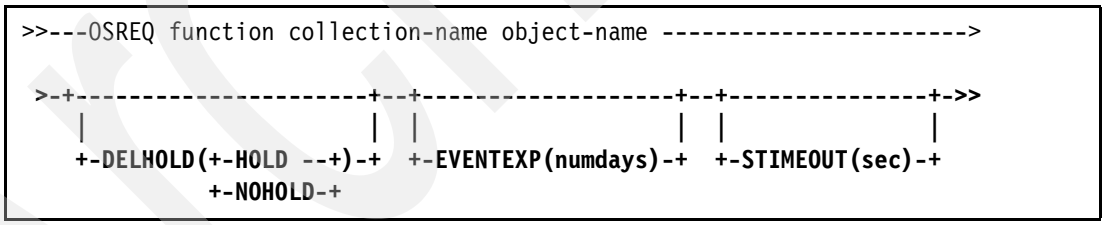


Figure 3-20 New OSREQ TSO/E command syntax

Changes in OSREQ TSO/E functions

The functions shown in this section have been changed in z/OS V1.11.

CHANGE function

An OSREQ CHANGE command results in an OSREQ CHANGE macro invocation to change the Management Class, retention period, Storage Class, deletion-hold status, and event-based-retention status associated with the specified object.

The following optional keywords are valid for an OSREQ CHANGE command:

- MANAGEMENTCLASS
- RETENTIONPERIOD
- STORAGECLASS

- DELHOLD
- EVENTEXP

QUERY function

An OSREQ QUERY command results in an OSREQ QUERY macro invocation for the specified collection name and object name. There are no required or optional keywords for this command.

You might specify the object name for the OSREQ QUERY command as a generic object name by specifying an asterisk (*) for the low-level qualifier of the object name. In the case where the low-level qualifier of the object name is an asterisk, the OSREQ QUERY macro that is issued is a generic query that might result in the directory information of multiple objects being displayed. In the case of a generic query, the query buffer obtained by the OSREQ TSO/E command processor is sufficient to hold the directory information for 10,000 objects. If there are more than 10,000 objects, only the directory information for the first 10,000 objects is listed, and the OSREQ TSO/E command processor ends with return code 4.

The following OAM directory information is listed for each object through the OSREQ TSO/E command processor:

- Collection name
- Object name
- Creation date
- Creation timestamp
- Last reference date
- Expiration date
- Storage Class
- Management Class
- Object length
- Object location
- Pending action date
- Object status flags for retention-protection, event-based-retention, and deletion-hold status
- Retention protect date for retention-protected object
- Deletion-protection status

STORE function

An OSREQ STORE command results in an OSREQ STORE macro invocation to store an object with a specified collection name and object name.

The LENGTH keyword is a required keyword for an OSREQ STORE command. The value specified with the length keyword is the length, in bytes, of the object to store. If the LENGTH keyword or value is omitted, an error message is issued.

The OSREQ command processor creates an object of the specified length and fills the object with a predefined pattern. The predefined pattern is a combination of the collection name, object name, and the 4 byte binary counter. The binary counter in the data pattern is incremented by one for each replication of the data pattern within the object.

The following keywords are valid for an OSREQ STORE command:

- LENGTH (required)
- MANAGEMENTCLASS (optional)
- RETENTIONPERIOD (optional)
- STORAGECLASS (optional)
- RETCODE2 (optional)
- DELHOLD (optional)
- STIMEOUT (optional)

New attributes are assigned to the object, depending upon the attributes indicated for the keywords associated with the OSREQ STORE command. If these attributes are not specified, the defaults for the collection or the ACS overrides are assigned.

3.2.11 DB2 tables

New ODSTATF and ODRETDT columns have been added to the OAM object directory tables. They include the following columns:

- ODSTATF contains status flags for this object with the following values:
  - 0000 - Default.
  - 0001- ODSTATF_RETPROT: Retention-protection is in effect for the life of this object. This flag is set to ON when the object is stored in an object storage group with OAM Retention Protection enabled, and it is set OFF when an object is stored in an object storage group with OAM Retention Protection disabled.
  - 0010 - ODSTATF_DELHOLD: Deletion-hold is active, which prevents this object from being deleted. This flag is valid for both retention-protected and non-retention-protected objects.
  - 0100 - ODSTATF_EBR: Event-based-retention is active. The object is waiting for event notification (via OSREQ CHANGE with EVENTEXP) before calculating its expiration date. This flag is valid for both retention-protected and non-retention-protected objects.
- ODRETDT contains the latest expiration date derived for a retention-protected object. The retention-protected object cannot be deleted prior to the date specified in this column.

Note: This is another representation of the flag status. See Figure 3-5 on page 38 as well.

3.2.12 Changes in ISMF

Two new parameters have been added to the ISMF panels for Object Storage Group:

- OAM Retention Protection: This parameter has two valid values: Y (enable) and N (disable). New objects stored in an object storage group with this parameter enabled are flagged as retention-protected for the entire life of the objects. A retention-protected object cannot be deleted prior to its expiration date, and its expiration date can never move to an earlier date.
- OAM Deletion Protection: This parameter has two valid values: Y (enable) and N (disable). This parameter, in conjunction with the DP=P keyword in IEFSSNxx PARMLIB member, determines the current deletion-protection mode (enabled or disabled) for all objects in this object storage group. This value is ignored when DP=A or DP=N. When

deletion-protection is enabled, objects in this object storage group cannot be deleted prior to their expiration date. Deletion-protection does not restrict any changes to an object's expiration date.

Changing ISMF to add parameters to the object storage group construct to indicate whether or not OAM Deletion Protection or OAM Retention Protection is enabled for this object storage group might involve modifying the following items:

- Object Storage Group Define/Alter
- Storage Group List
- Storage Group List Print
- Storage Group List Sort
- Storage Group List View

The following examples illustrate changes in ISMF panels for Object Storage Group.

Figure 3-21 shows the Storage Group option 6 in the main panel.

```
 STORAGE GROUP APPLICATION SELECTION
Command ===>

To perform Storage Group Operations, Specify:
  CDS Name  . . . . . SYS1.SMS.SCDS
                              (1 to 44 character data set name or 'Active' )
  Storage Group Name    TEST            (For Storage Group List, fully or
                                        partially specified or * for all)
  Storage Group Type    OBJECT          (VIO, POOL, DUMMY, COPY POOL BACKUP,
                                        OBJECT, OBJECT BACKUP, or TAPE)

Select one of the following options  :
  2  1. List    - Generate a list of Storage Groups
     2. Define  - Define a Storage Group
     3. Alter   - Alter a Storage Group
     4. Volume  - Display, Define, Alter or Delete Volume Information

If List Option is chosen,
   Enter "/" to select option      Respecify View Criteria
                                   Respecify Sort Criteria
Use ENTER to Perform Selection;
```

Figure 3-21 Option 6 Storage Group main panel selection

Figure 3-22 shows changes in the Object Storage Group Define ISMF panel.

```
Panel  Utilities  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                        OBJECT STORAGE GROUP DEFINE            Page 2 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : TEST

To DEFINE Storage Group, Specify:
  Volume Full Threshold  . . . .       (0-9999)
  Drive Start Threshold  . . . .       (0-9999)
  Volume Full at Write Error . .       (Y or N)

  OAM Deletion Protection  . . . N     (Y=Enable or N=Disable)
  OAM Retention Protection . . . N     (Y=Enable or N=Disable)









Use ENTER to Perform Verification; Use UP Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 3-22 Changes in the DEFINE/ALTER panel

There are modifications to the LISTPRT panel, as shown in Figure 3-23 and Figure 3-24 on page 71.

```
STORAGE GROUP LIST
Command ===> LISTPRT                                        Scroll ===> CSR
                                                         Entries 1-1 of 1
                                                         Data Columns 3-6 of 48
CDS Name : SYS1.SMS.SCDS

Enter Line Operators below:

     LINE       STORGRP  SG                VIO      VIO   AUTO
     OPERATOR   NAME     TYPE              MAXSIZE  UNIT  MIGRATE
    ---(1)----  --(2)--- -------(3)------  --(4)--  (5)-  --(6)---
                TOGRP1   OBJECT            -------  ----  --------
    ----------  -------- ------  BOTTOM  OF  DATA  ------  --------  ------
```

Figure 3-23 First panel of LISTPRT

```
Panel  Utilities  Scroll  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                       STORAGE GROUP PRINT ENTRY PANEL            Page 2 of 3
Command ===>

Select Format Type  . . . . . 1  (1 - Standard, 2 - Roster)
Report Data Set Name  . . . .
Replace Report Contents . . . N  (Y or N)  Lines/Page . . . 55  (12 to 99)
Specify Tags to be Printed:
===>  *

 (17)  Last Modified Userid            (15)  MB Free Space
 (19)  Last Time Modified              (14)  MB Total Space
 (29)  Library Name                     (7)  Migrate System or Sys Group
 (30)  Library Name                    (12)  Migration High
 (31)  Library Name                    (45)  Migration High Track
 (32)  Library Name                    (13)  Migration Low
 (33)  Library Name                    (46)  Migration Low Track
 (34)  Library Name                    (47)  OAM Deletion Protection
 (35)  Library Name                    (48)  OAM Retention Protection
 (36)  Library Name                    (40)  OSMC Processing System

Use ENTER to Perform Print; Use UP/DOWN Command for previous Panels;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 3-24 Change in Storage Group LISTPRT panel

The changes in the ISMF STORAGE GROUP SORT ENTRY PANEL are shown in Figure 3-25.

```
 Panel  Utilities  Scroll  Help
 ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                        STORAGE GROUP SORT ENTRY PANEL            Page 2 of 3
 Command ===>

 Specify One or More Attribute Numbers for Sort Sequence:
   Major Field . . . . 2   Minor Field 1 . . .     Minor Field 2 . . .
 Specify A for Ascending or D for Descending Sort Order:
   Major Field . . . . A   Minor Field 1 . . .     Minor Field 2 . . .

  (17)  Last Modified Userid                (15)  MB Free Space
  (19)  Last Time Modified                  (14)  MB Total Space
  (29)  Library Name                         (7)  Migrate System or Sys Group
  (30)  Library Name                        (12)  Migration High
  (31)  Library Name                        (45)  Migration High Track
  (32)  Library Name                        (13)  Migration Low
  (33)  Library Name                        (46)  Migration Low Track
  (34)  Library Name                        (47)  OAM Deletion Protection
  (35)  Library Name                        (48)  OAM Retention Protection
  (36)  Library Name                        (40)  OSMC Processing System


 Use ENTER to Perform SORT; Use UP/DOWN Command to View previous Panels;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 3-25 Changes in Storage Group SORT panel

3.2.13 Changes in NaviQuest

NaviQuest is a testing and reporting tool that speeds and simplifies the tasks that are associated with DFSMS initial implementation and ongoing ACS routine and configuration maintenance. NaviQuest provides the following functions:

- A familiar ISPF panel interface to functions
- Fast and easy bulk test case creation
- ACS routine and DFSMS configuration testing automation
- Storage reporting assistance
- Additional tools to aid with storage administration tasks
- Batch creation of data set and volume listings
- Printing of ISMF LISTs
- Batch ACS routine translation
- Batch ACS routine validation

In z/OS V1.11, NaviQuest JCL for DEFINE, ALTER, DISPLAY Object Storage Group will be modified to support the new enhancements. Example 3-23 shows JCL to define and alter a copy pool.

Example 3-23 Example NaviQuest JCL to DEFINE/ALTER Object Storage Group

```
//MHLRES4N   JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES4,REGION=0M
//MYLIB JCLLIB ORDER=SYS1.SACBCNTL
//*****************************************************************/
//*$MAC(ACBJBAJB) COMP(5695DF123): BATCH - OBJECT SG DEFINE/ALTER    */
//*                                                                 */
...
//*****************************************************************
//STEP1   EXEC ACBJBAOB,
//        TABL2=MHLRES4.TEST.ISPTABL
//SYSUDUMP DD  SYSOUT=*
//SYSTSIN  DD *
ISPSTART CMD(ACBQBAJB +
DEFINE/ALTER +
SCDS(TEST.CDS) +
STORGRP() +
DESCR() +
QUALFR() +
CYCLEST() +
CYCLEET() +
LIBNAME() +
VOLFTHRS() +
DRVSTHRS() +
VOLFWERR() +
OAMDELPRO() +
OAMRETPRO() +
SGSTATUS() +
SGSTSALL() +
UPDHLVLSCDS() +
)
/*
```

3.2.14 Migration considerations

Regardless of whether or not the installation intends to exploit the new function, you must modify and run the CBRSMR1B migration job to add the new ODSTATF and ODRETDT columns to your object directory tables. Bind jobs must be run as documented in Chapter 3, “Migrating, Installing, and Customizing OAM”, of *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426.

3.2.15 Coexistence considerations

Coexistence support (PTFs) will be provided at pre-DFSMS V1.11 release levels for the OAM archive retention enhancements. Pre-V1.11 level systems in the OAMplex will not recognize the archive retention enhancements introduced in V1.11, and therefore could inadvertently bypass the protection modes defined on the V1.11 level system. For this reason, do not exploit the archive retention enhancements until all the systems in the OAMplex are migrated to z/OS V1.11 or higher.

Note: Retention-protection and deletion-protection can be activated at an object Storage Group level on a V1.11 system. In a mixed level OAMplex environment, if an installation opts to activate retention-protection or deletion-protection for a given object Storage Group, then they should mark the object storage group status as NOTCON for any pre-V1.11 level systems to ensure the protection is not overridden. The object storage group status is set via the Storage Group Alter panel in ISMF.

3.3 Wildcard characters for OSREQ QUERY enhancement

Prior to V1.11, a generic search could be requested by putting a single asterisk (*) in the right most qualifier of the name. Now a percent sign (%) and underscore (_) wildcard are implemented. One or more percent signs (%) or underscores (_) can be used anywhere in the object name. A mix of asterisk and the new wildcard characters is not supported.

The QUERY function obtains descriptive information about an object within a collection. The object information is presented in query element (QEL) format.

QUERY searches the directory containing the objects that belong to the collection name specified in the COLLECTN keyword parameter for a match on the fully qualified object name specified in the NAME keyword parameter, and returns a single query element (QE). QUERY also supports a generic search that returns a QE for each object whose name matches the partially qualified name specified in the NAME keyword.

Request a generic search by one of the following methods.

- Substituting an asterisk (*) for the right most part of the name (right most qualification level). This indicates that the search request applies to all objects whose names match the characters to the left of the asterisk. For example, 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.MAIL.PEL* 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.
- Substituting one or more percent signs (%) or underscores (_) anywhere in the object name. The percent sign character is interpreted as a wildcard to replace zero or more characters in the object name. The underscore character represents a single character. For example, 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.

Note: You cannot mix the asterisk wildcard with either a percent sign or underscore wildcard in a single QUERY request. The generic search is only supported for OSREQ QUERY requests.

Example 3-24 shows a sample job to QUERY a generic name.

Example 3-24 Using a generic tag

```
READY
 OSREQ QUERY    TOGRP1.TESTXXX TOGRP1.*
CBRO400I OSREQ QUERY successful. Return code = 00000000, reason code = 00000000.
CBRO404I OSREQ QUERY response time is 461 milliseconds.
CBRO419I ********************************************************************
```

```
CBRO410I Collection name         = TOGRP1.TESTXXX
CBRO411I Object name             = TOGRP1.TESTXXX
CBRO412I Object size             = 2097152000
CBRO413I Creation date           = 2009-09-01
CBRO414I Creation timestamp      = 20.57.26.934107
CBRO415I Last referenced date    = 2009-09-01
CBRO416I Expiration date         = 2010-10-06
CBRO417I Management class        = TOGRP1D1
CBRO418I Storage class           = OBJDASD
CBRO427I Primary retrieve key    = '00000000000000000000'X
CBRO428I Backup retrieve key     = '00000000000000000000'X
CBRO429I Backup2 retrieve key    = '00000000000000000000'X
CBRO430I Estimated retrieve time = 300
CBRO443I Location                = D
CBRO444I Pending action date     = 2009-09-01
CBRO445I Status flags            = 0
CBRO446I Retention protect date  = 0001-01-01
CBRO447I Deletion-protected      = N
READY
END
```

Note: For OSREQ generic QUERY, the percent ('%') and underscore ('_') wildcard characters introduced in z/OS Version 1 Release 11 cannot be used with the TSO/E command processor. Only the asterisk ('*') wildcard character is valid for use with the TSO/E OSREQ command processor.

See *z/OS DFSMS OAM Application Programmer's Reference*, SC35-0425 and *z/OS V1R11 DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*, SC35-0426 for detailed information about OSREQ QUERY.

3.4 Utility to modify OAM collection defaults enhancement

This enhancement provides a utility to modify the Storage Class and Management Class defaults associated with an OAM collection.

3.4.1 Overview

Prior to Version 1.11, installations were required to execute a five step manual process, as documented in the OAM PISA, in order to modify existing Storage Class and Management Class defaults associated with a given OAM collection.

The following manual procedure to change the default Storage Class, Management Class, or both for an existing object Storage Group's collection ID is documented in the *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426:

1. Delete the collection definition from the catalog:

   ```
   DELETE COLLECTION_NAME NONVSAM CATALOG ('CATALOG_NAME') -
          FILE (DD1) PURGE NOSCRATCH
   ```

2. Change the Storage Class, Management Class, or both in the DB2 collection table to the desired value. Ensure that these values in the SMS CDS are valid.

3. Add the collection entry back to the catalog:

   ```
   DEFINE NVSAM (NAME(COLLECTION_NAME) COLLECTION RECATALOG)
   ```

   This creates a collection entry without a Management Class or Storage class.

4. Use OSREQ RETRIEVE to retrieve an existing object in the collection, or OSREQ STORE to store a new object into the collection. This command updates the catalog version.

5. Verify that the catalog matches DB2:

   ```
   LISTCAT CATALOG('CATALOG_NAME') FILE(DD1) ALL NVSAM
   ```

The new default Storage Class, Management Class, or both only applies to new objects stored after these changes are complete.

3.4.2 New enhancement: CHGCOL utility

In z/OS 1.11, the manual procedure listed in 3.4.1, “Overview” on page 75 can be replaced by using the OAM CHGCOL utility to change the default Storage Class name or default Management class name of a collection. To invoke this utility, modify and run the CBRSAMUT SAMPLIB job or issue a TSO/E OAMUTIL command to start the utility. This utility allows you to perform the following tasks:

- Change the default Storage Class Name only.
- Change the default Management Class Name only.
- Change both the default Storage Class Name and default Management Class Name.

Figure 3-26 shows the syntax of OAMUTIL CHGCOL.

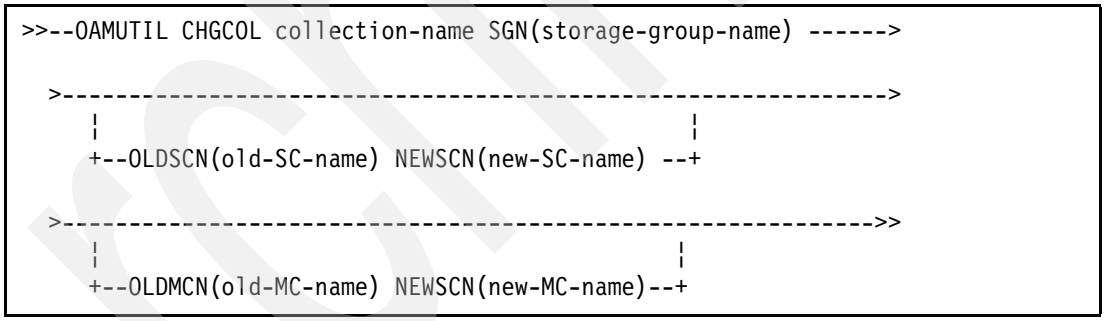


Figure 3-26 OAMUTIL CHGCOL command syntax

3.5 CBRUXSAE enhancements

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 return code 16 (‘Bypassed’), meaning that the current and all future user IDs are authorized to perform all OSREQ functions and that the exit does not need to be called again. Installations must substitute this code with a validation routine to determine the authority for a specific user ID in order for authorization checking to be performed at the application interface level.

In z/OS V1.11, this support provides more granular return codes to be processed by the CBRUXSAE security authorization user exit. Specifically, the CBRUXSAE sample will be modified to include the new return code values.

The additional return codes enable an installation to code the 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.
- Authorize users to store objects in an existing collection while preventing them from creating new collections.
- Issue return and reason codes, if the return code from CBRUXSAE is not 0, 16, or 255 (or 253 or 254 when storing to an existing collection) that indicate that the user ID is not authorized to perform the particular OSR function. For more information concerning return and reason codes associated with this exit, see *DFSMSdfp Diagnosis*, GY27-7618.

3.5.1 Migration and coexistence considerations

Verify that the CBRUXSAE exit is not using IBM reserved values 224-255.

3.6 New messages

Here we discuss messages that are new in V1.11.

3.6.1 CBR0435I

```
CBR0435I CHGCOL rejected due to invalid specification.
```

Explanation: Either collection-name was not specified or invalid collection-name was specified.

3.6.2 CBR0436I

```
CBR0436I CHGCOL unsuccessful for collection collection-name. {Collection not found
in catalog | Error returned from catalog alter | Catalog error occurred |
Collection not found in DB2 | Error attempting to locate collection in DB2 | Error
attempting to delete collection in DB2 | Error defining collection to DB2 | Error
returned from DB2 services | Error returned from PC routine | Internal error
occurred when processing CHGCOL request | OLDSCN does not match catalog | OLDSCN
does not match DB2 | OLDMCN does not match catalog | OLDMCN does not match DB2 |
SGN does not match catalog | SGN does not match DB2}. Return code = return-code,
Reason code = reason-code.
```

Explanation: The OAMUTIL CHGCOL request ended in an error. The specific errors are displayed and explained.

3.6.3 CBR0437I

```
CBR0437I OAMUTIL CHGCOL request for collection collection-name has caused the
catalog and DB2 to be out of synch.
```

Explanation: The OAMUTIL CHGCOL request has successfully updated DB2, but an error was encountered while updating the catalog. Because the catalog was not updated, an attempt has been made to back out the successful update to DB2 to keep the catalog and DB2 synchronized, with the intention of leaving the catalog and DB2 as they were prior to the issuance of the OAMUTIL CHGCOL TSO command. The attempt to update DB2 back to its original values has failed, thus causing the catalog and DB2 to be out of sync.

3.6.4 CBR0443I

```
CBR0443I Location = location-flag
```

Explanation: The OSREQ QUERY request was issued and completed successfully. location-flag indicates where the primary copy of the object resides. location-flag values are interpreted as follows:

- A D object resides on DASD.
- An R object resides on DASD in recalled mode.
- A T object resides on a tape sublevel 1 volume.
- A U object resides on a tape sublevel 2 volume.
- A blank object resides on an optical volume.

3.6.5 CBR0444I

```
CBR0444I Pending action date = pending-action-date
```

Explanation: The OSREQ QUERY request was issued and completed successfully. pending-action-date, in the form YYYY-MM-DD, is the next date the object is eligible to be selected for processing by OSMC storage management cycle.

3.6.6 CBR0445I

```
CBR0445I Status flags = status-flags
```

Explanation: The OSREQ QUERY request was issued and completed successfully. status-flags reflects the value contained in the ODSTATF field in the object directory row associated with this object.

Table 3-6 shows valid values for ODSTATF and how they map to ODSTATF status flags.

Table 3-6 Values for ODSTATF and how they map to ODSTATF status flags

ODSTATF Value	ODSTATF_EBR	ODSTATF_DELHOLD	ODSTATF_RETPROT
0	OFF	OFF	OFF
1	OFF	OFF	ON
2	OFF	ON	OFF
3	OFF	ON	ON
4	ON	OFF	OFF
5	ON	OFF	ON
6	ON	ON	OFF
7	ON	ON	ON

The ODSTATF status flags have the following meaning:

- When ODSTATF_EBR is ON, the object is in event-based-retention mode.
- When ODSTATF_DELHOLD is ON, the object is in deletion-hold mode.
- When ODSTATF_RETPROT is ON, the object is in retention-protection mode.

3.6.7 CBR0446I

```
CBR0446I Retention protect date = retention-protect-date
```

Explanation: The OSREQ QUERY request was issued and completed successfully. retention-protect-date, in the form YYYY-MM-DD, is the date that a retention-protected object is eligible to be deleted.

3.6.8 CBR0447I

```
CBR0447I Deletion protected = deletion-protect-mode
```

Explanation: The OSREQ QUERY request was issued and completed successfully. deletion-protect-mode indicates whether the object is deletion-protected:

- **Y** object is deletion-protected.
- **N** object is not deletion-protected.

3.6.9 CBR6429I

```
CBR6429I Error writing OAM tape [sublevel 1 or backup | sublevel 2] object
object-name, collection collection-name in storage group storage-group-name to MVS
scratch tape volume volser. The volume capacity volume-capacity [KB | MB] is less
than the object size object-size KB. [DATACLASS | L2DATACLASS]=data-class-name,
[TAPEUNITNAME | L2TAPEUNITNAME]=tape-unit-name. OAM return code=return-code,
reason code=reason-code.
```

Explanation: An attempt to write object object-name, collection collection-name to tape failed because the volume capacity volume-capacity of the MVS scratch tape allocated is less than the object size object-size.

3.6.10 CBR6430I

```
CBR6430I OAM [tape sublevel 1 or backup | tape sublevel 2] write request failed
for object object-name, collection collection-name in storage group
storage-group-name. Object size=object-size KB, [DATACLASS |
L2DATACLASS]=data-class-name, [TAPEUNITNAME | L2TAPEUNITNAME]=tape-unit-name, MVS
scratch tape threshold=object-threshold-size [KB | MB]. OAM return
code=return-code, reason code=reason-code.
```

Explanation: An attempt to write object, collection to [tape sublevel 1 or backup | tape sublevel 2] failed because MVS dynamic scratch tape allocation for objects greater than object-threshold-size [KB | MB] is disabled for Storage Group.

3.6.11 CBR9109I

```
CBR9109I The control task ctcname process process-name could not [acquire |
release] a buffer manager token during [read | route] processing for collection
collection-name, object object-name in storage group storagegroup-name. OAM return
code = return-code, reason code = reason-code.
```

Explanation: The control task ctcname process process-name either could not acquire or release a buffer manager token. An error occurred within the OAM Buffer Manager while trying to obtain or release a buffer manager token during read or route processing.

3.7 Updated messages

The following messages have been updated:

3.7.1 CBR0014I

```
CBR0014I Invalid option option specified with keyword {MSG= | OTIS= | UPD= | MOS=
| LOB= | QB= | DP=}, for the OAM entry in IEFSSNxx parmlib member. Default option
for the keyword is assumed.
```

Explanation: One of the keyword options specified in the OAM entry in the IEFSSNxx member of PARMLIB was specified incorrectly.

3.7.2 CBR0434I

```
CBR0434I {REFORMAT | CHGCOL} successful.
```

Explanation: The OAMUTIL request completed successfully.

3.7.3 CBR1100I

```
CBR1100I OAM STATUS:
```

Explanation: The OAM status is:

```
OPT. TOT USE TOT USE AVL TOT  USE   AVL  TOT  USE  AVL SCR REQ
     LIB LIB DRV DRV DRV LDR  LDR   LDR  SDR  SDR  SDR VOL CT
     aaa bbb ccc ddd eee fff  ggg   hhh  iii  jjj  kkk lll mmm
```

```
TAPE TOT ONL TOT TOT TOT TOT  TOT   ONL  AVL  TOTAL
     LIB LIB AL  VL  VCL ML   DRV   DRV  DRV  SCRTCH
     nnn ooo ppp qqq rrr sss  tttt  uuuu vvvv wwwwwww
exit-name processing {ENABLED|DISABLED|BYPASSED|OPERATOR DISABLED}.
CBRUXSAE processing {ENABLED|BYPASSED|BYPASSED_RS} for STORE.
CBRUXSAE processing {ENABLED|BYPASSED} for RETRIEVE.
CBRUXSAE processing {ENABLED|BYPASSED} for QUERY.
CBRUXSAE processing {ENABLED|BYPASSED} for CHANGE.
CBRUXSAE processing {ENABLED|BYPASSED} for DELETE.
Access Backup {ACTIVE|INACTIVE}for reason, using {1st|2nd|no} backup copy.
DB2 SSID: db2-ssid
XCF GROUP NAME: group-name
XCF MEMBER NAME: member-name
CBROAM: cbroam-parmlib-suffix
OAM1 Parms: TIME=xxx  MSG=xx  UPD=x  QB=x
            MOS=xxxx  OTIS=x  LOB=x  DP=x
```

The operator has entered the following command:

```
D SMS,OAM
```

A display of OAM address space status has been generated. If both optical and tape libraries have been defined in the SMS configuration, the sample display will be generated. Otherwise, only the data for the library type defined will be generated.

3.7.4 CBR1130I

```
CBR1130I      OAM storage group status:
```

Explanation:

```
OBJECT TY  REQ   OSMC     BACKUP   BACKUP   RET  DEL
STORGRP    COUNT SYSTEM   STORGRP1 STORGRP2 PRO  PRO
sgname   a   bbbbb  sysname    objbusg1    objbusg2    n      o
TAPE       DATA         L2TAPE        L2DATA
UNIT       CLASS        UNIT          CLASS
unitname     dataclass       unitname2       dataclass2
LIBRARYNAMES:     libname1  libname2  libname3  libname4
                  libname5  libname6  libname7  libname8
TAPE      LIBRARY
STORGRP   NAMES
sgname     libname1   libname2   libname3   libname4
           libname5   libname6   libname7   libname8
```

The operator has entered one of the following commands:

```
DISPLAY SMS,STORGRP(storage-group-name),DETAIL
     DISPLAY SMS,STORGRP(ALL),DETAIL
```

A display of OAM storage group status has been requested. When a storage group name is supplied, there is one data line that describes the specified storage group; when ALL is supplied, there is one data line for each storage group in the configuration. If both object and tape storage groups have been defined in the SMS configuration, the sample display above is generated. Otherwise, only the data for the storage group type defined is generated.

3.7.5 CBR9103I

```
CBR9103I A {Read | Write} error occurred during storage management processing for
{primary | backup | backup2} for collection collection-name, object object-name in
Storage Group storagegroup-name on volume volser . The return code is return-code
and the reason code is reason-code.
```

Explanation: The error was detected during processing in preparation of a read or write request. Retries were attempted and were also unsuccessful. The error might be due to a problem with the configuration database, the operating environment, or with the optical or tape library and media. If this was a read error, the volume will be the volume for which the read was attempted. If this was a write error, the volume will be N/A.

3.7.6 CBR9225I

```
CBR9225I One or more objects in collection collection-name were not processed by
OSMC {Object Processing | DASD Space Manager | Object Recovery | Volume Recovery |
Move Volume} because the object's {current location | size} is not supported by
this level of OAM.
```

Explanation: During the OAM storage management component (OSMC) process specified, one or more objects were encountered that either currently reside in a LOB storage structure or are greater than 268 435 456 bytes.

3.7.7 CBR9226I

```
CBR9226I One or more write requests to [tape or optical | tape | optical] were not
attempted for objects larger than the maximum object size supported by the
specified level of the OAM storage hierarchy in collection collection-name for
OSMC [Object Storage Group | Move Volume | Recovery] Processing.
```

Explanation: One or more objects were encountered in collection collection-name during the specified OAM Storage Management Component (OSMC) Processing. The processing requires writing the object or an object backup copy to the tape or optical levels of the OAM storage hierarchy; however, the object size exceeds the maximum object size supported for the tape or optical level. The write to tape or optical was not attempted because the object size exceeds the maximum size supported for the specified storage hierarchy level for this release of OAM.

4

Extended Address Volume enhancements

In this chapter, we describe the second wave of Extended Address Volume (EAV) enhancements delivered in z/OS V1.11. A summary of the first wave delivered in z/OS V1.10 is also given to provide a comprehensive understanding of EAV as it stands now.

The introduction of EAV at z/OS V1.10 raised a number of coexistence issues; with z/OS V1.11, EAV new migration and coexistence issues are raised. Because of the differences between EAV R1 and R2 (z/OS V1.10 and V1.11), there are coexistence issues between z/OS V1.10 and V1.11. We consider specific and general coexistence with pre-z/OS V1.10 systems that do not support EAVs but can tolerate their existence in limited ways.

4.1 EAV terminology

Note: Two sets of terms are used to reference an EAV. One set describes how space is managed and the other describes how the disk is addressed. The context of what is being described will dictate which terminology to use.

Extended Address Volume (EAV)
A volume with more than 65,520 cylinders. Only 3390 Model A devices can be an EAV.

Track address
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.

Extended Addressing Space (EAS)
On an Extended Address Volume, the cylinders whose addresses are equal to or greater than 65,536. These cylinder addresses are represented by 28-bit cylinder numbers.

Base addressing space
On an Extended Address Volume, the cylinders whose addresses are below 65,536. These cylinder addresses are represented by 16-bit cylinder numbers or by 28-bit cylinder numbers whose high order 12 bits are zero.

Multicylinder unit
A fixed unit of disk space that is larger than a cylinder. Currently, on an EAV, a multicylinder unit is 21 cylinders and the number of the first cylinder in each multicylinder unit is a multiple of 21.

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 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 exists only on EAV volumes.

Track-managed space
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.

Breakpoint value (BPV)
When a disk space request is this size or more, the system prefers to use the cylinder-managed space for that extent. This applies to each request for primary or secondary space for data sets that are eligible for the cylinder-managed space. If not enough cylinder-managed space is available, then the system will use the track-managed space or will use both areas. The breakpoint value is

expressed in cylinders. When the size of a disk space request is less than the breakpoint value, the system prefers to use the track-managed area, and if enough space is not available there, then the system will use the cylinder-managed space or will use both areas.

4.2 EAV z/OS V1.10 enhancement summary

With EAV, IBM is implementing an architecture that, over a number or releases, will provide capacities of hundreds of terabytes for a single volume, with an architectural maximum of 268,434,453 cylinders (225 TB). The first release of EAV in z/OS V1.10 was limited to a volume with a maximum of 223 GB (a maximum of 262,668 cylinders), which still remains in z/OS V1.11.

4.2.1 3390 Model A

To use EAV capacity, the volume has to be configured in an IBM System Storage DS8000 subsystem as a 3390 Model A. However, a 3390 Model A is not always an EAV. A 3390 Model A is any device configured in the IBM System Storage DS8000 subsystem to have from 1 to a possible 268,434,453 cylinders.

Figure 4-1 illustrates the 3390 device types.

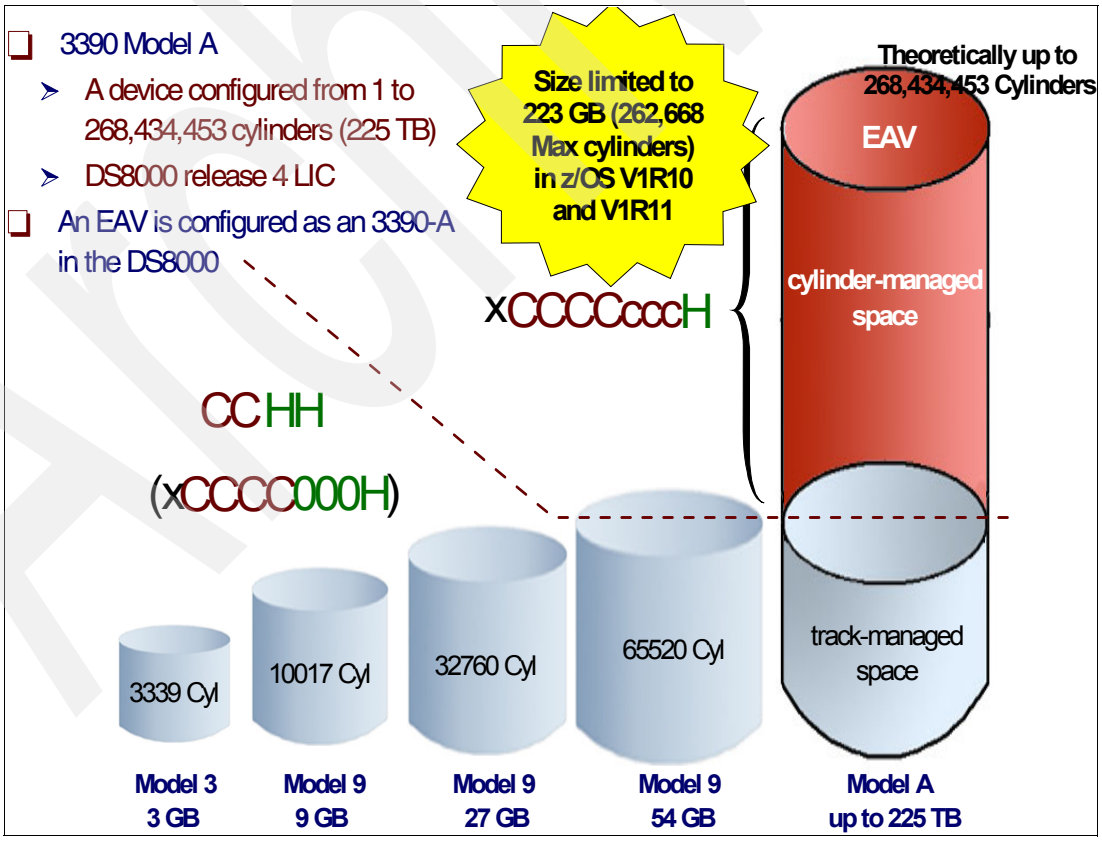


Figure 4-1 3390 device types

Important: The EAV support is provided in IBM System Storage DS8000 V4.0.

Note: With the 3390 Model A, the model A refers to the model configured in the IBM System Storage DS8000. It has no association with the 3390A notation in HCD that indicates a PAV-alias UCB in the z/OS operating system. The Model “A” was chosen so that it did not imply a particular device size as previous models 3390-3 and 3390-9 did.

4.2.2 Extended Address Volume attributes

In z/OS V1.10, each Extended Address Volume (Figure 4-2) has the following attributes. However, in a later release, they might be independent of each other. The new volume attributes are provided in the VTOC’s Format 4 data set control blocks (DSCBs) and the UCB’s device class extension. A volume with more than 65,520 cylinders requires Format 8 and Format 9 DSCBs to handle the 28-bit cylinder number. A normal volume has none of these attributes.

In order to qualify as an EAV, it is not necessary to have all the attributes listed in Figure 4-2, but an EAV has at least the following three attributes:

- The volume supports an extended addressing space (EAS).
- The volume supports cylinder-managed space.
- The volumes supports extended attribute DSCBs.

The other attributes associated with EAVs will be created after the EAV is populated with data sets, especially data sets the reside in the extended addressing space.

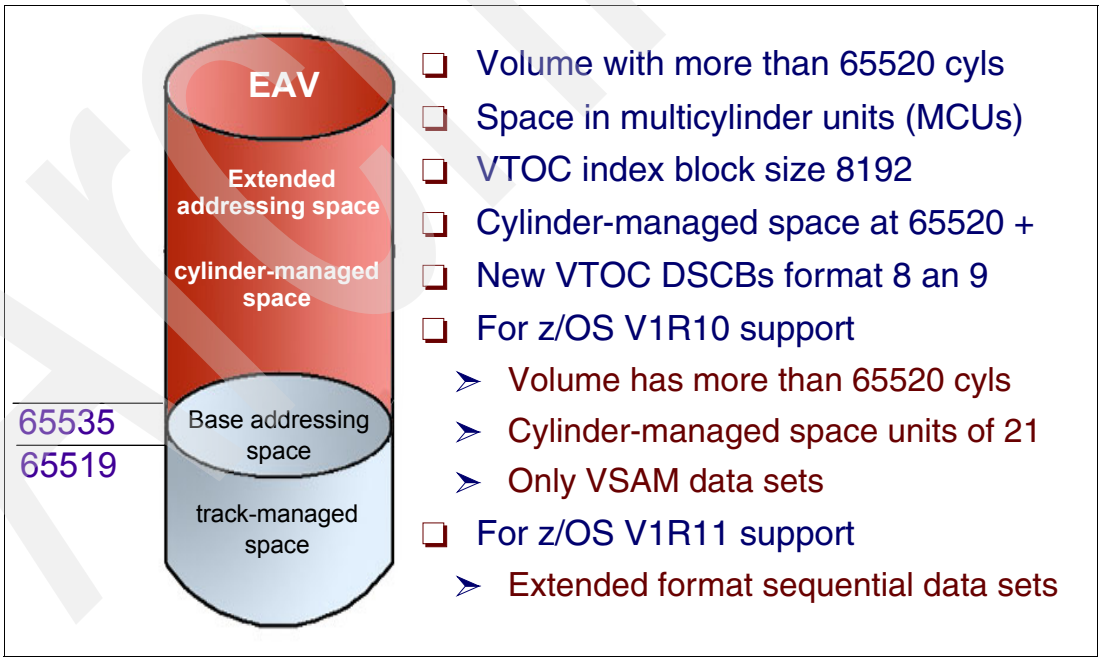


Figure 4-2 EAV volume attribute summary

4.2.3 28-bit cylinder addresses

Cylinder track addresses have now changed to use a 28-bit cylinder address from a 16-bit address. From *CCCC*HHHH to *CCCCcccH*, where CCCC are the lower order 16 bits, ccc are the high order 12 bits, and H is a 4-bit track number. The number of tracks per cylinder has been fixed at 15 since the creation of the 3380 geometry, so only a 4-bit track address is required. Essentially, 12 of the 16 bits were wasted, so the track address has been reduced from 16 bits to 4 bits. Thus, the overall size of the cylinder track address still remains at 32 bits and has not changed; just the internal format for cylinder number and track number has changed.

The new 28-bit cylinder addressing architecture still allows existing programs to address tracks in the base addressing space. The extended addressing space has a method to protect existing programs from accessing tracks in the EAS. This is done with the new data set control blocks (DSCBs) in the VTOC, which are discussed in 4.2.7, “New format DSCBs for EAV volumes” on page 92.

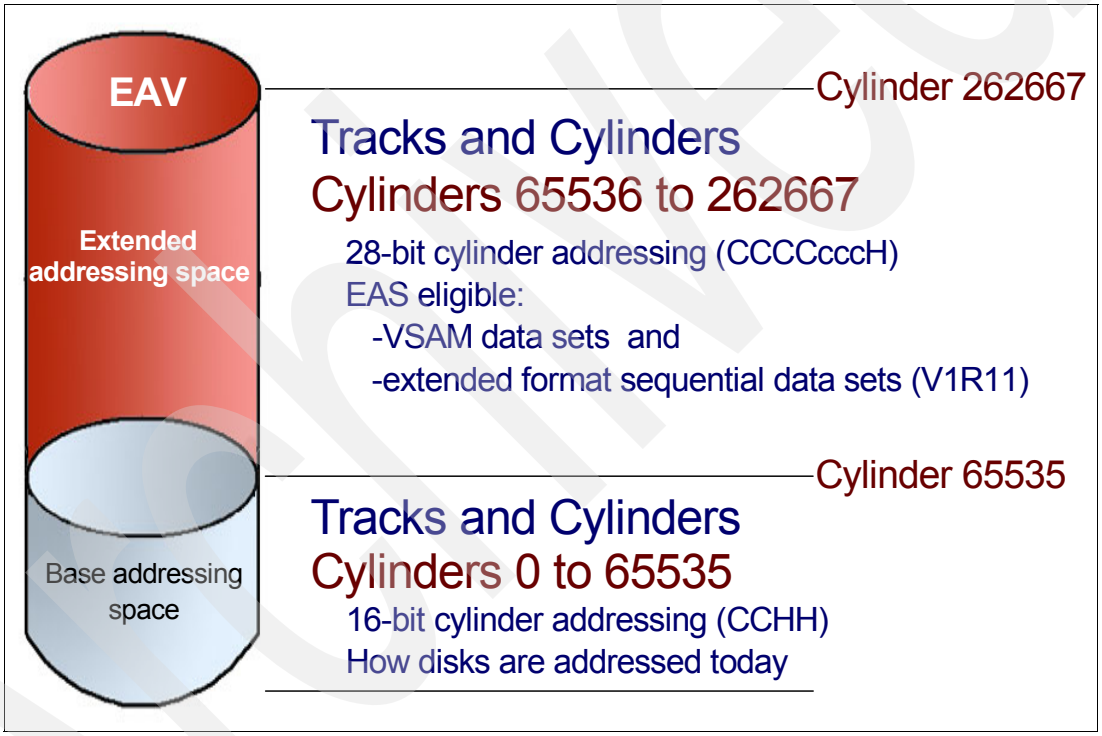


Figure 4-3 New track addressing for EAV volumes

Track address for non-EAV

The base addressing space is the area on an EAV located within the first 65536 cylinders. As shown in Figure 4-4, tracks are addressed in this area with 16-bit cylinder numbers described with the CCHH notation:

- CC represents 16 bits for a cylinder address.
- HH represents 16 bits for a track address, of which only the low order 4 bits are used.

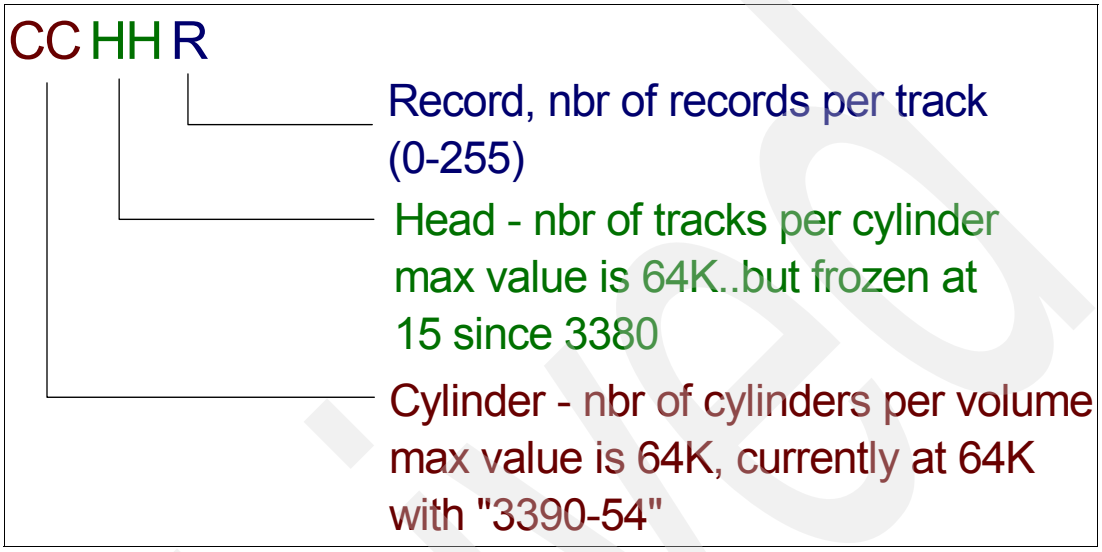


Figure 4-4 Old track address

This is how all disks are addressed today. We generally refer to a track address using the CCHH notation. However, now a track address can be shown using the CCCCHHHH notation. This track address is a 32-bit number that addresses each track within a volume. Each cylinder and track number uses a 16-bit number. For the track number only, the low order 4 bits are used. The high order 12 bits of the track number are not used. Thus, to handle cylinder numbers greater than 65,520, a new format for the track address is required.

New track address for Extended Address Volume

The extended addressing space (EAS) is the area on an EAV located above the first 65536 cylinders. Tracks are addressed in this area with 28-bit cylinder numbers and described with the CCCCcccH (showing hex digits) notation:

- CCCC represents the low order 16 bits of a 28-bit number.
- ccc represents the high order 12 bits of a 28-bit number.
- H represents a 4-bit track number.

This addressing is comparable to all 16-bit cylinder addressing. This area is similar to the cylinder-managed spaces, but is a subset of it. We often interchange the terminology of EAS and cylinder-managed space. This area is similar to the track-managed space, but has a larger set of cylinders.

The 28-bit cylinder addressing architecture allows existing programs to address tracks in the base addressing space. The extended addressing space provides a method to protect existing programs from accessing tracks in the EAS. This action is accomplished with the new data set control blocks (DSCBs) in the VTOC, as discussed in 4.2.7, “New format DSCBs for EAV volumes” on page 92.

For compatibility with older programs, the ccc portion is hexadecimal 000 for tracks in the base addressing space. This track address method is referred to as a 28-bit cylinder number. This format preserves the 3390 track geometry. Track addresses for space in track-managed space will be comparable to today's track addresses. However, track addresses for space in cylinder-managed space will *not* be comparable to previous track addresses.

Note: For compatibility reasons, the 32 bits in each track address on an EAV is in the CCCCcccH format. The 12 high order bits of the cylinder number are in the high order 12 bits of the two old HH bytes. This format might be written as CCCCcccH, as shown in Figure 4-5 on page 89.

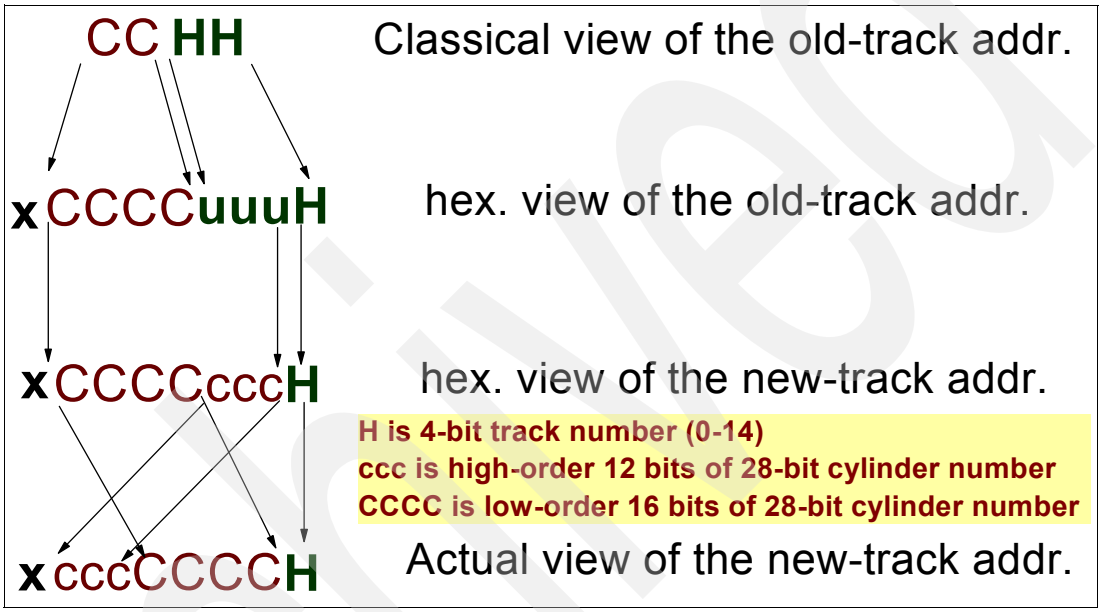


Figure 4-5 From old track address to new track address

Key points

The cylinder number is in a non-contiguous form. Reading this new track address, the hex digits must be rearranged, as shown in Figure 4-6. This format preserves the 3390 track geometry. Track addresses in existing channel programs and extent descriptors in DSCBs and elsewhere are in the form of *CCHH*, where CC is the 16-bit cylinder number and HH is the 16-bit track number in that cylinder. If the volume is an EAV, the cylinder number in these four CCHH bytes is 28 bits and the track number is four bits. For compatibility reasons, the 32 bits in each track address on an EAV are in the format CCCCcccH.

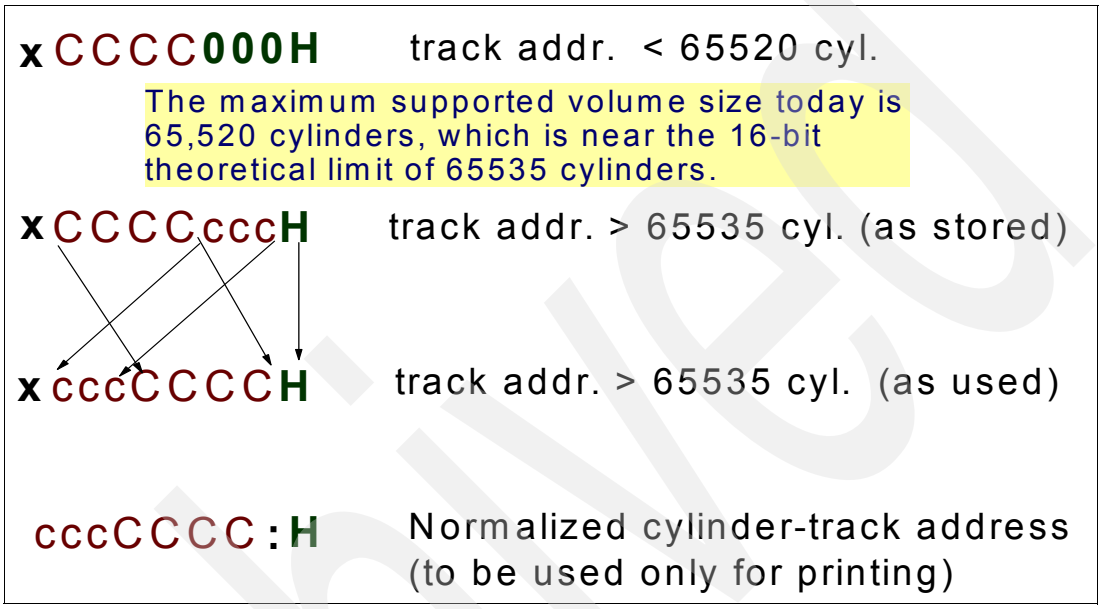


Figure 4-6 EAV: The two types of track address

4.2.4 Cylinder managed space

Cylinder managed space is the new scheme by which additional space in the EAV beyond 65,520 cylinders is managed. Because of the sheer amount of the cylinders, a new method was required to manage the space without unduly increasing the size of the VTOC and VTOC index. The following sections describe the principles, components, and development of the scheme.

Addressing Extended Address Volumes

An EAV has two addressing spaces:

- The base addressing space

  The base addressing space is the area on an EAV located within the first 65,520 cylinders (that is, cylinder 0-65,519), as shown in Figure 4-3 on page 87. Tracks are addressed in this area with 16-bit cylinder numbers and are described with the CCHH notation. The CC represents 16 bits for a cylinder address. The HH represents 16 bits for a track address, of which only the low order 4 bits are used. This is how all disks are addressed today.

- The extended addressing space (EAS)

  The extended addressing space (EAS), shown in Figure 4-3 on page 87, is the area on a EAV located above the first 65,536 cylinders. Tracks are addressed in this area with 28-bit cylinder numbers, described with the CCCCcccH notation, as follows:

  - The CCCC represents the low order 16 bits of a 28-bit number.
  - The ccc represents the high order 12 bits of a 28-bit number.
  - The H represents a 4-bit track number.

Cylinder-managed space

Cylinder-managed space is used beyond cylinder 65,520, while traditional track-managed space is still used for the first 65,520 cylinders, that is, from cylinder 0 to cylinder 65519. The area from cylinder 65,520 and above is known as Extended Addressing Space (EAS).

The cylinder-managed space is the space on the volume that is managed only in multicylinder units (MCUs)**.** 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 is rounded up to the next multicylinder unit. The cylinder-managed space exists only on EAV volumes. A data set allocated in cylinder-managed space may have its requested space quantity rounded up to the next MCU.

Data sets allocated in cylinder-managed space are described with a new type of data set control blocks (DSCB) in the VTOC. Tracks allocated in this space will also be addressed using the new track address. Existing programs that are not changed will not recognize these new DSCBs and therefore will be prevented from seeing how the tracks in cylinder-managed space are addressed.

Track-managed space

The track-managed space is the space on a volume that is managed in *track* and cylinder increments. All volumes today have track-managed space. Track-managed space ends at cylinder number 65519. Each data set occupies an integral multiple of tracks. Each data set occupies an integral multiple of tracks. The track-managed space allows existing programs and physical migration products to continue to work. Physical copies can be performed from a non-EAV to an EAV and make those data sets accessible.

4.2.5 Multicylinder units

Multicylinder units consist of 21 cylinders. Space requests for allocations on an EAV in EAS will be rounded up to the nearest multiple of this amount.

A multicylinder unit (MCU) is a fixed unit of disk space that is larger than a cylinder. Currently, on an EAV volume, a multicylinder unit is 21 cylinders and the number of the first cylinder in each multicylinder unit is a multiple of 21. Figure 4-7 shows the EAV and multicylinder units.

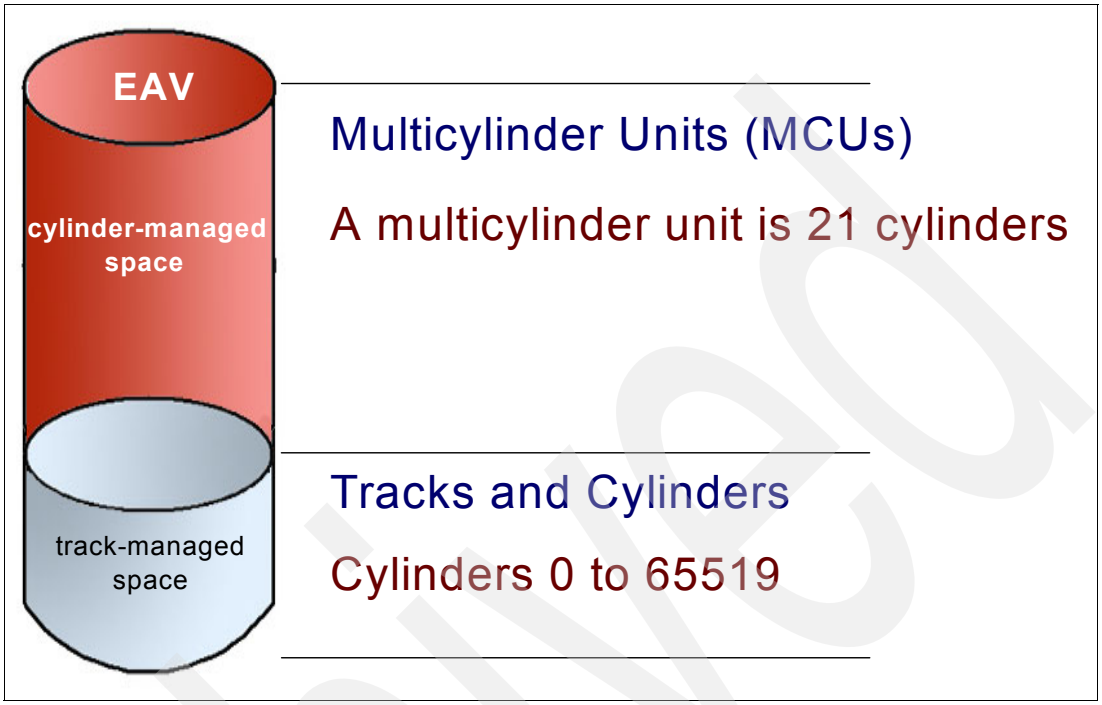


Figure 4-7 EAV and multicylinder units

The 21-cylinder value for the MCU is derived from being the smallest unit that can map out the largest possible EAV volume and stay within the index architecture (with a block size of 8192 bytes). It has the following characteristics:

- It is a value that divides evenly into the 1 GB storage segments of an IBM System Storage DS8000 subsystem.
- These 1 GB segments are the default allocation unit of the IBM System Storage DS8000 subsystem and are equivalent to 1113 cylinders.
- These segments are allocated in multiples of 1113 cylinders, starting at cylinder 65,520.

4.2.6 Support for VSAM data sets

Support for VSAM data sets includes both SMS and non-SMS managed data sets. However, the CA sizes need to be compatible and consist of 1, 3, 5, 7, 9, or 15 tracks in order to be EAS eligible.

All other data sets can reside in the base addressing space (BAS), which uses traditional track-managed space like any ordinary non-EAV 3390.

4.2.7 New format DSCBs for EAV volumes

DSCBs are volume table of contents (VTOC) entries that describe data set attributes and allocated extent information. This extent information describes allocated space by using beginning and ending track addresses. These are called *extent descriptors*. These extent descriptors might contain a 28-bit cylinder number for their track addresses. DSCBs also

contain metadata in the Format 1 DSCB that have the characteristics or attributes of the allocated data set. There is no more space available in the Format 1 DSCB to add additional attributes.

Extended attribute DSCBs

There are new DSCB types that provide a method of preventing existing programs from seeing unexpected track addresses (28-bit cylinder numbers) and new format DSCBs. They are described in the following list:

Format 8 DSCB This DSCB is equivalent to a Format 1 DSCB. It contains a chain pointer to a Format 9 DSCB

Format 9 DSCB This DSCB provides attribute data and a list of pointers to each possible Format 3 DSCB. It contains a chain pointer to the possible next Format 9 or Format 3 DSCB. These attributes are maintained only for the first volume. There is only one Format 9 DSCB in z/OS V1.10.

Format 9 DSCB

The Format 9 DSCB is new as of V1.10 and has all the information z/OS needs to record the attributes of a data set in the EAS of an EAV. The Format 9 DSCB can point to one or more Format 3 DSCBs, as shown in Figure 4-8. Thus, we see that the Format 8 DSCB exists strictly as an indication that a Format 9 DSCB with all the EAS information has been inserted between the Format 1 (Format 8) DSCB and the Format 3 DSCB chain.

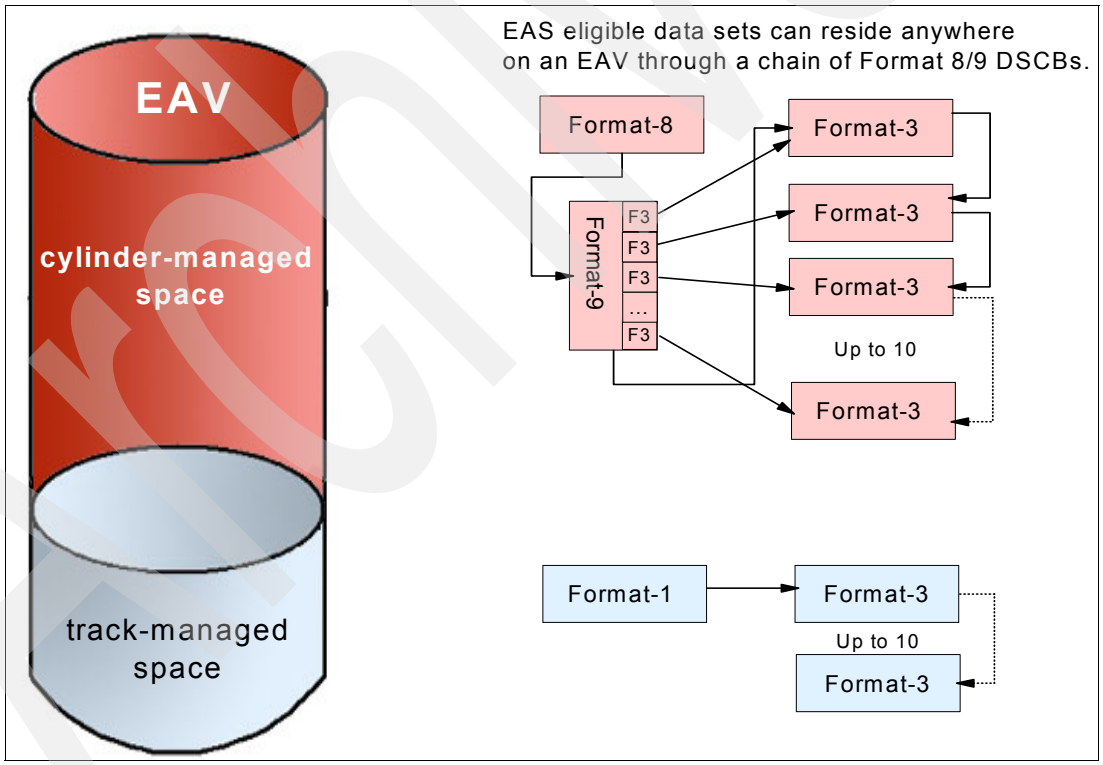


Figure 4-8 Access to EAS eligible data sets on EAV

Note: The logical DSCB chain today for a data set is a Format 1 and up to 10 possible Format 3 DSCBs. The logical DSCB chain for an EAS-eligible data set on an EAV is a Format 8 and one or more Format 9 DSCBs, and up to 10 possible Format 3 DSCBs. In both cases, the chain pointer in each DSCB points to the next one, if one exists.

The Format 9 DSCB is a place for additional attribute information and contains direct pointers to each possible Format 3 DSCB. With this new service in the system (OBTAIN, CVAFDIR), you can read the entire logical DSCB chain for a data set in one call. There are no more loops to read DSCBs until the chain pointer is zero.

z/OS V1.11: Data set attributes are recorded in the Format 9 DSCB. These fields indicate the job and step name and the time that has elapsed since midnight of the date that the data set was created.

In future releases, additional Format 9 DSCBs might be chained between the subtype 1 and any Format 3 DSCBs.

The Format 9 DSCB exists only for EAS-eligible data sets (VSAM support with z/OS V1.10), and extended-format sequential data sets (with z/OS V1.11)), and it contains the following EAV information:

- The format identifier is x'F9'.
- A subtype field.
- In the first EAV release, the subtype is 1.
- In future releases, additional subtypes might be added.
- It can track addresses that point directly to up to 10 Format 3 DSCBs.
- All the Format 3 DSCBs can be read with one channel program.
- A 20-byte field, DS9ATRV1, that IBM is reserving for vendors. IBM will not specify or monitor its content.
- A "next DSCB" address points to a possible Format 3 DSCB. The Format 3 DSCBs continue to be chained.

Format 4 DSCB

The Format 4 DSCB contains the following new information for EAV volumes:

- The number of cylinders on a volume.
- The existing 2-byte field DS4DSCYL with the Format 4 DSCB contains the value of x'FFFE' (65534). This identifies the volume as an EAV.
- DS4EAV is defined as a constant value of X'FFFE'.
- A new 4-byte field DS4DCYL contains the number of cylinders on the volume.
- A new allocation unit for cylinders above 65520.
- A new 2-byte field DS4LCYL contains a code value of x'0010' to indicate that the cylinder-managed space after the first 65,520 cylinders must be allocated in units that are larger than one cylinder.
- This value represents 65,520 cylinders divided by 4095. For a non-EAV, this will be zero (0).
- The new DS4MCU field (minimum allocation unit) contains the number of cylinders of which each extent in the cylinder-managed area must be a multiple. For an EAV, this is 21. For a non-EAV, this will be zero (0). It is valid only when the value in DS4DSCYL is DS4EAV.

Format 8 DSCB

This DSCB is identical to the Format 1 DSCB with the following exceptions:

- The format identifier (DS1FMTID) is x'F8' instead of x'F1'. The new symbols that are defined are:
  - A DS1IDC constant value of X'F1' in DS1FMTID.
  - A DS8IDC constant value of X'F8' in DS1FMTID.
- Track addresses in the extent descriptors (start with DS1EXT1) use a new track address format (that is, they may contain 28-bit cylinder numbers).
- The "next DSCB" address (DS1PTRDS) always points to a Format 9 DSCB (a new type of DSCB) instead of a possible first Format 3 DSCB.

Accessing extended attribute DSCBs

To access extended attribute DSCBs, the system requires the specification of a new permission keyword on system services that read DSCBs. By specifying the EADSCB=OK keyword, the invoking program indicates to the system service that it understands extended attribute DSCBs and the 28-bit cylinder numbers that could be present in the data set's extent descriptors.

Programs that do not used EADSCB=OK, that is, are not permitted to access these extended attribute DSCBs, will be abended and `msgIEC142I 113-48` is issued if they attempt to open a VTOC or data set that contains or uses DSCB Format 8 or Format 9.

EADSCB=OK keyword for CVAF, DADSM, and DCBE macros

This keyword specifies whether a program supports data sets with Format 8 and Format 9 DSCBs. Such data sets can appear on Extended Address Volumes.

Attention: Use EADSCB=NOTOK when your program does not support data sets that have Format 8 and Format 9 DSCBs. The extent descriptors in DSCBs for a data set described with these formats may have track addresses that contain cylinder addresses of 65,520 or larger. EADSCB=OK is accepted for data sets described by all DSCB types, including Format 1 DSCBs, regardless of the volume size where the data set resides. Your program can also run on an older level of the system that does not support this keyword. In these cases, EADSCB=OK is ignored. EADSCB=OK sets byte 2 bit 4 in the OBTAIN parameter list to on.

The EADSCB=OK keyword has been added to the following services:

- DADSM macros
  - OBTAIN
  - CAMLST (can be used to generate a parmlist for OBTAIN)
- CVAF macros
  - CVAFDIR
  - CVAFFILT
  - CVAFDSM
  - CVAFSEQ
- Access method macro

  DCBE (to be used with OPEN for a VTOC or VSAM data set with EXCP access)

Attention: Not specifying the EADSCB=OK keyword causes these services to fail if issued to a data set that supports extended attribute DSCBs or a volume that supports cylinder-managed space (CVAFDSM and OPEN). Set this keyword when an application supports EADSCB=OK. Specify it on all invocations of each service, regardless of whether the application runs on pre-z/OS V1.10 systems or accesses volumes that do not support extended attribute DSCBs. Macros on earlier releases do not recognize EADSCB=OK, but can be assembled on z/OS V1.10 and above systems and run on downlevel releases, where EADSCB=OK has no effect.

Specifying EADSCB=OK indicates that the program calling these macros understands 28-bit cylinder numbers and Format 8 and Format 9 DSCBs.

Format 9 DSCB vendor fields

Prior to z/OS V1.10, some vendor products used fields in the Format 1 DSCB that IBM reserved for IBM use.

A Format 9 DSCB is mapped by the IECSDSL1 macro, and it contains a 20-byte field DS9ATRV1 that IBM is reserving for vendors. IBM makes rules about DSCB updates, but does not enforce content rules. For this field, use the format shown in Figure 4-9.

```
Subfields that begin with the following two-byte header:
  +0       Flags.
     xxxx ....   Reserved
     .... xxxx   Number of bytes following two-byte header
    +1       Vendor code.  IBM will maintain a vendor code list
             specific for this field.
```

Figure 4-9 Format 9 DSCB vendor fields

Note: For information about VTOC usage and the VTOC DSCBs, see *z/OS DFSMSdfp Advanced Services*, SC35-0428.

4.3 EAV z/OS V1.11 enhancements

The following section describes the EAV R2 enhancements that are provided in z/OS V1.11 that add to the EAV R1 features provided in z/OS V1.10. Further EAV function and capability will be rolled out over a number of subsequent z/OS releases.

4.3.1 EAV maximum size

The maximum EAV size remains at 262,668 cylinders or 223 GB, as delivered in EAV R1 in z/OS V1.10. The maximum size of an EAV might be expanded in subsequent releases.

4.3.2 Support for non-VSAM data set types

Support was added so that non-VSAM data sets are eligible to reside in EAS (beyond cylinder 65519) on an EAV. However, this support has not been enabled, so these data sets cannot reside in the EAS except for extended-format sequential data sets (which by definition are SMS managed).

Support has been added but not enabled for (whether SMS or non-SMS managed):

- PDS
- PDSE
- Basic and large sequential
- Extended format sequential
- BDAM data sets

Support for non-VSAM data set types has only been enabled for extended-format sequential data sets.

4.3.3 EATTR data set level attribute

The same requirements that exist for VSAM data sets also exist for non-VSAM data sets, that is, an EAS eligible data set uses Format 8 and Format 9 DSCBs. Having Format 8 and Format 9 DSCBs implies having extents referencing 28-bit cylinder numbers.

In z/OS V1.10, customers controlled whether VSAM data sets could reside in the EAS by including or excluding EAVs in particular storage groups. For non-SMS managed data sets, the customer controlled the allocation to a volume by specifying a specific VOLSER or esoteric name.

This situation is different in z/OS V1.11 because V1.11 provides a new EATTR data set attribute for controlling the allocation of VSAM and non-VSAM data sets in regard to when extended attribute DSCBs and EAS can be used.

This new data set is used in AMS DEFINE and ALLOCATE, JCL, dynamic allocation, and data class. IDCAMS AAMS ALTER will modify this attribute in existing data sets. There is no JCL override of the current EATTR value for existing data sets. To change the EATTR value after a new allocation requires you to run the AMS ALTER command.

The precedence order of EATTR is JCL, LIKE/REFDD, and Data Class. If there is no data class, or the data class has no EATTR value, the processing defaults to the default value based on data set type.

The EATTR value resides in the Format 1 and Format 8 DSCBs for VSAM and non-VSAM data sets. It also resides in the primary VVR for VSAM data sets so that it is associated with all components of the VSAM data set.

Subparameter definitions in JCL

Use the EATTR JCL keyword, EATTR=OPT, to specify that the data set can have extended attribute DSCBs and can optionally reside in EAS:

- EATTR = OPT: Extended attributes are optional. The data set can have extended attributes and reside in EAS. This is the default value for VSAM data sets if EATTR(OPT) is not specified.
- EATTR = NO: No extended attributes. The data set cannot have extended attributes (Format 8 and Format 9 DSCBs) or reside in EAS. This is the default value for non-VSAM data sets. This is equivalent to specifying EATTR=NO on the JCL and is applicable to all data set types.

Figure 4-10 shows an example of setting EATTR = OPT.

```
//DD2 DD       DSNAME=PDS12,DISP=(,KEEP),UNIT=SYSALLDA,
//    VOLUME=SER=25143,SPACE=(CYL,(10000,,100),,CONTIG),
//             EATTR=OPT
```

Figure 4-10 Specifying EATTR in JCL on a DD statement

Attention: Take care when specifying EATTR=OPT; even if the data set type is not EAS eligible, it will still be accepted and kept in the Format 1 DSCB. At some later time, when support is added through a HMIGRATE/HRECALL or a DSS RESTORE, this data set might end up in EAS on an EAV. This might mean other systems in a sysplex that are downlevel will no longer be able to access the data set.

The EATTR parameter usage is not checked during JCL conversion or interpretation to see if it is being used correctly for an EAS eligible data set type. Figure 4-11 shows a PDS with EATTR=OPT. The EATTR specification is stored in the Format 1 DSCB and propagated through DFSMSdss backup and restore and DFSMShsm migrate and recall process. No attempt will be made during allocation to place it in EAS, as it is not EAS eligible, but at some later time if it does become EAS eligible, it might migrate to EAS through restore or recall actions.

```
 SYSTEMS SUPPORT UTILITIES---IEHLIST
DATE: 2009.226  TIME: 13.12.05
               CONTENTS OF VTOC ON VOL SBOXFH  <THIS VOLUME IS NOT SMS MANAGED>
---------------DATA SET NAME----------------  SER NO  SEQNO  DATE.CRE  DATE.EXP DATE.REF  EXT DSORG RECFM OPTCD
CVERNON.JCL.CNTL                             SBOXFH      1  2009.218    00.000 2009.226    1  PO   FB     00
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST BLK(T-R-L)  DIR.REM  F2 OR F3(C-H-R)  DSC
  R          80            CYLS              0                        2   4 53142                            5
EATTR
OPT
            EXTENTS  NO  LOW(C-H)   HIGH(C-H)
                       0  1255  0     1275 14
                                ----ON THE ABOVE DATA SET,THERE ARE    20 EMPTY CYLINDER(S) PLUS 12 EMPTY TRACK(
```

Figure 4-11 LISTVTOC showing PDS with EATTR=OPT

4.3.4 Other methods for EATTR specification

Different BCP components have to be adapted by using the EATTR attribute introduced with z/OS V1.11. This attribute can be specified by using any of the following methods:

- AMS DEFINE CLUSTER and ALLOCATE

  Using the ALLOCATE or DEFINE CLUSTER commands, you can allocate new data sets by using the EATTR(OPT) parameter to specify that the data set can have extended attribute DSCBs (Format 8 and Format 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 cannot have extended attribute DSCBs; optionally, it can reside in EAS. This is equivalent to specifying EATTR(NO) for the command.

- ISMF data class panel

  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.

Note: The EATTR specification is recorded in the Format 1 or Format 8 DSCBs for all data set types and volume types and is recorded in the VVDS for VSAM cluster names. EATTR is listed by IEHLIST, ISPF, ISMF, LISTCAT, and the catalog search interface (CSI).

- TSO/E support

  TSO/E support for data sets that reside in the extended addressing space, including those that reside in EAS in the ALLOCATE, TRANSMIT, RECEIVE, SUBMIT, RACONVRT, and LISTDSI commands. This support is provided to help ease space constraints by allowing you to move additional data to larger volumes and help simplify storage management by allowing you to store data on a smaller number of volumes.

- Language Environment®

  The Language Environment XL C/C++ Runtime Library supports processing extended-format sequential data sets that have extended attributes, including those data sets that reside in the extended addressing space.

- RACF support

  There is RACF support for discrete profiles for non-VSAM data sets that have extended attributes, including those data sets that reside in the extended addressing space, in addition to the support available now for VSAM data sets.

  - The ADDSD and ALTDSD commands have been updated to support data sets in the EAS.
  - The RACF database is not supported if it allocated in the EAS.
  - RACF utilities do not support allocation of database data sets in the EAS. EATTR=NO (default) should be used on allocations for the following data sets:
    - IRRUT200 SYSUT1 DD
    - IRRMIN00 SYSRACF DD
    - IRRUT400 OUTDD DD

- Communications Server

  z/OS Communications Server FTP supports reading from and writing to sequential extended-format data sets in the extended addressing space (EAS) of an Extended Address Volume (EAV) and the QDISK command.

- Binder support

  There is binder support for input and SYSPRINT data sets that have extended attributes, including those that reside in EAS.

- ISPF EATTR specification

  There is support in ISPF for displaying and setting the EATTR attribute for data sets that can reside on Extended Address Volumes (EAVs). ISPF has added support for the display of the Extended Address Volumes (EAV) data set level attribute EATTR by using option 3.2 on the allocation panel, as shown in Figure 4-12.

```
  Menu  RefList  Utilities  Help
 ------------------------------------------------------------------------
                          Allocate New Data Set
Command ===> ____________________________________________________________
                                                                More:     +
Data Set Name  . . . : ROGERS.TEST.EAV2

Management class . . . MCDB22__        (Blank for default management class)
Storage class  . . . . EAVGK___        (Blank for default storage class)
 Volume serial . . . . GKDD65          (Blank for system default volume) **
 Device type . . . . . ________        (Generic unit or device address) **
Data class . . . . . . EAVGK___        (Blank for default data class)
 Space units . . . . . CYLS _____      (BLKS, TRKS, CYLS, KB, MB, BYTES
                                        or RECORDS)
 Average record unit   _                (M, K, or U)
 Primary quantity  . . 30____________  (In above units)
 Secondary quantity    24____________  (In above units)
 Directory blocks  . . 0_____          (Zero for sequential data set) *
 Record format . . . . FB____
 Record length . . . . 80____
 Block size  . . . . . _____
 Data set name type    ________        (LIBRARY, HFS, PDS, LARGE, BASIC, *
                                        EXTREQ, EXTPREF or blank)
 Extended Attributes   OPT             (NO, OPT or blank)
 Expiration date . . . _________       (YY/MM/DD, YYYY/MM/DD
Enter "/" to select option              YY.DDD, YYYY.DDD in Julian form
_  Allocate Multiple Volumes            DDDD for retention period in days
                                        or blank)
```

Figure 4-12 ISPF Option 3.2 allocation using EATTR

- Service aids enhancements

  As discussed in Chapter 10, “Migration health checks”, of *z/OS Version1 Release11 Implementation,* SG24-7729, service aids have been enhanced for z/OS V1.11, such as:

  – AMASPZAP support for SYSIN and SYSPRINT data sets that reside in the extended addressing space, including those that reside in EAS.

  – SVC dump support for dump data sets that reside in the extended addressing space, including SYSMDUMPs and transaction dumps. Also, IPCS, SNAP, and ABDUMP services support is planned for the placement of dump data sets having extended attributes, including those that reside in EAS.

4.3.5 Determining the specified EATTR value

The specified EATTR value is determined in the JCL process by merging EATTR from the JCL, LIKE=, or data class, in that order.

Note: The EATTR value recorded for a data set type that is not supported as EAS-eligible will have no effect until that system supports that data set type for EAS.

Volume selection and EATTR

Volume selection is based on EATTR values:

- SMS volume selection
- DFSMShsm non-SMS volume selection

Programs can read DSCBs by issuing either OBTAIN, CVAFDIR, or CVAFFILT macros or by reading a VTOC. In a Format 1 or Format 8 DSCB, the EATTR value is recorded in two bits at offset 61, as shown in Figure 4-13.

```
DS1FLAG1
.... ..00  EATTR not specified.
.... ..01  DS1EATTR_NO   EATTR=NO.
.... ..10  DS1EATTR_OPT  EATTR=OPT.
```

Figure 4-13 EATTR in a Format 1 or Format 8 DSCB

4.3.6 IEHLIST LISTVTOC support

The IEHLIST LISTVTOC report has changed, as shown in Figure 4-14. It now shows the EATTR specification saved in the DSCB for a data set.

```
 CONTENTS OF VTOC ON VOL 1P9802  <THIS IS AN SMS MANAGED VOLUME>
 FORMAT 4 DSCB NO AVAIL/MAX DSCB /MAX DIRECT  NO AVAIL  NEXT ALT   FORMAT 6  LAST FMT 1     VTOC EXTENT       THIS DSCB
           VI   DSCBS   PER TRK  BLK PER TRK  ALT TRK   TRK(C-H)   (C-H-R)   DSCB(C-H-R)/LOW(C-H) HIGH(C-H)     (C-H-R)
           81   65499       50         45         0         0  0          1279  14  50    0   1  1279  14     0   1   1
          NUMBER OF   MULTICYLINDER UNITS
          CYLINDERS   FIRST CYL ADDR  SPACE
             262668         65520      21
---------------DATA SET NAME----------------  SER NO  SEQNO  DATE.CRE  DATE.EXP  DATE.REF  EXT DSORG RECFM OPTCD BLKSIZE
BRS8AM02.HANDLIN.V1P9802.NVSAM.FILL.CB1       1P9802      1  2008.176    00.000    00.000    1  PS   FB     00     6320
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST BLK(T-R-L)  DIR.REM  F2 OR F3(C-H-R)  DSCB(C-H-R)
S           80           TRKS              1                      0   0 58786                                0   2  29
            EXTENTS  NO  LOW(C-H)   HIGH(C-H)
                      0 50901  9    50901 14
                                 ----ON THE ABOVE DATA SET,THERE ARE         6 EMPTY TRACK(S)
---------------DATA SET NAME----------------  SER NO  SEQNO  DATE.CRE  DATE.EXP  DATE.REF  EXT DSORG RECFM OPTCD BLKSIZE
BRS8AM02.HANDLIN.V1P9802.VSAM.FILL.D1.DATA    1P9802      1  2008.176    00.000    00.000    1  VS    U      80     4096
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST BLK(T-R-L)  DIR.REM  PTR TO F3(C-H-R)  DSCB(C-H-R
S            0           CYLS              0                                                                 2   3  13
                                                VENDOR
EATTR   JOB        STEP       CREATE TIME      CODE  DATA
OPT     JOB1GALA   STEP0001  01:28:58.673275     1  X'11223344'    2  X'556677'  255  X'88'
EXTENTS  NO       LOW(C-H)      HIGH(C-H)
          0     193704  0      193724 14
                                 ----UNABLE TO CALCULATE EMPTY SPACE.
VPSM                 A = NUMBER OF TRKS IN ADDITION TO FULL CYLS IN THE EXTENT
            TRK       FULL              TRK        FULL              TRK       FULL              TRK       FULL
           ADDR       CYLS  A          ADDR        CYLS  A          ADDR       CYLS  A          ADDR       CYLS  A
        3784725         42  0       3785670          63  0       3786930         21  0       3787560         42  0
        3788505         63  0       3789765          21  0       3790395         42  0       3791340         63  0

 THERE ARE     52290 EMPTY CYLINDERS PLUS  4590 EMPTY TRACKS ON THIS VOLUME
 THERE ARE 12852 EMPTY CYLINDERS PLUS  4590  EMPTY TRACKS FROM THE TRACK-MANAGED SPACE
 THERE ARE  957221 BLANK DSCBS IN THE VTOC ON THIS VOLUME
New fields and extent descriptors for data and free space adjusted to support larger cylinder numbers
```

Figure 4-14 IEHLIST LISTVTOC report

You should update programs or procedures that depend on the LISTVTOC output. The report has the following changes:

- For data sets that support extended attribute DSCBs, the extent descriptions are adjusted to support larger cylinder numbers and show its EATTR specification.
- EATTR values can be:
  - NS: Not specified (when a blank is specified in the extended attribute field in ISPF Option 3.2 in the Allocation of a new data set panel, as shown in Figure 4-12 on page 100).
  - NO: the data set is not allowed to reside in EAS.
  - OPT: The data set can optionally reside in EAS.
- For volumes that support cylinder-managed space, the Format 4 DSCB display identifies the size of the volume, the location of the address of the cylinder where cylinder-managed space begins, and the allocation size of multiple cylinder unit that the system uses to allocate space in cylinder-managed space.
- At the end of the report, there is a listing of the number of empty cylinders and additional empty tracks for the track-managed space of volumes that support cylinder-managed space.
- In the VTOC Pack Space Map[1] (VMSM) report of free extents, the TRK ADDR and FULL CYLS columns are adjusted to support larger numbers for the track address and full cylinders when the volume is an EAV.

Note: For an explanation of IEHLIST VTOC report fields, see *z/OS DFSMSdfp Utilities*, SC26-7414.

FORMAT option

With the FORMAT option for data sets that support extended attribute DSCBs, the extent descriptions are adjusted to support larger cylinder numbers. With the DUMP option, Format 9 DSCBs are displayed.

4.3.7 LISTCAT support

LISTCAT reports have been enhanced to reflect the presence of the EATTR attribute. LISTCAT only shows the EATTR for VSAM data sets, for example, Example 4-1 shows that the EATTR value OPT was specified for a KSDS. Note that this VSAM KSDS starts at cylinder 65,520 track 0 (LOW-CCHH X'FFF00000') and finishes at cylinder 65,541 track 14 (HIGH-CCHH X'0004001E').

Example 4-1 EATTR in LISTCAT

```
CLUSTER ------- CVERNON.EAVTEST.KSDS0
    IN-CAT --- UCAT.VSBOX01
    HISTORY
     DATASET-OWNER-----(NULL)     CREATION--------2009.230
     RELEASE----------------2     EXPIRATION------0000.000
    SMSDATA
     STORAGECLASS ------EAVGK     MANAGEMENTCLASS---MCDB22
     DATACLASS ---------EAVGK     LBACKUP ---0000.000.0000
      EATTR--------------(OPT)
     BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
```

[1] The VPSM is a VTOC Index record; details can be found in *z/OS DFSMSdfp Advanced Services,* SC26-7400

```
      BWO---------------(NULL)
     RLSDATA
       LOG ----------------(NULL)   RECOVERY REQUIRED --(NO)     FRLOG --
      VSAM QUIESCED -------(NO)    RLS IN USE ---------(NO)
      LOGSTREAMID-----------------------------(NULL)
      RECOVERY TIMESTAMP LOCAL-----X'0000000000000000'
      RECOVERY TIMESTAMP GMT-------X'0000000000000000'
    PROTECTION-PSWD-----(NULL)    RACF----------------(NO)
     ASSOCIATIONS
      DATA-----CVERNON.EAVTEST.KSDS0.DATA
      INDEX----CVERNON.EAVTEST.KSDS0.INDEX
DATA----------CVERNON.EAVTEST.KSDS0.DATA IN-CAT---UCAT.VSBOX01
   HISTORY
   DATASET-OWNER-----(NULL)   CREATION--------2009.230
   RELEASE----------------2   EXPIRATION------0000.000
   ACCOUNT-INFO-----------------------------------(NULL)
  PROTECTION-PSWD-----(NULL)   RACF----------------(NO)
   ASSOCIATIONS
    CLUSTER--CVERNON.EAVTEST.KSDS0
   ATTRIBUTES
       KEYLEN----------------32     AVGLRECL-------------128     BUFSPACE
       RKP-------------------0      MAXLRECL-------------128     EXCPEXIT
    STRIPE-COUNT-----------1
       SHROPTNS(1,3)   RECOVERY     UNIQUE          NOERASE     INDEXED
    UNORDERED    NOREUSE   NONSPANNED   EXTENDED   EXT-ADDR
   STATISTICS
       REC-TOTAL--------------0     SPLITS-CI--------------0     EXCPS---
       REC-DELETED------------0     SPLITS-CA--------------0     EXTENTS-
       REC-INSERTED-----------0     FREESPACE-%CI---------30     SYSTEM-T
       REC-UPDATED------------0     FREESPACE-%CA---------10          X'0
   REC-RETRIEVED----------0   FREESPC---------16773120
    ALLOCATION SPACE-TYPE------CYLINDER     HI-A-RBA--------16773120
       SPACE-PRI-------------21     HI-U-RBA---------------0
       SPACE-SEC-------------21
     VOLUME
       VOLSER------------GKDD65     PHYREC-SIZE--------26624     HI-A-RBA
       DEVTYPE------X'3010200F'     PHYRECS/TRK------------2     HI-U-RBA
       VOLFLAG-----------PRIME      TRACKS/CA-------------15
       EXTENTS:
       LOW-CCHH-----X'FFF00000'     LOW-RBA----------------0     TRACKS--
       HIGH-CCHH----X'0004001E'     HIGH-RBA--------16773119
INDEX ---------- CVERNON.EAVTEST.KSDS0.INDEX
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.230
       RELEASE-------
```

An IEHLIST LISTVTOC shows the EATTR value for a non-VSAM data set (Figure 4-15), but a LISTCAT report (Figure 4-16) will not show the value.

```
 SYSTEMS SUPPORT UTILITIES---IEHLIST                              PAGE    1
DATE: 2009.230  TIME: 13.27.51
              CONTENTS OF VTOC ON VOL GKDD65  <THIS IS AN SMS MANAGED VOLUME>
   THERE IS A  1 LEVEL VTOC INDEX
   DATA SETS ARE LISTED IN ALPHANUMERIC ORDER
 FORMAT 4 DSCB NO AVAIL/MAX DSCB /MAX DIRECT  NO AVAIL  NEXT ALT   FORMAT 6  LAST FMT 1     VTOC EXTENT
THIS DSCB
          VI   DSCBS   PER TRK  BLK PER TRK  ALT TRK   TRK(C-H)   (C-H-R)   DSCB(C-H-R)/LOW(C-H) HIGH(C-H)
(C-H-R)
          81    8996       50         45         0         0  0            16  14  50    5   0    16  14     5
0   1
         NUMBER OF   MULTICYLINDER UNITS
         CYLINDERS   FIRST CYL ADDR  SPACE
            120204         65520     21
--------------DATA SET NAME----------------  SER NO  SEQNO  DATE.CRE  DATE.EXP  DATE.REF  EXT DSORG RECFM OPTCD
BLKSIZE
CVERNON.EAVTEST.EFSAM1                       GKDD65      1  2009.219    00.000    00.000    1  PS   FB     00
27920
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST BLK(TTTT-R) DIR.REM  F2 OR F3(C-H-R)
DSCB(C-H-R)
S  BE        80            CYLS             21                                                          5
0  17
EATTR   JOB        STEP      CREATE TIME
OPT     CVERNON4   ALXFS     18:08:58.538451
EXTENTS  NO       LOW(C-H)      HIGH(C-H)
         0      65562  0       65582 14
                               ----UNABLE TO CALCULATE EMPTY SPACE.
```

Figure 4-15 EATTR in LISTVTOC for non-VSAM data set

```
NONVSAM ------- CVERNON.EAVTEST.EFSAM1
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)      CREATION--------2009.219
       RELEASE----------------2      EXPIRATION------0000.000
       ACCOUNT-INFO-----------------------------------(NULL)
     SMSDATA
       STORAGECLASS ------EAVGK      MANAGEMENTCLASS---MCDB22
       DATACLASS ---------EAVGK      LBACKUP ---0000.000.0000
     VOLUMES
       VOLSER------------GKDD65      DEVTYPE------X'3010200F'      FSEQN--
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
       STRIPE-COUNT-----------1
       EXTENDED
```

Figure 4-16 EATTR is not displayed for non-VSAM

The EXTENDED and EXT-ADDR attributes are not associated with EAS eligibility. They indicate a data set that is in extended format and, in the case of EXT-ADDR, extended addressability, which means a VSAM data set that can be larger than 4 GB.

The other difference with LISTCAT is that when EATTR is not specified, its value is shown as NULL (Figure 4-17). If the EATTR is left off the DEFINE command or the JCL (if the VSAM dat set was created that way), it becomes NULL and the default for VSAM is EAS eligible, and the VSAM will be allocated in cylinder managed storage if it meets other criteria, such as being larger than the break point value (BPV) and uses multiples of 21 cylinders for space allocation.

```
CLUSTER ------- CVERNON.EAVTEST.KSDS1
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.230
       RELEASE----------------2     EXPIRATION------0000.000
     SMSDATA
       STORAGECLASS ------EAVGK     MANAGEMENTCLASS---MCDB22
       DATACLASS ---------EAVGK     LBACKUP ---0000.000.0000
       EATTR-------------(NULL)
       BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
       BWO---------------(NULL)
```

Figure 4-17 EATTR unspecified shown as NULL

Attention: The EATTR value is not displayed for non-VSAM data sets by LISTCAT.

Displaying the EAVGK EAV Storage Group shows that the GKDD65 EAV volume is enabled on all systems (Figure 4-18).

```
D SMS,SG(EAVGK),LISTVOL
IGD002I 13:29:19 DISPLAY SMS 593

STORGRP  TYPE    SYSTEM= 1 2 3 4
EAVGK    POOL            + + + +

VOLUME   UNIT    SYSTEM= 1 2 3 4                        STORGRP NAME
GKDD65   DD65            + + + +                          EAVGK
**************************** LEGEND ****************************
. THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
+ THE STORAGE GROUP OR VOLUME IS ENABLED
- THE STORAGE GROUP OR VOLUME IS DISABLED
* THE STORAGE GROUP OR VOLUME IS QUIESCED
D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
> THE VOLSER IN UCB IS DIFFERENT FROM THE VOLSER IN CONFIGURATION
SYSTEM  1 = SC63       SYSTEM  2 = SC64       SYSTEM  3 = SC65
SYSTEM  4 = SC70
```

Figure 4-18 Storage group display showing an EAV volume enabled on all systems

4.3.8 EAS eligible data set types

With z/OS V1.11 and z/OS V1.10 EAV support, the following data set types can reside in the EAS of an EAV:

- VSAM KSDS, RRDS, ESDS, and linear data sets, both SMS and non-SMS managed.
- z/FS data sets (which are VSAM anyway)
- Extended format sequential (This is a new addition for z/OS V1.11, and is currently the only non-VSAM type.)

Note: In a future release, some of these data sets might become EAS eligible. All data set types, even those listed here, can be allocated in the track-managed space on a device with cylinder-managed space on an EAV volume.

Eligible EAS data sets can be created and extended anywhere on an EAV. Data sets that are not eligible for EAS processing can only be created or extended in the track-managed portions of the volume.

4.3.9 Non-EAV z/OS V1.11 enhancements

z/OS V1.11 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 rejects 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 overusing the system resources, because an iteration of volume re-selection might consume a lot of system resources when there are a large number of candidate volumes.

4.3.10 EAS ineligible data set types

The following types of data sets are cannot reside in the EAS:

- HFS
- VSAM data sets with IMBED or KEYRANGE
- VSAM data sets with incompatible CA sizes (CA size must be 1, 3, 5, 7, 9, or 15 tracks.)
- VTOC and VTOC index
- Page data sets
- Catalogs
- VVDS
- PDSE

- Basic and large sequential
- PDS
- BDAM

4.3.11 Dynamic Volume Expansion request interface

Dynamic Volume Expansion (DVE) is performed by the IBM System Storage DS8000 Storage Manager and can be requested through its web GUI. 3390 volumes may be increased in size, for example, from a 3390 model 3 to a model 9 or a model 9 to a model A (EAV).
z/OS V1.11 introduces an interface that can be used to make requests for dynamic volume expansion of a 3390 volume on a IBM System Storage DS8000 subsystem from that subsystem.

Note: For the dynamic volume expansion function, volumes cannot be in Copy Services relationships (point-in-time copy, FlashCopy SE, Metro Mirror, Global Mirror, Metro/Global Mirror, or z/OS Global Mirror) during expansion.

There are two methods to dynamically grow a volume:

- Use the DS command-line interface (DSCLI), as discussed in “Command-line interface (DSCLI)” on page 107.
- Use a web browser GUI, as discussed in “Web browser GUI” on page 109.

Note: All systems must be at the z/OS V1.10 level or above for the DVE feature to be used when the systems are sharing the IBM System Storage DS8000 subsystem that has been updated with Release 4.0 Licensed Internal Microcode at a LCU level.

z/OS V1.10 is the minimum level required to share an online EAV.

Command-line interface (DSCLI)

You can use the DS command-line interface (DSCLI), which is an SMI-S industry-standard-compatible API.

The **chckdvol** command changes the name of a count key data (CKD) base volume. It can be used to expand the number of cylinders on an existing volume.

New options

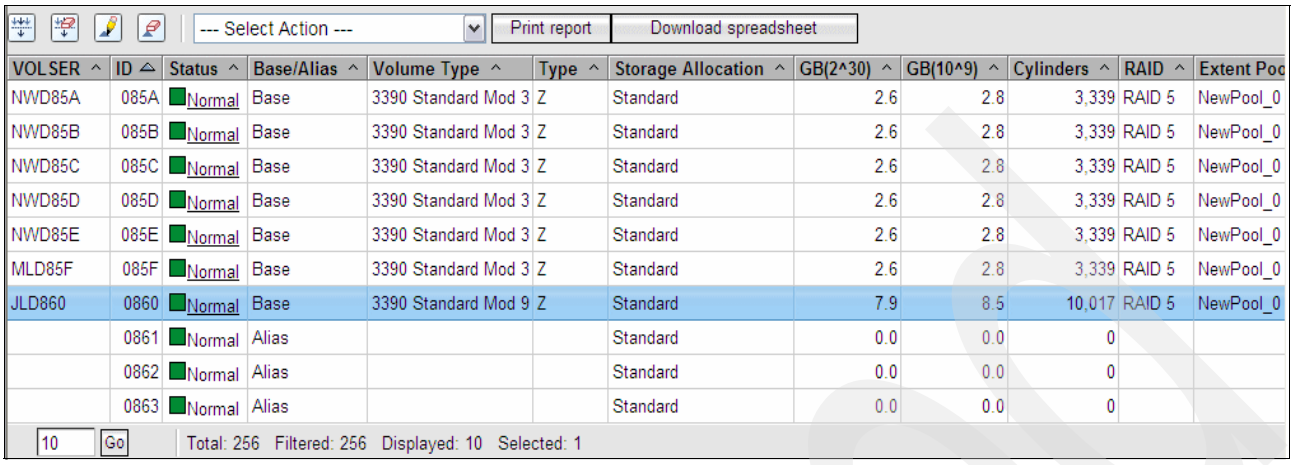
A new parameter for the **chckdvol** command (-cap new_capacity) allows a volume to be increased in size. It is optional and can only be used on a IBM System Storage DS8000 subsystem.

This option specifies the quantity of CKD cylinders that you want allocated to the specified volume. 3380 volumes cannot be expanded. For 3390 Model A volumes (DS8000 only), the -cap parameter value can be in the range of 1 to 65,520 (increments of 1) or 65,667 to 262,668 (increments of 1113). For 3390 volumes, the -cap parameter value can be in the range of 1 to 65,520 (849 KB to 55.68 GB).

Here is an example that shows how to use DSCLI to change the size of a 3390 Mod 3 device, ID 0860 (actual device address D860) to a 3390 Mod 9:

```
chckdvol –cap 10017 0860
```

Figure 4-19 shows changing the device to a 3390 Mod 9, which is displayed using the web browser GUI interface.



VOLSER	ID	Status	Base/Alias	Volume Type	Type	Storage Allocation	GB(2^30)	GB(10^9)	Cylinders	RAID	Extent Poo
NWD85A	085A	Normal	Base	3390 Standard Mod 3	Z	Standard	2.6	2.8	3,339	RAID 5	NewPool_0
NWD85B	085B	Normal	Base	3390 Standard Mod 3	Z	Standard	2.6	2.8	3,339	RAID 5	NewPool_0
NWD85C	085C	Normal	Base	3390 Standard Mod 3	Z	Standard	2.6	2.8	3,339	RAID 5	NewPool_0
NWD85D	085D	Normal	Base	3390 Standard Mod 3	Z	Standard	2.6	2.8	3,339	RAID 5	NewPool_0
NWD85E	085E	Normal	Base	3390 Standard Mod 3	Z	Standard	2.6	2.8	3,339	RAID 5	NewPool_0
MLD85F	085F	Normal	Base	3390 Standard Mod 3	Z	Standard	2.6	2.8	3,339	RAID 5	NewPool_0
JLD860	0860	Normal	Base	3390 Standard Mod 9	Z	Standard	7.9	8.5	10,017	RAID 5	NewPool_0
	0861	Normal	Alias			Standard	0.0	0.0	0		
	0862	Normal	Alias			Standard	0.0	0.0	0		
	0863	Normal	Alias			Standard	0.0	0.0	0		

Figure 4-19 Device change from a 3390 Mod 3 to a 3390 Mod 9

Using the IBM System Storage DS8000 Storage Manager

The 3390 Model A can be defined as a new subsystem or can be defined when a new EAV volume beyond 65,520 cylinders is specified. The number of cylinders when defining an EAV volume can be defined as low as 1 cylinder and up to 262,668 cylinders.

In addition, expanding a 3390 type of volume that is below 65,520 cylinders to be larger than 65,520 cylinders converts the volume from a 3390 standard volume type to a 3390 Model A and produces an expanded address volume.

Web browser GUI

You can use an intuitive GUI to manage the system from any web browser capable of accessing the system management server. The GUI can be used in offline mode to perform simulated configuration modeling. To increase the number of cylinders on an existing volume, you can use the web browser GUI by selecting the volume (shown as VOLSER NWDF64 in Figure 4-20) and then using the Select Action drop-down menu to select **Increase Capacity**.

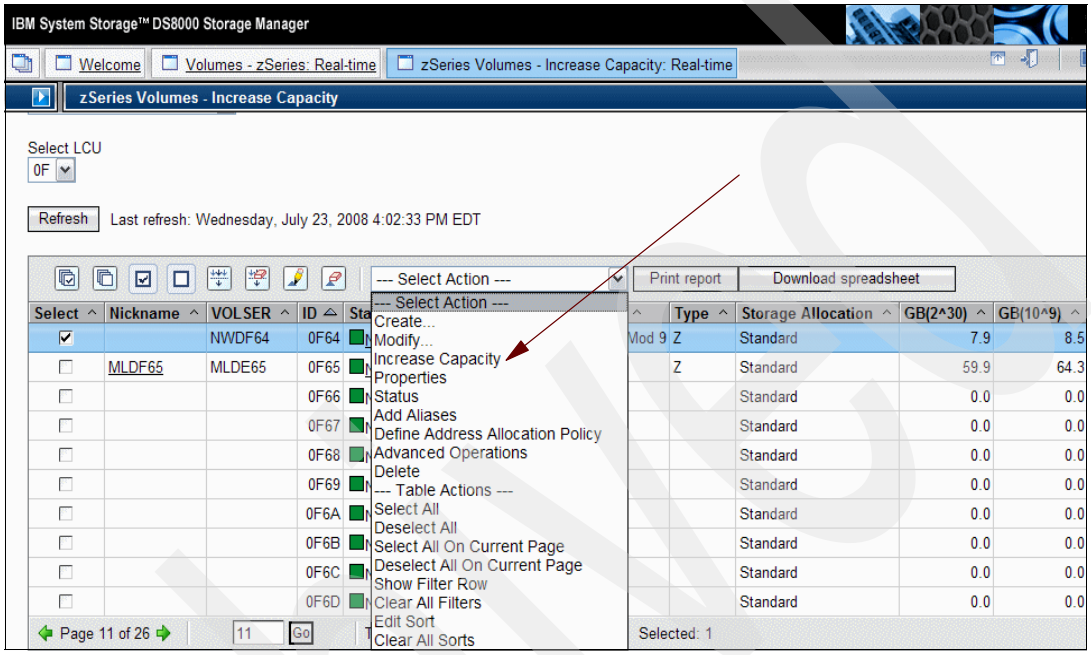


Figure 4-20 Window to increase the size of a volume

The window shown in Figure 4-21 opens after you select **Increase Capacity**. A warning message is issued regarding a possible volume type change to 3390 custom. You can then select **Close Message**. Notice that the window displays the maximum number of cylinders that can be specified.

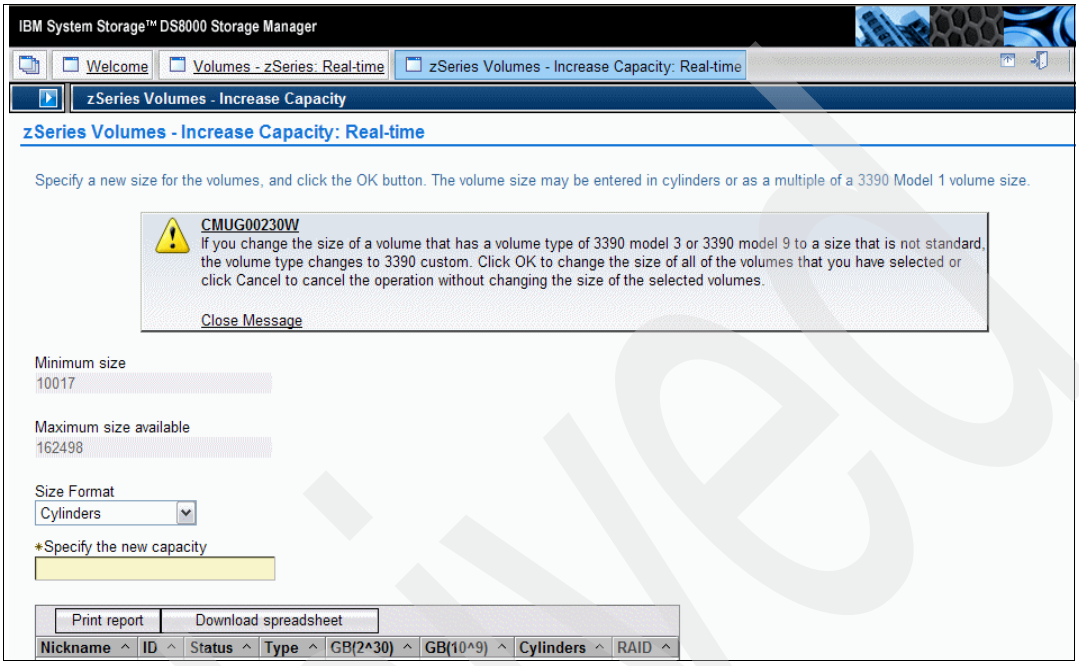


Figure 4-21 Warning message issued regarding the volume type change

Note: Only volumes of type 3390 model 3, 3390 model 9, and 3390 custom can be expanded. The total volume cannot exceed the available capacity of the storage image. Capacity cannot be increased for volumes that are associated with Copy Services functions.

After you close the message, the window shown in Figure 4-22 opens, where you can specify the number of cylinders to increase the volume size. When you specify a new capacity to be applied to the selected volume, specify a value that is between the minimum and maximum size values that are displayed. Maximum values cannot exceed the amount of total storage that is available.

.

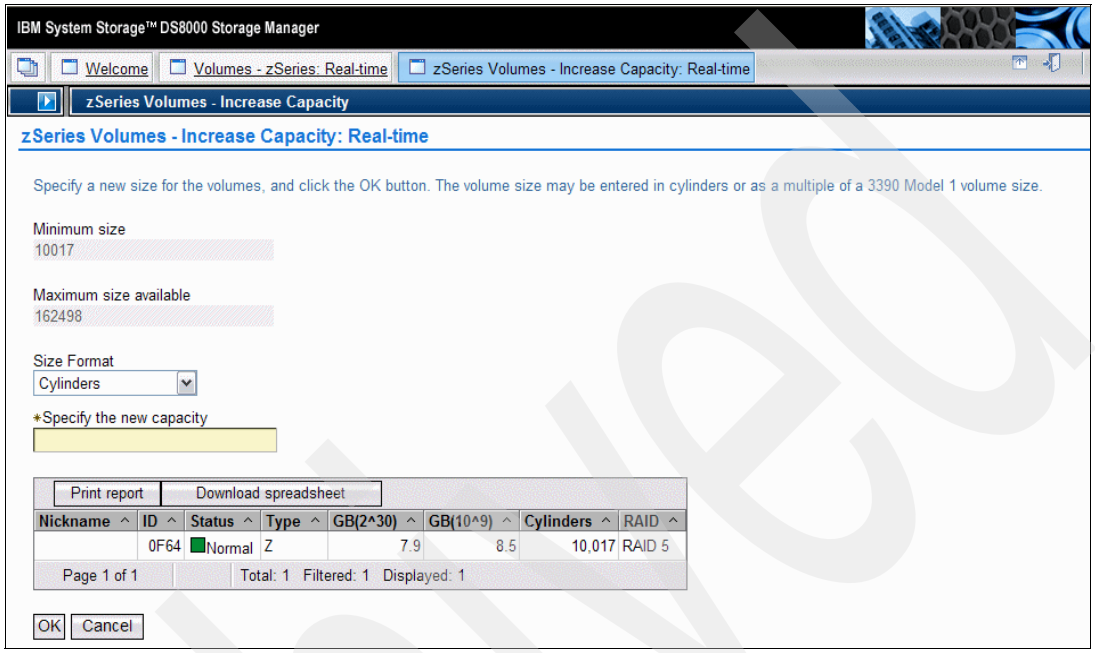


Figure 4-22 Window to specify the new volume capacity

Thus, if 75000 cylinders is specified in the window shown in Figure 4-22, the message displayed in Figure 4-23 is issued when you click **OK**.

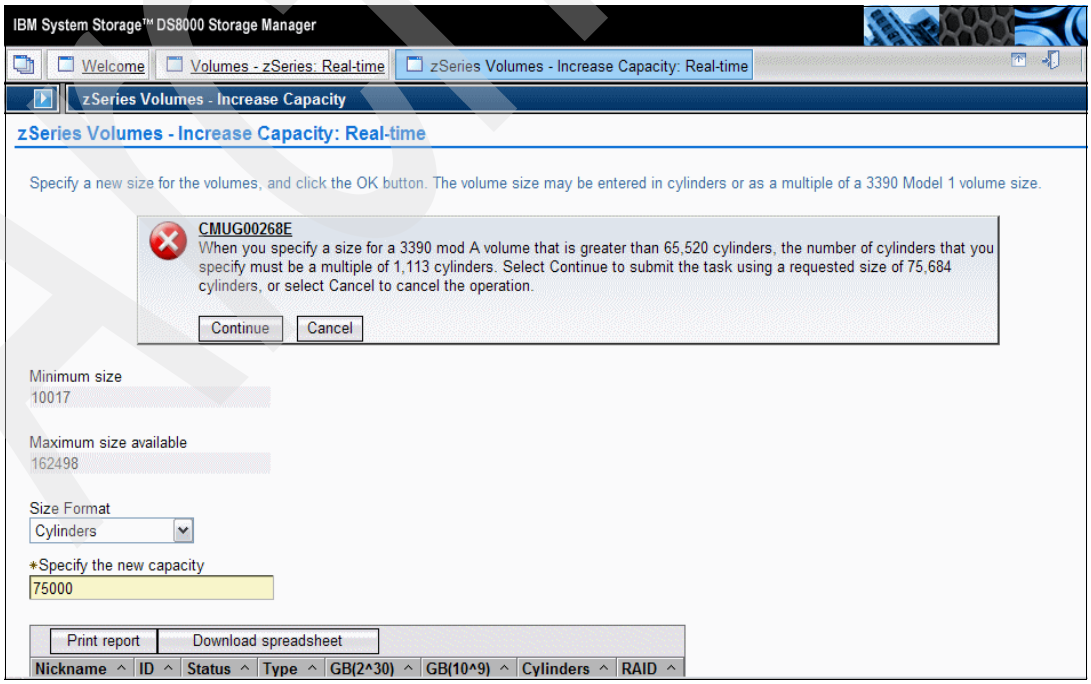


Figure 4-23 Message showing the actual number of cylinders required

Specify **Continue**, and a requested size of 75,684 is processed for the expansion of the volume.

Note: Remember that the number of cylinders must be as stated earlier. The reason an MCU value is 21 cylinders is because it is derived from being the smallest unit that can map out the largest possible EAV and stay within the index architecture (with a block size of 8192 bytes). It is also a value that divides evenly into the 1 GB storage segments of a IBM System Storage DS8000 subsystem. These 1 GB segments are the allocation unit of the IBM System Storage DS8000 subsystem and are equivalent to 1113 cylinders. Data sets allocated in cylinder-managed space might have their requested space quantity rounded up to the next MCU.

4.3.12 Automatic VTOC and index reformat after DVE

When a volume is dynamically expanded, the VTOC and VTOC index has to be reformatted to map the additional space. At z/OS V1.10 and earlier, this has to be done manually by the system programmer or storage administrator by submitting an ICKDSF REFVTOC job.

With z/OS V1.11, when a volume is increased in size, this is detected by the system, which does an automatic VTOC and index rebuild. The system is informed by state change interrupts (SCIs), which is controlled with new DEVSUPxx parmlib member options as follows:

REFVTOC=ENABLE Enables the automatic REFVTOC function of the device manager. With the REFVTOC function enabled, when a volume expansion is detected, the Device Manager causes the volume VTOC to be rebuilt. This allows the newly added space on the volume to be used by the system.

REFVTOC=DISABLE This is the default value. It disables the automatic REFVTOC function of the device manager. With the REFVTOC function disabled, when a volume expansion is detected, the following message is issued:

```
IEA019I dev, volser, VOLUME CAPACITY
CHANGE,OLD=xxxxxxxx,NEW=yyyyyyyy
```

If you use REFVTOC=DISABLE, the VTOC is not rebuilt. An ICKDSF batch job must be submitted to rebuild the VTOC before the newly added space on the volume can be used, as is the case for z/OS V1.10. Invoke ICKDSF with REFORMAT/REFVTOC to update the VTOC and index to reflect the real device capacity.

Note: The refresh of the index occurs under the protection of an exclusive SYSTEMS ENQ macro for major name SYSZDMO, minor name DMO.REFVTOC.VOLSER.volser.

You can use the SET DEVSUP=xx command to enable automatic VTOC and index reformatting after loading an initial program or disabling.

You can also use F DEVMAN,ENABLE(REFVTOC) after loading the initial program to enable automatic VTOC and index reformatting. You should update the DEVSUPxx parmlib member to ensure it remains enabled across subsequent loading of initial programs.

4.3.13 Straddled extents: Differences between V1.10 and V1.11

In z/OS V1.10, there is a restriction that a given extent cannot start in track-managed space and end in cylinder-managed space.

In z/OS V1.11, a given extent is now allowed to start in track-managed space and end in cylinder-managed space. You can only have one straddled extent per EAV. Essentially, the space is still contiguous, but starts at a low CCCCHHHH address, using a 16-bit cylinder number, and ends with a high CCCCcccH address, using a 28-bit cylinder number. An example is shown in Figure 4-24. This example shows the DATA portion of a KSDS that uses a single extent.

z/OS V1.10 can tolerate straddled extents created on z/OS V1.11 in VSAM data sets, but a straddled extent cannot be created on V1.10. So, if a VSAM data set needs to be extended on V1.10, even though there might be a contiguous free space of storage running from track-managed storage through to cylinder-managed storage, on V1.10 it will be two separate extents. In order to get the extent reduction benefits provided at V1.11, you might consider running any update processing from V1.11 and processing that only requires read access from V1.10.

```
DATA----------CVERNON.EAVTEST.KSDS0.DATA IN-CAT---UCAT.VSBOX01
   HISTORY
   DATASET-OWNER-----(NULL)  CREATION--------2009.230
   RELEASE----------------2  EXPIRATION------0000.000
   ACCOUNT-INFO-----------------------------------(NULL)
  PROTECTION-PSWD-----(NULL)  RACF---------------(NO)
   ASSOCIATIONS
    CLUSTER--CVERNON.EAVTEST.KSDS0
   ATTRIBUTES
       KEYLEN----------------32     AVGLRECL-------------128     BUFSPACE
       RKP-------------------0      MAXLRECL-------------128     EXCPEXIT
    STRIPE-COUNT-----------1
       SHROPTNS(1,3)   RECOVERY     UNIQUE           NOERASE     INDEXED
   UNORDERED    NOREUSE   NONSPANNED   EXTENDED   EXT-ADDR
   STATISTICS
       REC-TOTAL--------------0     SPLITS-CI--------------0     EXCPS---
       REC-DELETED------------0     SPLITS-CA--------------0     EXTENTS-
       REC-INSERTED-----------0     FREESPACE-%CI---------30     SYSTEM-T
       REC-UPDATED------------0     FREESPACE-%CA---------10          X'0
   REC-RETRIEVED----------0  FREESPC---------16773120
    ALLOCATION SPACE-TYPE------CYLINDER    HI-A-RBA--------16773120
       SPACE-PRI-------------21     HI-U-RBA---------------0
       SPACE-SEC-------------21
     VOLUME
       VOLSER------------GKDD65     PHYREC-SIZE--------26624     HI-A-RBA
       DEVTYPE------X'3010200F'     PHYRECS/TRK------------2     HI-U-RBA
       VOLFLAG-----------PRIME      TRACKS/CA-------------15
       EXTENTS:
       LOW-CCHH-----X'FFF00000'     LOW-RBA----------------0     TRACKS--
       HIGH-CCHH----X'0004001E'     HIGH-RBA--------16773119
```

Figure 4-24 A straddled extent starts with a 16-bit cylinder number and finishes with a 28-bit number

Attention: Coexistence support in z/OS V1.10 will be provided to recognize straddled extents in APAR OA26623. However, z/OS V1.10 does not support creating them. A straddled extent can only be created at z/OS V1.11 and later.

If space is fragmented, this might allow space allocation in some cases where it could not have succeeded before. One case is when the user coded CONTIG for large, single contiguous extent. Another case is where six extents would have been required and only five are allowed. If the ALX (all extents) or MXIG (maximum single extent) is coded in the SPACE parameter on V1.11, the same space may be obtained, as in V1.10, but in one less extent, because a straddled extent is now possible.

4.4 EAV migration assistance tracker

The EAV migration assistance tracker is a tool that can help you find most programs that require changes and identify most tasks required to support Extended Address Volumes, as well as help identify areas of concern.

The EAV migration assistance tracker is an extension of the console ID tracking facility provided as part of the operating system and uses an MVS command interface (see II13752 in “APAR II13752” on page 518).

The tracker helps you to:

- Identify select systems services by job and program name, where the invoking programs might require analysis for changes to use new services. The program calls are identified as informational instances for possible migration actions. They are not considered errors, because the services return valid information.
- Identify possible instances of improper use of returned information in programs, such as parsing 28-bit cylinder numbers in an output as 16-bit cylinder numbers. These instances are identified as warnings.
- Identify instances of programs that will either fail or run with an informational message if they run on an EAV. These are identified as programs in error. The migration assistance tracker flags programs when the target volume of the operation is non-EAV, and the function invoked did not specify the EADSCB=OK keyword.

Note: If an items is listed in the tracker report, this does not imply that there is a problem. The report is simply a way to help you determine what to examine and change. For example, instances of programs are not in error when the data set is not on an EAV, but if they were to become EAS eligible, then they would be in error.

Error detection by the tracker

Each category of errors is explained here:

- Identify interfaces with access to the VTOC or EAS eligible data sets that should be upgraded to have EADSCB=OK specified for the following functions:
  - OBTAIN, CVAFDIR, CVAFDSM, CVAFVSM, CVAFSEQ, CVAFFILT, OPEN VTOC, and OPEN of an EAS eligible data set.
- Identify programs that might want to use new services as informational messages.

- Identify the possible improper use of returned information, like parsing 28-bit cylinder numbers in output as 16-bit cylinder numbers as warning messages for the following commands and functions:
  - IEHLIST LISTVTOC, IDCAMS LISTCAT, and IDCAMS LISTDATA PINNED
  - LSPACE, DEVTYPE, and IDCAMS DCOLLECT

4.4.1 General information about the tracker

The tracker function allows component code to register tracking information as a text string of its choosing (up to 28 characters in length). The tracker records this string as a unique instance and appends additional information to it, such as job name, program name, and count of occurrences, to avoid duplicates.

The tracker allows an exclusion list, via a SYS1.PARMLIB member, to prevent an instance from being recorded. Instances already verified would then not show up in the tracker. The exclusion list filters are the registered tracking information, job name, and program name. Wild-carding (? and *) is allowed for these fields in the exclusion list.

In addition to these services, the tracker allows you, 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 send instances of programs recorded in the tracker to IBM.

4.4.2 Migration assistance tracker commands

The tracking facility can be manipulated through the following commands:

- The SETCON command is used to activate and deactivate the Console ID Tracking facility.
- The DISPLAY OPDATA,TRACKING command is used to display the current status of the console ID tracking facility, along with any recorded instances of violations. A sample output of this command is shown in Figure 4-28 on page 118.

CNIDTRxx parmlib member

The CNIDTRxx parmlib member is used to list violations that have already been identified to prevent them from being recorded again.

An optional CNIDTRxx parmlib member can be defined to exclude instances from being recorded. The exclusion list is picked up when the tracker is started or via the SET command. The filters for the exclusion list are the three items that make an instance unique. You can exclude all SMS instances or just LSPACE instances, or exclude all instances created by job names that begin with BRS*. There are many more possibilities. The filters support wildcarding in these three fields.

The recommended exclusion list and the list of DFSMS instances are available at the following website:

`http://www-03.ibm.com/servers/eserver/zseries/zos/downloads/`

Recommended exclusion list on website

Figure 4-25 shows the website exclusion list that you can place into a CNTRIDxx parmlib member.

```
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                             Jobname  Pgmname                              *
* Tracking Information Mask   Mask     Mask     Comments (ignored)          *
*---------------------------+--------+--------+----------------------+
|SMS-I:3 LSPACE*             |*MASTER*|IEE70110| Switch SMF           |
|SMS-I:3 LSPACE*             |ALLOCAS |IEFW21SD| VARY DEVICE OFFLINE  |
|SMS-I:3 LSPACE DATA=*       |*       |ADRDSSU | DSS DATA=            |
|SMS-I:3 LSPACE EXPMSG=*     |*       |ISRUDA  | ISPF EXPMSG=         |
|SMS-I:3 DEVTYPE*            |*       |IS*     | DEVTYPE Calls (ISPF) |
|SMS-I:3 DEVTYPE*            |*       |ADR*    | DEVTYPE Calls (DSS)  |
|SMS-I:3 DEVTYPE*            |*       |ASM*    | DEVTYPE Calls (ASM)  |
|SMS-I:3 DEVTYPE*            |*       |DISKMAP | DEVTYPE DISKMAP calls|
|SMS-I:3 DEVTYPE*            |*       |IEB*    | DEVTYPE Calls (Utili)|
|SMS-I:3 DEVTYPE*            |*       |CEE*    | DEVTYPE Calls (LDAP) |
|SMS-I:3 DEVTYPE*            |*       |EDG*    | DEVTYPE Calls (rmm)  |
|SMS-I:3 DEVTYPE*            |*       |ICH*    | DEVTYPE Calls (Racf) |
|SMS-I:3 DEVTYPE*            |*       |CEL*    | DEVTYPE Calls (LangE)|
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
* To allow DFSMS instances to be recorded remove one or more of the         *
* entries by putting an asterisk in column 1 (or delete them)               *
* and continue tracking.                                                    *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                             Jobname  Pgmname                              *
* Tracking Information Mask   Mask     Mask     Comments (ignored)          *
*---------------------------+--------+--------+----------------------+
|SMS*                        |*       |*       | SMS all instances    |
|SMS*E*                      |*       |*       | SMS Errors           |
|SMS*W*                      |*       |*       | SMS Warnings         |
|SMS*I*                      |*       |*       | SMS Informational    |
|SMS*:1*                     |*       |*       | EAV EADSCB=OK missing|
|SMS*:2*                     |*       |*       | EAV 28-bitcyls output|
|SMS*:3*                     |*       |*       | EAV new function     |
*                                                                           *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                           *
* End of SPECIAL PROCESSING FOR DFSMS                                       *
```

Figure 4-25 Website exclusion list

Tracker exclusion list

The exclusion list prevents instances from being recorded. As instances are identified and handled, you add them to be excluded. To set an exclusion list, use the SET CNIDTR=xx command, as shown in Figure 4-26, using the exclusion list in Figure 4-27. When you display the tracking instances, the active member is identified, as shown in Figure 4-28 on page 118.

```
SET CNIDTR=00
IEE252I MEMBER CNIDTR00 FOUND IN SYS1.PARMLIB
IEE536I CNIDTR   VALUE 00 NOW IN EFFECT
```

Figure 4-26 Setting the tracker exclusion list

```
*----EAV exclusion list------+--------+--------+----------------------+
|SMS-I:3 LSPACE*             |*MASTER*|IEE70110| Switch SMF           |
|SMS-I:3 LSPACE*             |ALLOCAS |IEFW21SD| VARY DEVICE OFFLINE  |
|SMS-I:3 LSPACE DATA=*       |*       |ADRDSSU | DSS DATA=            |
|SMS-I:3 LSPACE EXPMSG=*     |*       |ISRUDA  | ISPF EXPMSG=         |
|SMS-I:3 DEVTYPE*            |*       |IS*     | DEVTYPE Calls (ISPF) |
|SMS-I:3 DEVTYPE*            |*       |ADR*    | DEVTYPE Calls (DSS)  |
|SMS-I:3 DEVTYPE*            |*       |ASM*    | DEVTYPE Calls (ASM)  |
|SMS-I:3 DEVTYPE*            |*       |DISKMAP | DEVTYPE DISKMAP calls|
|SMS-I:3 DEVTYPE*            |*       |IEB*    | DEVTYPE Calls (Utili)|
|SMS-I:3 DEVTYPE*            |*       |CEE*    | DEVTYPE Calls (LE)   |
|SMS-I:3 DEVTYPE*            |*       |EDG*    | DEVTYPE Calls (rmm)  |
|SMS-I:3 DEVTYPE*            |*       |ICH*    | DEVTYPE Calls (Racf) |
|SMS-I:3 DEVTYPE*            |*       |CEL*    | DEVTYPE Calls (LE)   |
|SMS-I:3 DEVTYPE*            |*       |GZD*    | DEVTYPE Calls (zDMF) |
```

Figure 4-27 Sample tracker exclusion list in CNIDTR00 parmlib member

Tracking command example

In Figure 4-28, the tracking information column identifies the instance as being DFSMS-related with the SMS prefix. The additional I, E, or W appended to SMS identifies the instance as being an informational, error, or warning event.

```
D OPDATA,TRACKING
CNZ1001I 14.07.15 TRACKING DISPLAY 412
STATUS=ON       NUM=28   MAX=1000 MEM=00  EXCL=1165  REJECT=0
----TRACKING INFORMATION---- -VALUE-- JOBNAME  PROGNAME+OFF-- ASID NUM
SMS-E:1 DCB OPEN EAS 113-44        01 MHLRES4D ADRDSSU  2410E   2C   1
SMS-I:3 DEVTYPE                    03 *MASTER* ICHSEC06 14598   01   2
SMS-I:3 DEVTYPE                    03 DFHSM70  ARCCTL  24AD7C   76   4
SMS-I:3 DEVTYPE                    03 DFHSM70  ARCCTL  24AD7C   7B   9
SMS-I:3 DEVTYPE                    03 D9C3XXXE IEWBIND 181C6A   21   1
SMS-I:3 DEVTYPE                    03 D9D4MSTR DSNZINIT   1C4   68   5
SMS-I:3 DEVTYPE                    03 GDGTEST  IKJLDI99   1BA   3D  17
SMS-I:3 DEVTYPE                    03 HAIM0    IKJLDI99   1BA   70   2
SMS-I:3 DEVTYPE                    03 HERING   EZAFTPLS  1F24   7A   1
SMS-I:3 DEVTYPE                    03 HERING   INMXM     84F0   6F   3
SMS-I:3 DEVTYPE                    03 MHLRES7  INMRM     D66A   7D   1
SMS-I:3 DEVTYPE                    03 PAUL     IKJLDI99   1BA   7C   4
SMS-I:3 LSPACE DATA=              2003 VTFMSRV  HVTSMS    1716   6D >1K
SMS-W:2 IDCAMS LISTCAT             02 CVERNON  IDCAM01     7A   83   2
SMS-W:2 IDCAMS LISTCAT             02 CVERNON6 IDCAMS    E8A6   21  12
SMS-W:2 IEHLIST LISTVTOC           02 CVERNON# IEHLIST   2B06   21   1
SMS-W:2 IEHLIST LISTVTOC           02 CVERNON# IEHLIST   3908   21   1
SMS-W:2 IEHLIST LISTVTOC           02 CVERNON@ IEHLIST   3908   21   2
SMS-W:2 IEHLIST LISTVTOC           02 CVERNONV IEHLIST   2B06   2C   5
SMS-W:2 IEHLIST LISTVTOC           02 CVERNON1 IEHLIST   3908   2C   1
SMS-W:2 IEHLIST LISTVTOC           02 CVERNON2 IEHLIST   3908   21   1
------------------------------------------------------------------------
TO REPORT THESE INSTANCES, SEND THIS MESSAGE VIA E-MAIL TO
CONSOLES@US.IBM.COM. FOR ADDITIONAL INFORMATION OR TO OBTAIN A CURRENT
EXCLUSION LIST, SEE APAR II13752.
```

Figure 4-28 Tracker instance report output

The remaining text in the tracking information describes the event that was recorded or, for error events, the type of error that would have occurred if the function were executed on an EAV. The tracking value is a value unique to the error being recorded. JOBNAME, PROGRAM+OFF, and ASID identifies what was being run at the time of the instance.

Only unique instances are recorded. Duplicates are kept tracked by the NUM column being incremented. The tracking information, jobname, and program name fields make up a unique instance.

As events are recorded by the tracking facility, report the instance to the product owner. After the event is reported, update the parmlib member so that the instance is no longer recorded by the facility. In this way, the facility only reports new events.

4.4.3 DFSMS tracking instances

In this section, we discuss EAV migration assistant instances that will be recorded. See Appendix C, “Using the extended address volume (EAV) migration assistance tracker”, of *z/OS V1.11 DFSMSdfp Advanced Services,* SC26-7400 for more details. The tracking of these instances helps you identify the programs and jobs that might need migration actions performed during implementation and migration of data to EAVs.

LSPACE (SVC 78)

An LSPACE request with the DATA=, MSG=, or EXPMSG= keywords is issued. Additional data from track-managed space is available through the EXPDATA= and XEXPMSG= keywords. See 4.5.7, “LSPACE macro usage with EAVs” on page 131 and 4.10.3, “LSPACE macro” on page 144 for more information.

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 4-byte field. See 4.5.8, “DEVTYPE macro usage with EAVs” on page 133 for more details.

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 might contain 28-bit cylinder numbers. See 4.5.7, “LSPACE macro usage with EAVs” on page 131 for more information.

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 might contain cylinder addresses 65,520 or larger in size. Free space descriptors might contain track addresses 982800 or larger or full cylinders 65,520 or larger in size. The generated report displays the information in different columns as compared to reports generated on releases prior to z/OS V1.10. See 4.3.6, “IEHLIST LISTVTOC support” on page 101 for more information.

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 V1.10). The returned extent descriptors might contain 28-bit cylinder numbers. This instance is recorded for both EAS and non-EAS capable volumes. Please note that AMS LISTCAT output format might change as a result of service and new function support. IBM recommends applications that process LISTCAT output be updated to obtain results directly from the Catalog Search Interface (CSI). For more information about CSI, see *z/OS DFSMS Managing Catalogs*, SC26-7409 and *High Level Assembler Programmer's Guide*, SC26-4941. See 4.3.7, “LISTCAT support” on page 102 for more information. 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 is recorded in the tracker as an error message. This action becomes a problem if the volume becomes an EAV and the VTOC contains DSCB Format 8 or Format 9. See 4.2.7, “New format DSCBs for EAV volumes” on page 92 and 4.5.4, “What programs require modification for EAV” on page 125 for more information.

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 a 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. This actions becomes a problem if the volume becomes an EAV and the VTOC contains DSCB Format 8 or Format 9. See 4.2.7, “New format DSCBs for EAV volumes” on page 92 and 4.5.4, “What programs require modification for EAV” on page 125 for more information.

CVAFSEQ

CVAFSEQ was issued to provide 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 is recorded in the tracker as an error message. This action becomes a problem if the volume becomes an EAV and the VTOC contains DSCB Format 8 or Format 9. See 4.2.7, “New format DSCBs for EAV volumes” on page 92 and 4.5.4, “What programs require modification for EAV” on page 125 for more information.

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' are set if CVAFDSM is issued to a volume that supports extended attribute DSCBs.

When this instance occurs for a volume that does not support extended attribute DSCBs, it is recorded in the tracker as an error message. This actions becomes a problem if the volume becomes an EAV and the VTOC contains DSCB Format 8 or Format 9. See 4.2.7, “New format DSCBs for EAV volumes” on page 92 and 4.5.4, “What programs require modification for EAV” on page 125 for more information.

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 EADSCB=OK so that it supports the extended attribute DSCBs and the qualified data set is EAS-eligible. CVAF return code 4 and CVSTAT of X'56', along with data set name status in the FCL (FCLDSNST) of X'06', are 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 is recorded in the tracker as an error message. This action becomes a problem if the volume becomes an EAV and the VTOC contains DSCB Format 8 or Format 9. See 4.2.7, “New format DSCBs for EAV volumes” on page 92 and 4.5.4, “What programs require modification for EAV” on page 125 for more information.

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 is recorded in the tracker as an error message. This action becomes a problem if the volume becomes an EAV and the VTOC contains DSCB Format 8 or Format 9. See 4.2.7, “New format DSCBs for EAV volumes” on page 92 and 4.5.4, “What programs require modification for EAV” on page 125 for more information.

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 is recorded in the tracker as an error message. This action becomes a problem if the volume becomes an EAV and the VTOC contains DSCB Format 8 or Format 9. See 4.2.7, “New format DSCBs for EAV volumes” on page 92 and 4.5.4, “What programs require modification for EAV” on page 125 for more information.

Note: Details of the tracker can be found in APAR II13752 and Appendix A, “Tracking Facility”, in *z/OS V1.11 Planning: Operations*, SA22-7601. Additional details about tracking instances can be found in Chapter 9, “Using extended address volumes”, in *z/OS V1.11 DFSMS Using the New Functions,* SC26-7473 and Appendix C, “Using the extended address volume (EAV) migration assistance tracker”, in z/OS V1.11 DFSMSdfp Advanced Services, SC26-7400.

4.5 EAV migration issues

A number of migration issues have to be considered:

- VSAM CA size compatibility.
- SMF records changes.
- VVDS access.
- What programs require migration actions.
- How to find affected programs.
- Programs that access VTOCs.
- Updates to LSPACE macro usage.
- Updates to DEVTYPE macro usage.
- Migration from non-EAV to EAV volumes in z/OS V1.11, mainly due to the introduction of the EATTR attribute.

4.5.1 VSAM CA size compatibility

All VSAM data sets created in z/OS V1.10 might have different CA sizes than in prior releases. The reason for this change is that a CA must be compatible with the multicylinder unit (MCU) for cylinder-managed space. The list of CA sizes allows any VSAM data to be set as EAS-eligible that otherwise would not have been. All these are divisors of 315 tracks (21 cylinders).

VSAM data sets allocated with compatible CAs on a non-EAV are eligible to be extended to additional volumes that could be EAVs. Also note that VSAM data sets physically copied from a non-EAV to an EAV might have an incompatible CA and thus would not be EAS eligible. This means that additional space would not use cylinder-managed space. The system may adjust the primary and secondary quantity to select a CA.

APAR II14458

The title of APAR is “VSAM data set index cisize may change when defined or redefined”. When a VSAM data set or ICF catalog is defined or redefined on z/OS V1.10, the index cisize may be increased. This can occur if the data CA size was in tracks and the amount specified was not 1,3, 5, 7, 9, or 15 tracks. For example, if the space requested was TRK(2,2), the allocation amount will be TRK(3,3) on z/OS V1.10 from that point. Because the data CA size is now larger, the minimum index cisize may also increase. For those applications such as IMS™ or CICS® that have or might have a static LSR pool definition, a change to the index cisize might cause a data set to no longer open after being redefined.

There is a new tool that is now available to find those data sets that have a larger CA size. You may obtain this tool at the following website:

`ftp://ftp.software.ibm.com`

The tool is in the following directory:

`/servers/storage/support/software/dfsms/ as INDXCI10.JCL.CNTL.TRSD.`

This tool must be downloaded in binary mode to a data set with the following attributes:

`LRECL=1024,BLKSIZE=6144,RECFM=FB,DSORG=PS`

After downloading the tool, it needs to be expanded through either AMATERSE or TRSMAIN into a PDS. The PDS contains a REXX exec member called INDXCI10 and a job member INDXJCL to run the exec. Review the comments in the JCL for more information about the tool. Figure 4-29 is an example of the INDXJXL job that runs the INDXCI10 exec.

```
//STEP1     EXEC PGM=IKJEFT1B
//SYSPROC   DD DISP=SHR,DSN=CVERNON.INDXCI10.JCL.CNTL
//* the above DSN should be this data set or wherever you placed the
//* member INDXCI10
//SYSPRINT  DD SYSOUT=*
//SYSTSPRT  DD SYSOUT=*
//SYSTSIN   DD *
    %INDXCI10  ** CATALOG(MCAT.SANDBOX.Z1B.SBOX00)
    %INDXCI10  ** CATALOG(UCAT.VSBOX01)
/*
//*
//* A filter key of '**' causes a search of the entire catalog in the
//* CATALOG() parm. Anything other than a '**' would do a search based
//* on the key and report accordingly. In the above examples the first
```

Figure 4-29 INDXJCL job

Figure 4-30 shows the output from the INDXJCL job in Figure 4-29 on a test system where only one data set was found. More data sets would be expected to be found on production and development systems that have many older VSAM files.

```
READY
    %INDXCI10  ** CATALOG(MCAT.SANDBOX.Z1B.SBOX00)

 Catalog Search Interface is being called with filter key: **
 and catalog field name(s):  VSAMTYPE NOTRKAU AMDCIREC VSAMSTAT AMDKEY

                                      Version 1.0

              DATA CA Size and INDEX CI Size Changes When Migrating to z/OS Version 1 Release 10

CLUSTER NAME                            PRE-R10  CA-SIZE IN  PRE-R10    CI-SIZE IN
                                        CA-SIZE     R10      CI-SIZE       R10

SYS1.APPCTP                             12 TRKS    15 TRKS     512      1024
READY
```

Figure 4-30 Example of output from INDXCI10 exec

The full text of APAR II14458 that documents the tool's purpose and its procurement is listed in "APAR II14458" on page 519.

4.5.2 SMF record changes

There are two new SMF flags to indicate that the EADSCB=OK keyword on the DCBE macro might need to be specified only if at some point this data set becomes EAS eligible. In z/OS V1.11, EAS eligible data sets are VSAM data sets and are extended format sequential. The VTOC is not EAS eligible, but is a data set that might contain Format 8 and Format 9 DSCBs. For data sets that are currently not EAS eligible, the specification or absence of the EADSCB=OK keyword will have no effect. Upgraded programs continue to operate correctly on down-level systems.

SMF record types 14 and 15 (non-VSAM data set EOV or close) have been modified in the following ways:

- A new flag, SMF14EADSCB, indicates whether a program specified the EADSCB=OK keyword on the DCBE macro and indicates the following items:
  - Specific migration aid in z/OS V1.10 to identify programs that open the VTOC or VSAM data sets with EXCP (MACRF=E).
  - In z/OS V1.10 or later this data set may have a Format 8 DSCB. Upgrade the program to handle 28-bit cylinder numbers and code EADSCB=OK.
- A new flag, SMF14EXCPBAM, indicates that a program used a non-EXCP OPEN DCB (BSAM, QSAM, BPAM) and issued EXCP or XDAP.
  - When SMF14EADSCB is off, it identifies programs that might need to be upgraded to handle 28-bit cylinder numbers and have EADSCB=OK coded. These actions are taken in preparation for a future release.

Note: The VTOC cannot be in cylinder-managed space, but opening a VTOC on a volume that supports extended attribute DSCBs still requires the EADSCB=OK keyword, because the VTOC might contain Format 8 and Format 9 DSCBs.

If the new flag, SMF14EXCPBAM, is not set, then EXCP or XDAP was not issued while it was opened for BAM processing. No changes are required in this case, as BAM internally handles the 28-bit cylinder numbers, for example, in the case of extended format sequential. If it is set, then an EXCP or XDAP was specified while it was opened for BAM processing and this program needs to be upgraded to handle 28-bit cylinder numbers. In the future, this instance will be fail with message IEC142I 113-44 if a data set has a Format 8 DSCB and EADSCB=OK was not specified on the DCBE macro.

Other changed SMF records

Other SMF records have also been changed:

- SMF Type 19 record (volume statistics)

  The LSPACE statistics recorded is expanded with track-managed free space statistics and total volume/track-managed space sizes.
- SMF Type 60, 61, 64, 65, and 66 records (various VSAM events)

  Each of these SMF records contains a copy of a catalog record for a VSAM data set. They might contain extent descriptors with cylinder numbers greater than 65535.
- SMF Type 74, subtype 1 (RMF™ device activity)

  A new device capacity field is saved.

4.5.3 VVDS access

The VVDS, a catalog volume data set, contains extent descriptors for VSAM data sets. For data sets that are eligible to be allocated or extended on an EAV volume, its extent descriptors might contain 28-bit cylinder addresses. Prior to EAV support, these extents are stored in an internal VVDS structure called a VVR.

Some in-house or third-party programs might access the VVDS even though no interfaces are provided for this purpose. Programs that access the extents in the VVR record must ensure that they support 28-bit cylinder numbers. Use the TRKADDR macro or the IECTRKAD routine.

Important: It is very important that these programs that refer to the extents within the VVR be reviewed, and possibly modified, to ensure they can handle the 28-bit cylinder addresses.

4.5.4 What programs require modification for EAV

In a future release, programs issuing EXCP or XDAP will fail for a non-VSAM data set when:

1. The access method is BSAM, BPAM, or QSAM.
2. The data set has a Format 8 DSCB and EADSCB=NOTOK (default value).

The techniques described in the following sections should be used to find and upgrade these programs and add the EADSCB=OK keyword option to the DCBE macro. The upgraded program will continue to operate correctly on downlevel systems.

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 future z/OS releases, IBM expects to support all non-VSAM data sets, which might lead to problems with these types of programs if they are not properly coded.

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

Programs that do not use OPEN and issue EXCP, EXCPVR, XDAP

Each of these macros requires a DCB and a DEB. If the application program builds its own DEB, it can perform that action 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.

Detection: It is necessary for a programmer to search the source code and run tests to find these programs.

Action: Make code changes to handle and process 28-bit cylinder numbers correctly. See “Changing programs that issue EXCP, EXCPVR, or XDAP” on page 126 for more information.

EXCP OPEN and issue EXCP, EXCPVR or XDAP

The DCB has MACRF=E, which signifies that the DCB is for EXCP. This signifies that the DCB is only for EXCP and that bit 0 (DCBMRECP) in DCBMACRF is set.

To find instances of these EXCP DCBs, the programmer can search the source code and also 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 2-byte SMFDCBMF field contains a copy of the DCBMACRF field. If bit 0 is set, it means that the DCB is only for EXCP (MACRF=E). The JFCB contains 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 as been upgraded to handle EAV because the program must supply a DCBE with the EADSCB=OK keyword coded. In z/OS V1.10, if an attempt is made to open an EAS eligible data set (VSAM) where a Format 8 DSCB exists, or open the VTOC and the EADSCB=OK keyword is not coded, a new ABEND and message IEC142I 113-44 is issued. In a future release, where other data set types will be EAS eligible, then this condition will occur for them as well.

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 this program is expected to be used with a data set that has a Format 8 DSCB in z/OS V1.10 or later, then upgrade the program to handle 28-bit cylinders by coding the EADSCB=OK keyword.

Detection: You can search SMF14 and 15 records as outlined above, but you can also take advantage of the migration tracker. See “DCB Open of EAS eligible data set (VSAM)” on page 121 for more details.

Action: Make code changes to handle and process 28-bit cylinder numbers correctly. See “Changing programs that issue EXCP, EXCPVR, or XDAP” on page 126 for more information.

Non-EXCP OPEN and EXCP or XDAP

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

The DCB is for an access method other than EXCP. In this case, bit 0 of DCBMACRF and in SMFDCBMF are not 1. It is valid to issue EXCP but not EXCPVR in this case. An EXCPVR would result in an ABEND400.

Programmers should search the source code for those instances of EXCP and XDAP. In addition, the following items are true:

- A new SMF14/15 flag, SMF14EXCPBAM, is defined to aid in the discovery of programs that issue a non-EXCP OPEN and issue EXCP or XDAP. This flag is 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 the EADSCB=OK keyword on the DCBE macro. Upgrade the program to handle 28-bit cylinders and code the EADSCB=OK keyword.

Detection: You can search SMF14 and 15 records as outlined above, but there are no migration tracker instances that can help identify this class of programs.

Action: Make code changes to handle and process 28-bit cylinder numbers correctly. See “Changing programs that issue EXCP, EXCPVR, or XDAP” on page 126 for more information.

Changing programs that issue EXCP, EXCPVR, or XDAP

The simplest way to handle 28-bit cylinder numbers is to use the TRKADDR macro, or use the IECTRKAD routine for high level language programs. The macro works equally well with track addresses that contain 28-bit and 16-bit cylinder numbers and with any DASD data set. It also works correctly on pre-z/OS V1.10 systems. This is also prepares support of non-VSAM data sets beyond the currently supported extended format sequential type in z/OS V1.11.

In a future release, issuances of EXCP or XDAP are expected to fail for a non-VSAM data set when the access method is BSAM, BPAM, or QSAM and the data set has a Format 8 DSCB.

Find and upgrade these programs and add the EADSCB=OK keyword option to the DCBE macro. The upgraded program continues to operate correctly on downlevel systems.

Programs that access VTOCs

Programs that access VTOCs through either the DADSM OBTAIN macro of the CVAF macros need to be modified to correctly handle the new format DSCBs (8 and 9). Here are some examples of these programs:

- When issuing an OBTAIN macro to read a DSCB for an EAS-eligible data set or issuing a CVAF macro for an EAS-eligible data set, a new EADSCB=OK keyword option must be coded. For performance reasons, one might want also to use a new option to read all the DSCBs for a specified data set with one call and use MULTIPLEDSCB=YES.
- Programs that read VTOCs or DSCBs. In V1.10, programs designed to read an EAV VTOC expect to see Format 8 DSCBs for only VSAM data sets. In V1.11, those programs can also see Format 8 DSCBs for extended-format sequential data sets. If EAVs are shared between V1.10 and V1.11, avoid opening non-VSAM data sets that have Format 8 DSCBs or prevent those data sets from being on those volumes.
- Programs that can read VTOCs or individual DSCBs with BSAM, QSAM, EXCP, OBTAIN, CVAFDIR, and CVAFFILT.
- Examine those VTOC reading programs to see whether they might be affected by a Format 8DSCB when they expect a Format 1 DSCB or they might be affected by a Format 9 DSCB when they expected a Format 3 DSCB.

Detection: You can take advantage of the extensive number migration tracker instances. See 4.4.3, “DFSMS tracking instances” on page 119 for more information.

Action: Ensure that the code handles 28-bit cylinder numbers and Format 8 and Format 9 DSCBs correctly and add the EADSCB=OK to DADSM OBTAIN or the CVAF macros. See 4.2.7, “New format DSCBs for EAV volumes” on page 92 and 4.5.6, “Further considerations about accessing EAV VTOCs” on page 128 for more information. For correct track address handling, see “Changing programs that issue EXCP, EXCPVR, or XDAP” on page 126.

4.5.5 Sharing EAVs, EAS data sets, and other considerations

An existing product or program might be affected by the EAV support if any of the following situations occur. This list might be incomplete, but it specifies some of the necessary processing for data sets that are EAS-eligible with some z/OS V1.11 enhancements.

- If a program bypasses OPEN, it is likely that any channel program issued for an EAS-eligible data set is affected, so that program needs to be checked that it properly handles 28-bit cylinder numbers and Format 8 and Format 9 DSCBs. We consider the ones that use the EXCP I/O driver, but there might be other I/O drivers used for third-party software or developed in house. These programs need to be examined carefully.
- Any program using a track address for an EAS-eligible data set, such as those in IOSEEK in the IOSB, IOBSEEK in the IOB, or DS1EXT1 in the DSCB. As of the first release of EAV, these control blocks contain 28-bit cylinder numbers.
- Any program using the BBCCHHR field in the I/O error text returned by the SYNADAF macro. This is in EBCDIC form. It is similar to text returned by VSAM.

- Any program that calculates the size of an EAS-eligible VSAM or non-VSAM data set from the cylinder and track numbers of its extents. These cylinder and track numbers might be from a VTOC, DEB, or from an access method internal control block. The TRKADDR macro or IECTRKAD routine can assist with this calculation.
- Examine programs that calculate the size of a data set on a volume. The value might be larger than the program has ever seen. The TRKADDR macro is available to assist with these calculations.
- Although there are no intended programming interfaces for channel programs with extended-format data sets, any such programs must take the 28-bit cylinder numbers into account at V1.10 if sharing with a V1.11 system. The 28-bit cylinder numbers might be in a DSCB, IOB, or channel program. The TRKADDR macro is available to assist with manipulating track addresses.

Attention: If a non-VSAM data set is created on V1.11 on an EAV and it might be used on V1.10, then EATTR=OPT must not be used and EATTR=NO should be used for that data set. This action is not strictly necessary, because if an EATTR value is not specified for non-VSAM, it will be treated like EATTR=NO. If the data set has a Format 8 DSCB, it cannot be opened in V1.10.

If the program issues the OBTAIN, CVAFDIR, or CVAFFILT macro for a data set that has a Format 8 DSCB, then the macro must have EADSCB=OK.

4.5.6 Further considerations about accessing EAV VTOCs

With z/OS V1.11, access to the VTOC records has been enhanced to allow a program to read the logical DSCB chain for a data set in one invocation of a service. The OBTAIN and CAMLST routines have a new option to specify the number of DSCBs that might be read.

You can use either DADSM or common VTOC access facility (CVAF) macros to access a VTOC and its index. The DADSM macros include:

- LSPACE provides information about volume size, free space on the volume, free space on the VTOC and INDEX, volume fragmentation, and VTOC status. Information on the size of the track-managed space and its free space statistics is also provided.
- OBTAIN reads one or more DSCBs from the VTOC.
- PARTREL releases unused space from a sequential or partitioned data set or a PDSE.
- REALLOC allocates DASD space.

If you specify a data set name using OBTAIN and the CAMLST SEARCH option, the OBTAIN routine reads the 96-byte data portion of the Format 1 DSCB and the absolute track address of the DSCB into virtual storage. The absolute track address is a 5-byte field in the form CCHHR (it contains zero VIO data sets).

Accessing DSCBs on EAV volumes

To read one or more DSCBs into virtual storage, use the OBTAIN and CAMLST macro instructions. Identify the DSCB to be read using the name of the data set associated with the DSCB, or the absolute track address of the DSCB. Provide a 140-byte data area in virtual storage to contain the DSCB. For a request to read multiple DSCBs, specify the NUMBERDSCB= keyword. This is a new keyword with z/OS V1.10 and z/OS V1.11 for the OBTAIN and CAMLST macros and provides consecutive 140-byte return areas in virtual storage to contain this number of DSCBs.

When you specify the name of the data set, an identifier (Format 1, Format 4, or Format 8) DSCB is read into virtual storage. To read a DSCB other than a Format 1, Format 4, or Format 8 DSCB, specify an absolute track address. Code the EADSCB=OK in the OBTAIN or CAMLST macro when your program supports DSCBs that describe data sets with Format 8 and Format 9 DSCBs. The extent descriptors in DSCBs for a data set described with these formats might have 28-bit cylinder track addresses. Use the TRKADDR macro or IECTRKAD service to manipulate 16-bit or 28-bit cylinder track addresses.

MULTIPLEDSCB parameter

Specifying MULTIPLEDSCB=YES indicates that the calling program requests to read and write multiple DSCBs to and from a buffer list that contains more than one buffer list entry:

- This flag resolves to a new indicator in the CVPL, CV4MULTD.
- Multiple DSCB processing for reads and writes is requested by specifying the MULTIPLEDSCB=YES keyword and providing a buffer list that contains more than one buffer list entry (BFLHNOE>1).

Specifying MULTIPLEDSCB=NO indicates that the calling program requests that only one DSCB should be processed. This is the default for MF=L and MF=I forms of the CVAFDIR macro:

- When the MULTIPLEDSCB keyword is not specified on the MF=E form, the existing setting of CV4MULTD is left unchanged.
- When MULTIPLEDSCB=NO is specified or defaulted, only the first available buffer list entry is processed.

OBTAIN macro

OBTAIN search processing stores an additional two bytes in the callers return area right after the 101 bytes that is set by the OBTAIN macro in prior releases. You must ensure programs provide the minimum 140-byte return area. These two bytes are set to the total number of consecutive 140-byte areas that are needed to read all the DSCBs for the data set.

For SEARCH and SEEK requests, number_dscbs is an absolute expression with a value between 0 and 255 that designates the number of consecutive 140-byte return areas that are provided in a work area called wkarea_relexp. The system treats a value of 0 as 1. Prior to z/OS V1.11, the system did not support a chain of more than 12 DSCBs for one data set, but it is valid to provide an area that is longer than currently needed. The system verifies that the provided area is valid. When an area is provided that is long enough to contain more than one DSCB, obtain processing returns DSCBs for the requested data set name in logical VTOC order until all the 140-byte return areas are used.

The logical VTOC order is a Format 1 DSCB, followed by zero or more Format 3 DSCBs or a Format 8 DSCB, followed by one or more Format 9 DSCBs, followed by zero or more Format 3 DSCBs. No absolute maximum number of DSCBs for a data set should be assumed. For SEARCH requests, the actual number of DSCBs is returned in a field located in the first 140-byte return area. For SEEK requests where the target of the seek operation is not a Format 1 or Format 8 DSCB, the NUMBERDSCB value is treated as though it were 1 and only that single DSCB is returned.

Note: For programs running on an older level of the system that does not support this keyword, the NUMBERDSCB value is treated as though it were 1.

CVAFDIR macro

For an indexed or non-indexed VTOC, you can use the CVAFDIR macro to perform the following functions:

- Read or write one or more DSCBs by specifying the name of the data set they represent.
- Read or write one or more DSCBs by specifying their addresses.

CVAFDIR provides a new keyword MULTIPLEDSCB=YES to indicate to CVAFDIR processing to use the multiple buffers passed in the buffer list.

Multiple DSCB reads

When you use the MULTIPLEDSCBS parm, CVAFDIR reads more than 1 DSCB at a time:

- Only the first buffer list entry, seek or search, argument is used. This provides orientation to the data set from which the subsequent data set DSCBs will be read. As each one is read, the DSCB argument (BFLEARG) in each buffer list entry is set in the format specified by the caller in each buffer list entry. The buffer list argument is updated with the BFLEAUPD flag set. The data set DSCBs are ready in the logical VTOC order and will continue as long as buffer list entries are available to return the DSCB. A new field in the buffer list header (BFLHNOEN) is set by CVAFDIR read processing to indicate the number of buffer list entries that are needed to read the entire set of DSCBs for the data set. This number is set in the header of the first buffer list and is independent of the specification of the MULTIPLEDSCBS keyword, the target volume type, and whether the number of provided buffer list entries, BFLHNOE, is short.
- The logical VTOC order would be either:
  - Format 1 -> Format 3s
  - Format 8, -> Format 9s -> Format 3s
- Buffer list entries other than the first one must provide a 140-byte buffer. The first buffer list entry buffer size follows the same rules as today, that is, with the seek option, it provides either a 96-byte or 140-byte buffer and provides a 96-byte buffer for the search option. All other buffer list entry processing flags, as described in the buffer list entry flag byte (BFLEFL), should continue to be supported. These flags include data in the buffer modified, skip, I/O error, no key verify, and argument format qualifiers. The DSCB argument (BFLEARG) returned in each buffer list entry is in the format determined by the argument format qualifiers (BFLECHR or BFLETTR) in each buffer list entry.

Multiple DSCB writes

Here is more information about the MULTIPLEDSCB CVAFDIR writes:

- The buffer list header must indicate the number of buffer list entries passed. Only buffer list entries without the skip flag set are processed. The order in which the DSCBs are passed in the buffer list entries must correspond to the logical VTOC order. A new flag in the buffer list header can be set by the caller to indicate that the logical order in which the DSCBs appear in the buffer list must be written in reverse order (BFLHWREV).
- For write processing, the first buffer list entry can be a 96-byte buffer if the DSCB is in Format 1. The same holds true for a Format 8 DSCB. A 140-byte buffer can also be provided for these DSCBs as long as the BFLEARG points to the actual DSCB that needs to be written. Buffer list entries that do not describe Format 1 or Format 8 DSCBs must provide a 140-byte buffer and its buffer address (BFLEARG) must point to the actual DSCB that needs to be written. The caller must also specify the buffer list entry argument (BFLEARG) as a CCHHR for these buffer list entries. A new flag in the buffer list entry could be set for entries where you do not need to verify a Format 0 DSCB before a write (BFLENVER). This overrides the VERIFY=YES setting.

4.5.7 LSPACE macro usage with EAVs

The LSPACE macro can be used to get free space, volume fragmentation, and volume table of contents (VTOC) status information for a direct access storage device (DASD) volume. The LSPACE macro returns status information (such as LSPACE subfunction, return code, and reason code) in the parameter list. The LSPACE macro also returns the return code in register 15. For volumes that are configured with more than 9999 cylinders, the EXPMSG option can be used to create an expanded message return area that the LSPACE macro needs. For volumes that are configured with more than 65,520 cylinders, the XEXPMSG option can be used to create an extended expanded message return area that the LSPACE macro needs. The expanded data return area (EXPDATA) returns binary data of free space information for volumes with more than 65,520 cylinders. LSPACE macro can return additional information, such as the Format 4 DSCB, the total number of free extents on the volume, or the fragmentation index.

LSPACE macro new keywords

Four new keywords are added to the LSPACE macro:

- XEXPMSG
- EXPDATA
- DATATYPE
- PLISTER

The LSPACE macro returns information about a DASD volume, as follows:

- Returned information can be in character or binary format.
- A new XEXPMSG= keyword keyword performs the following action:
  - Specifies the address of a caller-provided 95-byte extended expanded message return area into which LSPACE returns either a free space message or, for unsuccessful requests, status information. Specify this keyword to obtain free space information in the message return area for volumes that are configured with more than 65,520 cylinders. The returned free space includes space for the total volume and space from the track-managed space on a volume. For volumes with 65,520 cylinders or less, both sets of free space information are returned, but they are the same.
- A new EXPDATA= keyword keyword performs the following action:
  - Specifies the address of a caller-provided expanded data return area into which LSPACE returns expanded free space and volume information. Specify this keyword to obtain free space information in the LSPACE data return area for volumes that are configured with more than 65,520 cylinders. The returned free space includes space for the total volume and space from the track-managed space on a volume. For volumes with 65,520 cylinders or less, both sets of free space information are returned, but they are the same.
- There is a new DATATYPE keyword, as shown in Figure 4-31.

```
  {ALL        }Return all of the following information
      {VOLUME    } Return free space for volume
      {VTOC         } Return free space for VTOC
      {INDEX        } Return free space for index
      {FRAGINDEX}Return the fragmentation index
```

Figure 4-31 LSPACE new DATATYPE keyword

Where:

- VOLUME: Provides free space information for the VOLUME.
- VTOC: Provides free space information related to the VTOC.
- INDEX: Provides free space information related to the VTOC INDEX.
- FRAGINDEX: Provides the fragmentation index.
- ALL: Provides all the available LSPACE statistics. This is the default.

This DATTATYPE keyword is only allowed when the DATA or EXPDATA keyword is specified. Only the information specified is returned to the caller. DATATYPE is valid for both non-EAV and EAV. This keyword eliminates the unnecessary I/O required to retrieve free space information that is not be required by the caller. DATATYPE=ALL is the default.

- There is also a new PLISTVER keyword to manage the use of the longer LSPACE parameter list:

```
PLISTVER=plistver | IMPLIED_VERSION | MAX.
```

This keyword defines the version of the LSPACE parameter list that should be generated for the MF=L form of the LSPACE macro. The value for PLISTVER is a byte input decimal value in the “1-2” range that specifies the version of the LSPACE parameter list that should be generated. The macro keys associated with each supported version of the macro are listed below. This PLISTVER= keyword is required for any macro keys associated with Version 2 or later to be specified as shown in Figure 4-32.

```
VERSION  KEY
   1          MSG
              EXPMSG
              DATA
              SMF
              F4DSCB
   2          XEXPMSG
              EXPDATA
              DATATYPE
```

Figure 4-32 PLISTVER= keyword of LSPACE

- When MAX is specified, the generated parameter list is the largest size currently supported. This size might grow from release to release, thus possibly affecting the amount of storage needed by programs. If a program can tolerate this growth, IBM always specify MAX when creating the list form parameter list, as it ensures that the list form parameter list is always long enough to hold whatever parameters might be specified on the execute form.
- When IMPLIED_VERSION is specified, the generated parameter list is the lowest version that allows all of the parameters on the invocation to be processed.
- When PLISTVER is omitted, the default is the lowest version of the parameter list, which is version 1.

4.5.8 DEVTYPE macro usage with EAVs

The DEVTYPE macro with the existing INFO=DASD parameter returns the number of cylinders in a 16-byte area provided by the caller. The new fields returned include the following fields:

- 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.
- The block size of the index.

The DEVTYPE macro allows you to obtain device characteristic information about I/O devices:

- The DEVTYPE macro issued without the INFOLIST parameter returns a 2-byte value for the number of cylinders, and is not valid for an EAV.
- The DEVTYPE macro issued with the INFOLIST parameter (INFO=DASD) returns a different format of the device characteristics information. This includes a 4-byte value for the number of cylinders and is mapped to field DVAICYL in the shipped mapping macro, IHADVA. The 2-byte field is mapped by DVACYL.

  In addition, the macro now returns the following information:

  – Multicylinder unit value
  – First cylinder address where cylinder-managed space begins
  – Cylinder-managed space supported indicator
  – Extended attribute DSCBs supported indicator
  – Block size of index data set

The added DEVTYPE field names and their description are listed in Table 4-1.

Table 4-1 DEVTYPE new fields

DVAIXVLD	BIT	DVACYLMG, DVAEADSCB, DVAVIRSZ valid.
DVACYLMG	BIT	Cylinder-managed space exists on this volume and begins at DVALCYL in multicylinder units of DVAMCU. DVAEADSCB is also set with this flag on. Valid when DVAIXVLD is set.
DVAEADSCB	BIT	Extended attribute DSCBs, Format 8 and Format 9 DSCBs, are allowed on this volume. Valid when DVAIXVLD is set.
DVAMCU	8-bit integer	The minimum allocation size in cylinders for cylinder-managed space. Each extent in this space must be a multiple of this value. Also referred to as the multicylinder unit (MCU). This is the smallest unit of disk space in cylinders that can be allocated in cylinder-managed space. Valid when DVACYLMG is set. This field is zero on releases before z/OS V1.V1.10 or if the status is not yet known. In these two cases, DVAIXVLD is not set.

DVAIXVLD	BIT	DVACYLMG, DVAEADSCB, DVAVIRSZ valid.
DVALCYL	16-bit integer	The first cylinder address divided by 4095 where the space is managed in multicylinder units. Cylinder-managed space begins at this address. Valid when DVACYLMG is set. This field is zero on releases before z/OS V1.10 or if the status is not yet known. In these two cases, DVAIXVLD is not set.
DVAVIRSZ	16-bit integer	The block size of the index data set. Valid when DAVIXVLD is on. When valid and at zero, the volume has no working VTOC index. This field is zero on releases before z/OS V1.10 or if the status is not yet known. In these cases, DVAIXVLD is not set.

4.5.9 Sharing EAVs between z/OS V1.11 and V1.10

Migration and coexistence considerations for the new EATTR data set attribute have to be considered. Care must be taken when specifying EATTR=OPT for data sets that are not EAS-eligible in z/OS V1.10. These data sets are extended-format sequential data sets. These data sets would not be allowed to be opened from z/OS V1.10 if they were allocated in z/OS V1.11 with extended attributes. Even if they do not currently reside in EAS, they have Format 8 and 9 DSCBs for the data set in the VTOC.

You must be certain that any application that accesses any non-VSAM data set with EATTR=OPT (though only extended format sequential are EAS-eligible in V1.11) can handle extended attributes and 28-bit cylinder addresses because they might be EAS eligible in the future.

APARs might be needed to allow DB2 or IMS to support extended-format sequential data sets in EAS.

EAV capable systems

z/OS V1.11 can share EAVs with z/OS V1.10 systems until explicit action is taken to begin allowing non-VSAM data sets to reside in the EAS of an EAV. This is because EATTR defaults to NO for non-VSAM data sets. However after EATTR=OPT is specified, whether it actually resides in EAS or not it will still have extended attribute DSCBs created for it, making it unable to be opened on V1.10.

EATTR=OPT can be actually specified for non-EAS eligible data sets. Because this setting will be remembered and stored in the Format 1 DSCB, it becomes an accident waiting to happen the moment the data set type is enabled for EAS eligibility. There is no effect until the data set type is enabled in the system for EAS.

Attention: A non-VSAM data set allocated with Format 8 and Format 9 DSCBs in z/OS V1.11 is not able to be opened on a pre-z/OS V1.11 system.

Be sure that certain applications can handle extended attribute DSCBs and 28-bit cylinder numbers, as they might become EAS-eligible data sets in the future.

Considerations for migration to z/OS V1.11

Because z/OS V1.11 now supports the non-VSAM type of extended format sequential data set and the creation of extents straddled across track-managed storage and cylinder-managed storage, you need to be aware of the differences and limitations when shared with V1.10.

VTOC entry and DCSBs can still be read for unsupported data sets

Application programs issuing the CVAF and OBTAIN macros with the EADSCB=OK keyword will be able to see Format 8 and Format 9 DSCBs for types of data sets that cannot be opened on V1.10. A Format 8 DSCB can be found, where none might have been before, even if that release does not support that type of data set for EAS. This change is being made via PTF on z/OS V1.10 as well.

The programmer might expect a failure of CVAF or OBTAIN if the data set has a Format 8 DSCB and cannot be opened in the current release. Only the OPEN of the data set will fail, but not read its VTOC entry or DSCBs. This situation occurs for Format 8 DSCBs created on V1.11 when the EAV is shared with 1R10.

Straddled extents

A single extent can begin in track-managed space and end in cylinder-managed space. Extents of this type will be recognized on z/OS V1.10 but not created there. This consideration is for programs that might have expected each extent to begin and end in either track-managed or cylinder-manager space, as it did in V1.10, but not begin in one and end in the other.

You might expect a large space request on an empty volume to be split as it was in V1.10. If ALX (all extents) or MXIG (maximum single extent) is coded in the SPACE parameter on V1.11, the same amount of space can be obtained as in V1.10, but it might have less extents than before. If space is fragmented, this might allow space allocation in some cases where it could not have succeeded before. One case is where six extents would have been required and only five are allowed. Another case is where a large contiguous space request using CONTIG might have failed, but might now succeed through a contiguous straddled single extent.

APAR OA26623 on V1.10 causes DADSM to tolerate an extent that straddles the track-managed and cylinder-managed space. DADSM with the PTF in V1.10 does not create such an extent.

4.5.10 Indication of EAS eligible data sets

The bits in the DFAFEAT8 field can be tested in the DFA and mapped by IHADFA, to learn which data set types can be opened with Format 8 DSCBs, as shown in Figure 4-33.

```
DFAVSAMFOREAS  EQU X'80'      VSAM enabled for EAS
DFASEQFOREAS   EQU X'40'      Basic, large format seq (QSAM, BSAM
*                             BDAM access) enabled for EAS
DFAPDSEFOREAS  EQU X'20'      PDSE enabled for EAS
DFAPDSFOREAS   EQU X'10'      PDS enabled for EAS
DFADIRFOREAS   EQU X'08'      Direct (BDAM access) enabled for
*                             EAS
DFAEFSEQFOREAS EQU X'04'      Ext format seq enabled for EAS
DFAUNDEFFOREAS EQU X'02'      Undefined DSORGs enbld for EAS
```

Figure 4-33 Indication of EAS eligible data sets

Attention: In V1.10, only the VSAM bit will be set (with OA27577 on V1.10).

In V1.11, with DFAFEAT8=X'84', the VSAM and extended format sequential bits will be set. (However, additional data set types might be eligible in the future.)

4.6 TSO/E support for EAV volumes

Some SO/E services and commands fail without EAV support. REXX execs and CLISTs could use information about EAV data sets. In z/OS V1.11, DFSMS added support for extended format sequential data sets to reside in the EAS on EAV. This setup allows data sets to reside on volumes with more than 65,520 addressable cylinders by using a new format of DSCB. While VSAM data sets were supported for EAV in z/OS V1.10, TSO/E commands and services did not support VSAM data sets in V1.10 that could reside in EAS.

In z/OS V1.11, data sets can now be allocated using the EATTR keyword with the ALLOCATE command. Support for the new ALLOCATE command keywords was added by the allocation component, but it works like any other keyword for ALLOCATE. The options are shown in the following list:

EATTR(NO) No extended attribute. The data set cannot have extended attributes (Format 8 and Format 9 DSCBs) or reside in the EAS. This is the default behavior for non-VSAM data sets in z/OS V1.11 and higher, if NO or OPT are not specified.

EATTR(OPT) Extended attributes are optional. The data set can have extended attributes and can optionally reside in the EAS. This is the default behavior for VSAM data sets in z/OS V1.11 and higher, if NO or OPT are not specified.

TSO/E services and commands

TSO/E services and commands now support EAV data sets in z/OS V1.11:

- Size calculations are based on how relative track addresses work.
- If OBTAIN is used, the EADSCB=OK keyword option is specified.
- TRANSMIT and ALLOCATE LIKE preserve EATTR settings.
- There are two new LISTDSI variables: SYSEATTR and SYSEADSCB.

Updated services

Here is a full list of services or commands that were updated. If a service or command is not listed, you should assume that they tolerate EATTR data sets as well.

- TSO/E I/O services
- SEND parmlib support routine
- ALLOCATE command processor
- EDIT command processor (APAR OA28295)
- SUBMIT I/O and control routine
- PRINTDS allocation/unallocation routine
- LISTDSI output routine
- LISTDSI data set information routine
- LISTDSI data management linkage assist routine
- RACONVRT utility I/O service routine
- TRANSMIT allocate input and read DSCB routine
- TRANSMIT unload a PDS with IEBCOPY routine

4.6.1 Usage and invocation

TSO/E commands or services work normally with EAV data sets:

- ALLOCATE DA(SAMPLE1) EATTR(OPT) NEW. DOC APAR OA28338 documents the new keyword.
- TRANSMIT/RECEIVE uses a new text unit that is transparent to users.
- The REXX and CLIST LISTDSI function sets have two new variables:
  - SYSEATTR

    Indicates the current status of the EATTR bits in the DSCB that describe the EAS eligibility status of a data set. The EAS can only contain data sets that are EAS-eligible. A blank indicates that the EATTR bits are '00'b. The defaults for EAS eligibility apply:

    - VSAM data sets are EAS-eligible, and can have extended attributes (Format 8 and Format 9 DSCBs).
    - Non-VSAM data sets are not EAS-eligible, and cannot have extended attributes (Format 8 and Format 9 DSCBs).

      Blank: This is the default setting. No EATTR is specified.

      NO: There are no extended attributes and the data sets cannot reside in EAS.

      OPT: The data sets can have extended attributes and reside in EAS.
  - SYSEADSCB

    Indicates whether the data set has extended attributes:

    - YES: The data set has extended attributes (Format 8 and Format 9 DSCBs) and can reside in the EAS.
    - NO: The data set does not have extended attributes (Format 8 and Format 9 DSCBs) and cannot reside in the EAS.

Sample REXX code for EATTR

Figure 4-34 shows a sample REXX exec that is used to display the EATTR parameter.

```
/* REXX*/
say 'Enter the parameters for the LISTDSI call'
pull dsname
X = LISTDSI(dsname)
SAY "SYSDSNAME              " SYSDSNAME
SAY "SYSEATTR               " SYSEATTR
SAY "SYSEADSCB              " SYSEADSCB
```

Figure 4-34 Sample REXX displaying EATTR

The output produced by the REXX code is shown Figure 4-35.

```
SYSDSNAME              IBMUSER.SAMPLE1
SYSEATTR               OPT
SYSEADSCB              YES
```

Figure 4-35 Output of the REXX exec

Sample CLIST for EATTR

Figure 4-36 shows a sample CLIST to display the EATTR parameter.

```
PROC 0
WRITE  ENTER THE PARAMETERS FOR THE LISTDSI CALL
READ
LISTDSI &SYSDVAL
WRITE  SYSDSNAME               &SYSDSNAME
WRITE  SYSEATTR                &SYSEATTR
WRITE  SYSEADSCB               &SYSEADSCB
```

Figure 4-36 Sample CLIST displaying EATTR

4.6.2 Migration and coexistence considerations

A new TRANSMIT/RECEIVE text unit should not cause exits to fail. Records with the INMEATTR key '8028'x can be ignored. New LISTDSI variables are not defined in earlier releases; if they are used in an earlier release, the new variables will not be set. For example, the sample REXX exec shown in Figure 4-34 on page 137 would return the output shown in Figure 4-37.

```
SYSDSNAME              IBMUSER.SAMPLE1
SYSEATTR               SYSEATTR
SYSEADSCB              SYSEADSCB
```

Figure 4-37 Output of the REXX exec prior to V1.11

4.7 FTP EAV large-volume access

z/OS V1.11 provides support for FTP for reading and writing extended format sequential data sets that have been allocated in the EAS of an EAV or are eligible for such an allocation (Format 8 DSCB).

FTP will read and write to such data sets in the EAS. FTP does not in R11 provide explicit configuration options to allocate data sets in the EAS or as eligible to reside in the EAS (unless implicitly done through SMS data class assignments).

FTP has added support to provide information about EAV volumes through the **qdisk** command.

If you "screen-scrape" qdisk output, note that you will have to modify the "screen-scraping" logic, because the **qdisk** output had to be changed between z/OS V1.10 and z/OS V1.11 to accommodate EAV volumes.

FTP adds support for reading to and writing from existing EAS data sets, but not creating them (toleration mode). FTP has the following support:

- FTP can understand Format 8 DSCBs.
- FTP can use TRKADDR for track calculations.
- The FTP qdisk option for the SITE/LOCSITE output format will change, as shown in Figure 4-38 on page 139.

```
ftp> quote site qdisk
200-        Percent      Free      Free          Largest    Free
200- Volume Free         Cyls      Trks         Cyls-Trks   Exts  Use Attr
200- CPDLB3   45         1507       108         1440    2     22  Storage
200- CPDLB0   44        80486       156          461    0     25  Storage
200- CPDLB1   99        66619         5        65362    5      3  Storage
200 SITE command was accepted
ftp>
```

Figure 4-38 Output of FTP command with EAV support

4.8 Service aids enhancements for EAV volumes

The first stage of EAV support was shipped in z/OS V1R10. This support allowed most VSAM objects to reside in cylinder-managed space on extended access volumes. The second stage of EAV support in z/OS V1R11 adds the ability to place extended format sequential data sets in cylinder-managed space. This support is largely implemented by DFSMS.

The goal of service aids support is to provide continuity of support, allowing extended format sequential data sets to be supported in both track-managed and cylinder-managed space to the extent that track-managed data sets have been supported previously. With z/OS V1R11, the following service aids components support EAV volumes:

- SPZAP
- SDUMP
- SNAP/ABDUMP
- Stand-alone DUMP
- CTRACE
- GTF
- AMATERSE
- IPCS

4.8.1 EAV support for SPZAP

SPZAP fully supports placement of SYSIN and SYSPRINT data sets in cylinder-managed space. SYSLIB references to extended format sequential data sets other than program libraries were not supported in prior releases and are not supported by z/OS V1R11 SPZAP.

4.8.2 EAV support for SDUMP

SDUMP fully supports placement of dump data sets in cylinder-managed space. This includes SYSMDUMPs and TDumps

4.8.3 EAV support for SNAP/ABDUMP

SNAP and ABDUMP fully support the placement of dump data sets in cylinder-managed space.

4.8.4 EAV support for CTRACE/GTF

CTRACE and GTF support placement of NOWRAP traces in cylinder-managed space. Only NOWRAP traces can be recorded to extended format data sets in earlier releases. CTRACE and GTF EAV1 support allowed both WRAP and NOWRAP traces to be written in cylinder managed space occupied by VSAM linear data sets. That support remains in place.

4.8.5 AMATERSE

AMATERSE fully supports the placement of dump data sets in cylinder-managed space.

4.8.6 IPCS

IPCS fully supports the placement of dump data sets in cylinder-managed space.

4.9 EAV coexistence maintenance

For systems sharing EAVs not at z/OS V1.11, there are coexistence support requirements. You should obtain the PSP ZOSV1R11 DFSMS and review it as part of your standard procedure when installing z/OS V1.11 and implementing DFSMS. More maintenance will be added as it becomes available. The following maintenance is for coexistence of EAVs with other sharing systems that are not at z/OS V1.11 DFSMS. Use the R10COEXS, R11COEXS, and EAV keywords to also search for coexistences and toleration maintenance.

The current coexistence and toleration APARs are:

- OA21113

  DVE support was packaged with IBM System Storage DS8000 R3 SPE for 3390A coexistence with PTFs for z/OS V1.9 back to V1.7.

- OA21487

  For z/OS V1.8 and V1.9 3390-A, there is support for “DEVSERV QDASD more cylinders”, “DEVSERV PATHS show 3390A”, and for the IDCAMS LISTDATA PINNED command.

- IO03548

  There is EREP formatting support for 3390-A.

- OA26623 OA27545

  There is IDCAMS new function toleration support. Additional EAS-Eligible data sets and a new EATTR data set attribute needs to be recognized in HDZ1A10.

- OA27441

  Converter/interpreter. Toleration support is provided in HBB7750 to recognize and ignore the EATTR JCL DD keyword for systems where EAV volumes are in a shared environment with a future release of z/OS.

- OA24505

  DFSMSdss 1K0, 180, and 190 are modified to restore data from a DFSMSdss dump data set containing data from an EAV.

- OA22900

  DFSMSdss 1K0, 180, and 190 are modified to restore data from a DFSMSdss dump data set containing data from an EAV.

- OA26839

  Support for all users of z/OS V1.10 sharing Extended Address Volumes with future releases beyond z/OS V1.10 and invoking VSAM partial space release.

- OA27146

  DFSMShsm Toleration for EAV R2. z/OS V1.10 and V1.9 levels of DFSMShsm need coexistence to handle the additional EAS-Eligible data sets and the new EATTR data set attribute.

- OA22804

  This APAR provides EAV coexistence support for pre-V1.10 DFSMShsm versions to restore EAS eligible data sets from a backup or migrated copy on an online volume and enable pre-V1.10 systems to process the new VCC keywords that might be specified in the Management Class Backup Copy.

4.10 Coexistence issues with pre-z/OS V1.10 systems

The APARs and PTFs discussed in this section are available for pre-z/OS V1.10 systems that do not allow an EAV volume to be online. If an EAV is not allowed to be online, it might be because of the following issues:

- Coexistence support is not supported.
- EAV volume use must be planned by applications.
- EAV volumes that are defined 'shared' must be set offline in HCD.

Attention: Use the R10COEXS, R11COEXS, and EAV keywords when researching coexistence and toleration maintenance.

The following items in this section have implications related to EAV volumes and older levels of z/OS that do not support them.

4.10.1 DEVSERV QDASD command

You can use the DEVSERV QDASD command to display diagnostic information about the status of direct access storage devices and storage control units. You can also use it to validate MVS storage resident control blocks for extended function status for the data acquired directly from the storage subsystem.

If you are using EAVs, install the appropriate coexistence PTF for APAR OA21487 on z/OS V1.9, if you have not already done so, so that the DEVSERV QDASD command will display the correct number of cylinders on z/OS V1.9. The DEVSERV command has the following changes:

- DEVSERV QDASD: The fields have been shifted to support a larger number of cylinders.
- DEVSERV PATHS: Support for 3390 Model A (3390A) has been added.

Figure 4-39 shows the DEVSERV QDASD command for an Extended Address Volume and shows the expanded positions for the CYL numbers (70119) for EAV volumes that were changed in z/OS V1.10.

```
DS QD,DC65,1
IEE459I 16.10.10 DEVSERV QDASD
 UNIT VOLSER SCUTYPE DEVTYPE       CYL  SSID SCU-SERIAL DEV-SERIAL EFC
0DC65 MLDC65 2107922 2107900     70119  89EC 0175-BALB1 0175-BALB1 *OK
****      1 DEVICE(S) MET THE SELECTION CRITERIA
****      0 DEVICE(S) FAILED EXTENDED FUNCTION CHECKING
```

Figure 4-39 DEVSERV QDASD command for an EAV volume

Note: In z/OS V1.11, or after the PTF for APAR OA25793 is installed on z/OS V1.10 and z/OS V1.9, DEVSERV PATHS and the QDASD command display five digit device numbers, 0DC65, as shown in Figure 4-39 on page 142, that consist of the subchannel set number and the normal device number. The blank that was in the first character position no longer exists and is replaced by 0 or 1 (subchannel set number).

DEVSERV SMS command

You can use the DEVSERV SMS or DEVSERV S command to display the volume and storage group status of devices that are either non-SMS managed (see Figure 4-40) or SMS managed (see Figure 4-41). Both these figures show that the 3390A device type is recognized and displayed.

```
DEVSERV SMS,DC65
IGD001I 15:16:18 DEVSERV SMS 720
  UNIT DTYPE  M VOLSER VOLSTAT      STORGRP  SGSTAT
  DC65,3390A ,O,MLDC65,PRIV/RSDNT,  VOLUME NOT MANAGED BY SMS
 ****************************** LEGEND ******************************
 A = ALLOCATED                      F = OFFLINE
 M = MOUNT PENDING                  N = NOT ALLOCATED
 O = ONLINE                         P = PENDING OFFLINE
 Note: Tape device type is not supported by DEVSERV SMS
```

Figure 4-40 DEVSERV command to display volume and storage group status

```
DS SMS,DD65
IGD001I 13:23:09 DEVSERV SMS 029
 UNIT DTYPE  M VOLSER VOLSTAT      STORGRP  SGSTAT
 DD65,3390A ,O,GKDD65,ENABLED      EAVGK   ,ENABLED
****************************** LEGEND ******************************
A = ALLOCATED                       F = OFFLINE
M = MOUNT PENDING                   N = NOT ALLOCATED
O = ONLINE                          P = PENDING OFFLINE
Note: Tape device type is not supported by DEVSERV SMS
```

Figure 4-41 DERVSERV SMS displaying SMS managed device and volume

DEVSERV PATH command

The DEVSERV PATH command displays the new 3390A device type, as shown in Figure 4-42.

```
DS P,DD65
IEE459I 12.59.50 DEVSERV PATHS 990
 UNIT DTYPE  M CNT VOLSER  CHPID=PATH STATUS
      RTYPE  SSID CFW TC   DFW PIN DC-STATE CCA DDC       CYL CU-TY
0DD65,3390A ,0,000,GKDD65,8D=+ 8F=+ 9F=+ A1=+
      2107   89ED  Y  YY.  YY.  N  SIMPLEX   65  65    120204 2107
*********************** SYMBOL DEFINITIONS ***********************
O = ONLINE                          + = PATH AVAILABLE
```

Figure 4-42 DEVSERV PATH command displays a new 3390A device type

The following APARs and PTFs must be installed on downlevel releases in support of the new functions available for Extended Address Volumes.

3390 Model A support

3390 Model A support was included in IBM System Storage DS8000 release 3 SPE APAR. This release affected one operator command response of the DEVSERV command, which displays the following items:

- It allowed 3390A to be displayed instead of 3390 in the DESVERV PATHS command response. This support exists for z/OS V1.7 and later. If you do not have this support, 3390 will be displayed.
- PTFs for OA21216 are available:
  - UA38011 5695DF1111K0 (z/OS V1.7)
  - UA38012 5695DF111180 (z/OS V1.8)
  - UA38013 5695DF111190 (z/OS V1.9)

4.10.2 LISTDATA PINNED command

The track addresses in the output of the IDCAMS LISTDATA PINNED command are in a different format, regardless of the volume size. Before z/OS V1.10, the report listed each pinned track and its associated data set name.

Beginning with z/OS V1.10, the report identifies a range of pinned tracks associated with a data set. Each track address is printed in eight hex digits in the native format of the device, using the form CCCCcccH. In addition, instead of one track being listed per line, consecutive tracks for each data set are gathered into one line and the range of track addresses is shown.

Programs that use DASD channel programs and messages 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. Use the new TRKADDR macro for all track address comparisons and calculations. This macro is available in z/OS V1.10, 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 V1.10 and later, 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 the IDCAMS LISTDATA PINNED command 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 CCCCcccH format. The modified report format applies to all volumes, not only EAVs.

LISTDATA PINNED with UNITNUMBER

This command has the following restrictions:

- LISTDATA PINNED with UNITNUMBER requires APAR OA21487.
- LISTDATA PINNED with UNITNUMBER should not be issued on pre-z/OS V1.10 systems to an EAV volume when the storage subsystems has pinned data. In this case, the LISTDATA PINNED command should be issued on a z/OS V1.10 system.
- UA40229 for z/OS V1.9 is not allowed.
- UA40228 for z/OS V1.8 is not allowed.

4.10.3 LSPACE macro

The LSPACE macro change when you install APAR OA22449. This APAR adds support to DFSMS z/OS V1.7, V1.8, and V1.9 to allow these releases to map V1.10 LSPACE parameters XEXPMSG and EXPDATA to EXPMSG and DATA respectively by:

- Treating expanded requests as normal and changing pre-V1.10 LSPACE systems to accept the extended parameter list and not fail it with return code 12.
- Changing EXPDATA and XEXPMSG to DATA and EXPMSG. This changes pre-V1.10 LSPACE systems to treat an EXPDATA= or XEXPMSG= requests as DATA= or EXPMSG=, respectively. The following PTFs are available:
  - z/OS V1.8 UA40220
  - z/OS V1 R9 UA40221

4.10.4 ICKDSF support

The IDCAMS LISTDATA PINNED and DEVSERV QDASD commands needs ICKDSF R17 support for Extended Address Volumes: PK56092 to recognize EAV volumes under z/OS R1V8.0 and R1V9.0.

4.10.5 EREP V3R5 support

EREP V3R5 support for Extended Address Volumes: IO03548 provides formatting support in detailed edit and System Exception reports for EAV devices, Model A.

4.10.6 DFSMShsm support

The pre-V1.10 versions of z/OS DFSMShsm can coexist with z/OS DFSMShsm V1.10 in a HSMplex. In this situation, the EAV is inaccessible for the pre-V1.10 systems because they will be offline. Data sets that have been migrated or backed up from an EAV to ML1, ML2, and backup volumes (non-EAV) under z/OS DFSMShsm V1.10 will be accessible for the pre-V1.10 DFSMShsm systems. This APAR is required to support recall, recovery, and recycling for these data sets in the pre-V1.10 systems. If the APAR is not installed, the recall, recovery, and recycling of these data sets will fail during pre-V1.10 DFSMShsm processing. AUDIT and RECYCLE are changed to process a data set CDDs with a Format 8 DSCB.

DFSMShsm APAR OA22804 provides coexistence support of the pre-V1.10 DFSMShsm versions to:

- Allow pre-V1.10 systems to recognize MCV records for EAVs.
- Restrict the selection of extent reduction requests from the CRQ, if a request is directed to an EAV.
- Restore EAS eligible data sets from a backup or migrated copy on an online volume.
- Enable pre-V1.10 systems to process the new VCC keywords that may be specified in the management class backup copy technique field.
- Support recall and recovery on V1.9 or lower of an EAS data set with Format 8 DSCB that was migrated or backed up on V1.10:
  - Converts to Format 1 DSCBs.
  - Issues the new ARC0784I message and updates the FSR record if Format 9 DSCB vendor attributes are lost during the RECALL/RECOVER/ARECOVER function:

    ```
    ARC0784I EXTENDED ATTRIBUTES FOR DATA SET dsname WERE NOT RETAINED DURING
    THE RECALL | RECOVER | ARECOVER
    ```

    Explanation: The data set was recalled, recovered, or arecovered from a migration copy, backup copy or an aggregate backup successfully. However, JOBNAME, STEPNAME, creation time attributes, and vendor attributes from the Format 9 DSCB of the recalled or recovered data set were not retained, because the volume on which it was placed did not support Format 8 and Format 9 DSCBs. The recall, recovery, or aggregate recovery continues.

Coexistence support

Coexistence is added for pre-V1.10 systems to understand the new MCV records for EAV volumes. These fields will be ported from V1.10 systems to the pre-V1.10 systems.

Consideration: If a recall for extent reduction is directed to an EAV, the selection of this request from the CRQ will be restricted in the pre-V1.10 systems.

Use of EAV volumes in mixed environments where the volume cannot be brought online to all systems can also result in problems in recalling or recovering data sets that belong to a Storage Class having the guaranteed space attribute.

Recommendations: If you need to recover a data set backup made from a SMS EAV volume to a lower level system, then, prior to the recovery, ensure that you perform the following actions:

- If the EAV is still available on the higher level system, delete or rename the data set there.
- If the EAV is no longer available and a catalog entry exists, then use IDCAMS DELETE NOSCRATCH to remove the data set from the catalog.

The backup copy must reside on a non-EAV and the volume must be online to the lower level system.

Coexistence considerations

Without APAR OA22804 installed, DFSMShsm is impacted if it is actually managing EAV volumes on a V1.10 system:

- On a pre-z/OS V1.10 system, if an attempt is being made to recall or recover an EAS data set with a Format 8 that was migrated or backed up on z/OS V1.10, the function fails.
- This directly impacts recall, which is done implicitly. It also impacts recycling and auditing.
- VCC keywords specified in the management class are processed correctly. If VPREF or VREQ is specified in the management class, data set is backed up using a standard I/O method without a coexistence APAR.

Important: Apply PTFs UA40303 for (V1.6), UA40304 for (V1.7), UA40305 for (V1.8), and UA40306 for (V1.9).

DFSMShsm and coexistence scenarios and considerations are discussed further in 5.4, “DFSMShsm and Extended Address Volume (EAV)” on page 205. Specifically, migration and recall, backup and restore, and abackup and arestore are considered.

4.10.7 DFSMSdss restores from EAV volumes

DFSMSdss releases are modified to restore data from a DFSMSdss dump data set containing data from an Extended Address Volume (EAV) in the following manner:

- Restore of data sets dumped from an EAV with physical or logical data set is supported.
- Track restore of data dumped from the track-managed space of an EAV using physical dump processing is supported.
- When a data set that was dumped from an EAV that has Format 8 and Format 9 DSCBs is restored with physical or logical data set processing and vendor attributes existed in the Format 9 DSCB, a MSGADR556W is issued with reason code 1:

```
ADR556W (ttt)-,mmmm(yy), EXTENDED ATTRIBUTES FOR DATA SET dsname WERE NOT
    RETAINED DURING {COPY | RESTORE},rsn
```

  - Explanation: The data set was copied or restored, but some extended attributes were lost for the following reasons (rsn):
    - Vendor attributes from the Format 9 DSCB of the dumped data set were not retained for the target data set because the volume on which it was placed did not support Format 8 and Format 9 DSCBs.

- Vendor attributes from the Format 9 DSCB of the dumped or copied data set were not retained for the target data set due to problems updating the target's Format 9 DSCB.

– Programmer Response:

- If the extended attributes are desired, rerun the COPY or RESTORE and specify target volumes or an SMS group that supports Format 8 and Format 9 DSCBs.
- An error occurred while reading or writing an F9 DSCB. Retry the restore or copy for the data set.

► Full volume restore to a non-EAV will fail because the EAV is larger than a non-EAV volume. MSGADR309E is issued:

```
ADR309E (ttt)-mmmmm(yy), SOURCE AND TARGET DEVICE CAPACITIES DO NOT MATCH.
CYLINDER CAPACITY OF SOURCE VOLUME nnnnnnn, TARGET VOLUME nnnnnnn
```

– Explanation: The capacity of the source volume is greater than that of the target volume. The nnnn is the highest cylinder number in hexadecimal.

– System action: The task ends. Processing continues with the next control statement. The return code is 8.

► For a physical track restore, an attempt to process tracks dumped from the cylinder-managed space of an EAV fails. One of the following messages is issued:

```
MSGADR024E (issued when the track address specified is within the range of
65,520 and 65535)
```

or

```
ADR024E (ttt)-mmmmm(yy), TRACKS/OUTTRACKS VALUE IS INVALID FOR DEVICE.
    VALID CYLINDER RANGE IS 0 TO X'ccccccc'. VALID TRACK RANGE IS 0 TO X'h'
```

– Explanation: The following values are not valid for the specified device types:

- The range specified in the TRACKS keyword of a DUMP, COPY, RESTORE, or PRINT.
- The range specified for the OUTTRACKS keyword for COPY or RESTORE. The TRACKS values must fall within the cylinder and track range indicated in the message. The cccc and hhhh are high cylinder and head numbers, respectively, in decimal notation.

– System Action: The task ends. Processing continues with the next control statement. The return code is 8.

- Application Programmer Response: Correct the error, and rerun the job.

► MSGADR136E is issued when the track address specified is great than 65535:

```
ADR136E (ttt)-mmmmm(yy), CONSTANT 'xxxx' IS NOT WITHIN VALUE RANGE
```

– Explanation: The value of the constant (xxxx) is outside the range of values allowed for the associated parameter.

– System Action: The task is not performed. Processing continues with the next control statement. The return code is 8.

– Application Programmer Response: Check the command syntax for allowed values, correct the error, and then reissue the command.

DFSMSdss considerations

When you use DFSMSdss to back up data sets with the extended attributes variable DS1EATTR set in the VTOC, you must make sure that these data sets can be restored into an environment that supports them. If the vendor attributes in the Format 9 DSCB are essential to the validity of the data set, make sure that the environment on which they might be restored has EAV volumes to support these Format 9 fields.

When you use DFSMSdss restore with APAR OA22900, consider the following items:

- The DFSMSdss restore functions and limitations that are supported via the coexistence APAR for pre-z/OS V1.10 systems 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 is changed to convert Format 8 and Format 9 DSCBs to Format 1 DSCBs.
- When a data set is restored that had a Format 8 and Format 9 pair when it was dumped, if attributes are being lost due to the inability to restore the Format 9, a new message, ADR556W, is issued. If no attribute values exist, no message is issued.

  ```
  ADR556W (ttt)-,mmmm(yy), EXTENDED ATTRIBUTES FOR DATA SET dsname WERE NOT
  RETAINED DURING {COPY | RESTORE},rsn
  ```

  Explanation: The data set was copied or restored but some extended attributes were lost for the following reasons (rsn), where rsn is 1 or 2:

  1. Vendor attributes from the Format 9 DSCB of the dumped data set were not retained for the target data set because the volume on which it was placed did not support Format 8 and Format 9 DSCBs.
  2. Vendor attributes from the Format 9 DSCB of the dumped or copied data set were not retained for the target data set due to problems updating the target's Format 9 DSCB.

- Additional reference information about restore toleration is given in the following list:
  - DFSMSdss full volume and track dumps of EAVs are not compatible with dumps of volumes that consist of 65,520 cylinders or fewer due to the changes required to format the extended addressing space in the dump. Changes have been made in z/OS V1.10 DFSMSdss to identify dumps of EAVs.
  - It is possible to run more than one level of z/OS in an environment when you must provide data in the form of DFSMSdss dumps to other environments that are not at the same z/OS level. DFSMSdss provides limited restore capability on supported lower levels of z/OS for data dumped from an EAV on z/OS V1.10 or higher levels.
- Coexistence PTF number: UA40263 for V1.6, UA40264 for V1.7, UA40265 for V1.8, and UA40266 for V1.9 describe the restore functions and limitations that are supported via a coexistence APAR on a pre-z/OS V1.10 system.
- The latest coexistence APAR is OA26996, which provides coexistence to z/OS V1.9 and V1.10 for the restore of extended format sequential (EFSAM) data sets. An EFSAM data set can be restored to a non-EAV volume, and the EATTR specification is still kept.

4.10.8 DB2 support for EAVs

The following items are DB2 V8 and DB2 V9 considerations for EAV volumes:

- Admin enablement and log manager

  As long as BSDS and active logs are kept on non-EAV volumes and are not using admin enablement functions, DB2 V8 and DB2 V9 should be run without any new DB2 EAV support code. Otherwise, the coexistence support to DB2 V8 and DB2 V9 needs to be applied, which adds support to allow DB2 BSDS(s) and active logs to reside anywhere on an EAV volume with APAR PK58292.

  Therefore, DB2 admin enablement for EAV requires DB2 users to apply the appropriate coexistence PTFs before placing any BSDS or active logs on Extended Address Volumes, or before using admin enablement functions on EAVs.

- DB2 log manager changes for EAVs are not needed as long as BSDS and active logs are kept on non-EAV volumes and are not using admin enablement functions. DB2 V8 and V9 should be run without any new DB2 EAV support code.

  However, DB2 users need to apply the appropriate coexistence PTFs before placing any BSDS or active logs on Extended Address Volumes, or before using admin enablement functions on EAVs. These PTFs are APAR PK61105 for DB2 V8 and V9.

4.11 Coexistence issues with z/OS V1.10

Data set types with extended attributes are not supported in z/OS V1.10. However, consider the following items:

- Coexistence support provided in z/OS V1.10 allows data sets with extended attributes to be processed by DADSM for functions such as scratch, rename, and partial release.

  As long as no attempt is made to open the data set itself (that is, if it is not supported for OPEN and I/O processing on V1.10, as is the case with extended format sequential data sets).

  Tip: Use system affinity to ensure jobs run on the correct system in a sysplex that has systems at both z/OS V1.11 and V1.10. For example:

  ```
  /*JOBPARM SYSAFF=SC70
  ```

- Coexistence support provided in z/OS V1.10 allows straddled extents that could be created for VSAM data sets in z/OS V1.11 to be processed. A VSAM data set allocated in z/OS V1.11 with straddled extents should be able to be processed from z/OS V1.10 with this coexistence support (OA26623) so that it could be referenced, extended, partially released, and scratched. In addition the DADSM convert routines were changed to tolerate and validate straddled extents.
- Extents that start in track-managed space and end in cylinder-managed space are tolerated.
- Coexistence support for the EATTR data set attribute is provided in DADSM, VSAM, and VSAM RLS functions to honor EATTR=NO and its other values for VSAM files allocated in z/OS V1.11, but it is processed in z/OS V1.10. For VSAM and VSAM RLS files, this support allows an extend to a new volume to see the passed EATTR value. DADSM create functions in turn record the passed EATTR value in the Format 1 and Format 8 DSCBs and search for a space by way of the EATTR attribute value.

- Application programs issuing CVAF and OBTAIN macros with the EADSCB=OK keyword are able to see Format 8 and Format 9 DSCBs for types of data sets that cannot be opened on z/OS V1.10. (Status code 84 will no longer be issued.)
  - CVAF and OBTAIN macros have been changed to return DSCB information for data sets that have extended attributes, but are not supported in z/OS V1.10.
  - Removing CVAF status stat084 errors in the OBTAIN, CVAFSEQ, CVAFDIR, and CVAFFILT macros allow data set list builders, such as ISPF, IEHLIST LISTVTOC, and all others that use these services, to display and see DSCBs with extended attributes as long as the EADSCB=OK keyword has been specified on the request. In other words, what can be seen on z/OS V1.11 is what can be seen on z/OS V1.10.
  - CVAF status code stat084 was set for a data set with extended attribute DSCBs where the data set type described by this DSCB was not supported in the release for EAS. This restriction has been removed to facilitate support for non-VSAM data sets. This indirectly affects the OBTAIN macro. Issuers of OBTAIN or a CVAF macro with EADSCB=OK might see a Format 8 DSCB for a type of data set that cannot be opened in the current release. This allows data set list builders to see non-EAS eligible data sets.
- Coexistence support is provided for DFSMSdss in z/OS V1.10 in order to detect extended-format sequential data sets in the EAS and fail them because they are not supported in the release, restore extended-format sequential data sets properly, to handle an extent that resides in both the non-EAS and EAS, and to properly handle the updating and preservation of Format 9 DSCB vendor and internal fields.
- Coexistence support is provided for DFSMShsm in z/OS V1.10 to enable DFSMShsm to tolerate the DFSMShsm support provided in z/OS V1.11. In z/OS V1.10, DFSMShsm can recall, recover, and arecover both VSAM and non-VSAM data sets from a z/OS V1.11 migration or backup copy with Format 8 and Format 9 DSCBs to non-EAVs. Format 8 and Format 9 DSCBs will be converted to Format 1 DSCBs and Format 9 DSCB attributes will be lost if such a data set is recalled, recovered, or arecovered to non-EAV. In this case, the ARC0784I message will be issued if DFSMSdss issues an ADR556W message. For z/OS V1.9, support is added for preserving the EATTR value.
- Coexistence support is provided in Access Method Services (AMS) on z/OS V1.10 to recognize the EATTR value of an import of a portable data set created from z/OS V1.11 and to support passing the EATTR value on an z/OS V1.10 export. Having the EATTR value propagated to the portable data set allows a subsequent import of this portable data set to an z/OS V1.10 or higher release to support the EATTR value. Releases prior to z/OS V1.10 will not support EATTR on an import or export, so the EATTR value will be not specified for an imported data set and in the exported portable data set.

z/OS BCP allocation support is changed in z/OS V1.10 to recognize and ignore the EATTR attribute. This is an internal use only change.

Coexistence APARs

Coexistence considerations for the EATTR data set attribute in z/OS V1.10 are provided with APAR numbers OA26623, OA27069, OA27578, OA27577, OA27286, OA27270, OA27441, OA27545, OA26996, OA27146, and OA27465.

Attention: Use the R10COEXS, R11COEXS, and EAV keywords when searching for coexistence and toleration maintenance.

4.12 Diagnosing EAV problems

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 the 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

  See message IEC144I 313-0C. An attempt was made to OPEN a data set that had a Format 8 DSCB, but it is not eligible for extents residing above cylinder 65,520.

- 113-44

  See message IEC142I 113-44. An OPEN of a data set failed on an EAV and the DCBE flag DCBEEADSCBOK was not set, indicating that the caller understands 28-bit cylinder numbers.

- 113-48

  See message IEC142I 13-48. An OPEN of a VTOC failed on an EAV and the DCBE flag DCBEEADSCBOK was not set, indicating that the caller understands Format 8 and 9 DSCBs.

Return code 4, CVSTAT STAT082 (X'52') indicates that a CVAF operation (CVAFDIR, CVAFSEQ, or CVAFDSM) 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'), indicates that a CVAFFILT operation failed due to the caller not specifying support for extended attribute DSCBs (with EADSCB=OK). 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 the caller not specifying support for extended attribute DSCBs (EADSCB=OK):

- 24(X'18')

  The SEEK option (reading by absolute device address) was specified. OBTAIN was issued to an 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 Format 9 DSCB.

- 24(X'18')

  The SEARCH option (reading by data set name) was specified. OBTAIN was issued to an 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.

For information about the 313 ABEND, see the description of message IEC144I in *z/OS MVS System Messages, Volume 7 (IEB-IEE)*, SA22-7637. For information about abends 113-44 and 113-48, see the explanation for message IEC142I in *z/OS DFSMSdfp Advanced Services*, SC26-7400. For information about the return codes, see *z/OS DFSMSdfp Advanced Services*, SC26-7400 as well.

5

DFSMShsm enhancements

In this chapter, we describe the following DFSMShsm topics:

- DFSMShsm data set backup retention period
- DFSMShsm ML1 enhancements
- DFSMShsm Fast Replication enhancements
- DFSMShsm and EAV

5.1 Data set backup retention period enhancements

Currently, there is no efficient method to allow users to keep individual data set backup versions for a defined period. For SMS managed data sets, the number of data set backup versions and their retentions are determined by their Management Class as well as their backup frequency. For non-SMS managed data sets, the SETSYS VERSIONS command dictates the default values for frequency and number of versions. The (H)ALTERDS command allows for exceptions to the default values for individual data sets.

In z/OS V1.11, a new keyword, RETAINDAYS, is introduced for the DFSMShsm (H)BACKDS command. It allows users to specify the number of days a backup version will be retained. This RETAINDAYS value is applied when a backup copy is rolled off and during EXPIREBV processing.

With this new feature, DFSMShsm backup copies will be managed as either active copies or retained copies:

- Active copies is a set of backup copies that have not yet been rolled off. They are determined by either SMS Management Class or the SETSYS value. The maximum number of active copies is still 100.
- Retained copies are backup copies that have rolled off from an active copies set and have not yet reached their retention period. A data set can have an unlimited number of retained backup copies.

5.1.1 Roll off processing

Roll off processing occurs when the number of existing active copies exceeds the maximum number of active backup versions defined. Roll off processing is modified to check all of the active copies to see if any active copy has RETAINDAYS specified and if the retained days have been met:

- If more than one of the active backup copies have met their retention days, the oldest active backup copy will be rolled off, and the rest of the active backup copies will be maintained as active copies even though their retention days have already been met.
- If one or more of the active backup copies have retention days specified, but none of them have met their retention days, then:
  - If the oldest active backup copy has a RETAINDAYS value, then it will be rolled off and replaced by the new backup copy, but it is still maintained as a retained backup copy.
  - If the oldest active backup copy does not have a RETAINDAYS value, then it will be rolled off as normal.
- If none of the active backup copies have RETAINDAYS values, roll off processing will be processed as normal.

Roll off example

The following example illustrates how the roll off process works:

In Example 5-1 on page 155, we have run DFSMShsm command HLIST DSNAME(MHLRES4.JCL.CNTL) BCDS TERMINAL. Data set MHLRES4.JCL.CNTL has five backup versions and has a maximum number of active copies (#BACK UP (DS EXIST) field in ISMF) of five.

The LIST date is 2009/07/24. Prior to creating the new backup version, the oldest backup version (VER=002) has exceeded its RETAINDAYS value (RETDAYS=00001), as shown in Example 5-1.

Example 5-1 Output of DFSMShsm LIST command prior to creating a new backup version

```
DSN=MHLRES4.JCL.CNTL                                  BACK FREQ = ***  MAX ACTIVE B
BACKUP VERSIONS = ***

  BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=000  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T481220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:12:48  CAT=YES  GEN=001  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T271220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:12:27  CAT=YES  GEN=002  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

  BDSN=HSM.BACK.T481120.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:11:48  CAT=YES  GEN=003  VER=003  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T321120.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:11:32  CAT=YES  GEN=004  VER=002  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00001

 TOTAL BACKUP VERSIONS = 0000000005

 ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000041 SENT TO DFSMSHSM
 ***
```

After creating a new backup version (VER=007), we refer to Example 5-2, which shows that backup copy version 002 (VER=002) has been rolled off and is no longer part of the output.

Example 5-2 Output of DFSMShsm LIST command after a new backup version has been created

```
DSN=MHLRES4.JCL.CNTL                                  BACK FREQ = ***  MAX ACTIVE B
BACKUP VERSIONS = ***

  BDSN=HSM.BACK.T052712.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/24  BACKTIME=12:27:05  CAT=YES  GEN=000  VER=007  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=001  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T481220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:12:48  CAT=YES  GEN=002  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T271220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
```

```
  BACKDATE=09/07/23  BACKTIME=20:12:27  CAT=YES  GEN=003  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

  BDSN=HSM.BACK.T481120.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:11:48  CAT=YES  GEN=004  VER=003  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

TOTAL BACKUP VERSIONS = 0000000005

ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
COMMAND REQUEST 00000045 SENT TO DFSMSHSM
***
```

The next new backup version (VER=008) contains backup copy version 3 (VER=003), which has no RETAINDAYS value, so it will be rolled off, as shown in Example 5-3.

Example 5-3 Output of DFSMShsm LIST command

```
DSN=MHLRES4.JCL.CNTL                                BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T211613.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/24  BACKTIME=13:16:21  CAT=YES  GEN=000  VER=008  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T052712.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/24  BACKTIME=12:27:05  CAT=YES  GEN=001  VER=007  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=002  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T481220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:12:48  CAT=YES  GEN=003  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T271220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:12:27  CAT=YES  GEN=004  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

 TOTAL BACKUP VERSIONS = 0000000005

 ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000048 SENT TO DFSMSHSM
 ***
```

At this point, the oldest active copy is VER=004 and its RETAINDAYS (RETDAYS=00005) value has not been met. After creating a new backup version, VER=004 is rolled off but is kept as a RETAINED COPIES version. Notice that this rolled off backup version has no GEN and VER values, as showed in Example 5-4. Also notice that the TOTAL BACKUP VERSION is 6.

Example 5-4 Output of DFSMShsm LIST command

```
DSN=MHLRES4.JCL.CNTL                                BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***
```

```
 BDSN=HSM.BACK.T292413.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/24  BACKTIME=13:24:30  CAT=YES  GEN=000  VER=009  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 BDSN=HSM.BACK.T211613.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/24  BACKTIME=13:16:21  CAT=YES  GEN=001  VER=008  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 BDSN=HSM.BACK.T052712.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/24  BACKTIME=12:27:05  CAT=YES  GEN=002  VER=007  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=003  VER=006  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

 BDSN=HSM.BACK.T481220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/23  BACKTIME=20:12:48  CAT=YES  GEN=004  VER=005  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 BDSN=HSM.BACK.T271220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/23  BACKTIME=20:12:27  CAT=YES  GEN=***  VER=***  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

TOTAL BACKUP VERSIONS = 0000000006

ARC0140I LIST COMPLETED,       14 LINE(S) OF DATA OUTPUT
COMMAND REQUEST 00000060 SENT TO DFSMSHSM
***
```

A new parameter named SELECT(ACTIVE/RETAINDAYS) has been added to the DFSMShsm (H)LIST command in order to list either active copies or retained copies only:

```
(H)LIST DSNAME(dsn) BOTH | BCDS SELECT(ACTIVE | RETAINDAYS)
```

An output of LIST DSNAME(MHLRES4.JCL.CNTL) BCDS SELECT(ACTIVE) TERMINAL is shown in Example 5-5.

Example 5-5 Output of DFSMShsm LIST active copies

```
DSN=MHLRES4.JCL.CNTL                                BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

 BDSN=HSM.BACK.T292413.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/24  BACKTIME=13:24:30  CAT=YES  GEN=000  VER=009  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 BDSN=HSM.BACK.T211613.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/24  BACKTIME=13:16:21  CAT=YES  GEN=001  VER=008  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 BDSN=HSM.BACK.T052712.MHLRES4.JCL.J9205          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/24  BACKTIME=12:27:05  CAT=YES  GEN=002  VER=007  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****
```

```
  BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=003  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T481220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:12:48  CAT=YES  GEN=004  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 TOTAL BACKUP VERSIONS = 0000000005

 ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000071 SENT TO DFSMSHSM
 ***
```

An output of LIST DSNAME(MHLRES4.JCL.CNTL) BCDS SELECT(RETAINDAYS) TERMINAL is shown in Example 5-6.

Example 5-6 Output of DFSMShsm LIST retained copies

```
DSN=MHLRES4.JCL.CNTL                                BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=000  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T271220.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:12:27  CAT=YES  GEN=***  VER=***  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

 TOTAL BACKUP VERSIONS = 0000000002

 ARC0140I LIST COMPLETED,        6 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000072 SENT TO DFSMSHSM
 ***
```

Guaranteed Backup Frequency and RETAINDAYS

Guaranteed Backup Frequency replacement is one of the ways to delete unwanted backup versions. Guaranteed Backup Frequency is defined as the most recent backup version being replaced with the version being made because the data set has not changed since the most recent backup was made. This function takes place only when the data set is being backed up by having met the Guaranteed Backup Frequency criteria defined in its associated SMS Storage Group attribute. Guaranteed Backup Frequency replacement is not involved in the backup of individual data sets through the (H)BACKDS command. The Guaranteed Backup Frequency replacement, however, will have effect on the most recent backup version of a specific data set.

Guaranteed Backup Frequency and RETAINDAYS value

If the version being replaced has a RETAINDAYS value specified at the time of backup and has not met that value, the RETAINDAYS value will be recalculated based upon today's date and copied to the newly created version.

5.1.2 EXPIREBV

During the EXPIREBV process, the retention days of the backup copies, including active and retained copies, are checked. Any copy that has met its retention day will be expired.

Expiring active copies

The logic for expiring SMS managed and non-SMS managed data sets are different.

SMS managed data sets

The EXPIREBV process will check the retention days for each active backup copy for each data set as follows:

- Starting with the oldest backup version and ending with the third newest version. If the version has a specified retention days value, DFSMShsm calculates the age of the version and compares the age to the value of the retention days. The second-newest version is treated as though it had been created on the same day as the newest backup version. Therefore, the second-newest version will not be expired unless the number of retention days specified by RETAINDAYS have passed since the creation of the newest backup version. The newest backup version will not be processed unless both the Management Class retention values and RETAINDAYS value have been met.
- The rest of EXPIREBV processing for SMS managed data sets will skip any backup copy that has RETAINDAYS specified.

Non-SMS managed data sets

For non-SMS data sets, the number of retention days of a backup version will be considered in addition to any other expiring criteria, such as DELETEDIFBACKUP(DAYS), CATALOGEDDATA(DAYS), and UNCATALOGEDDATA(DAYS). If retention days were specified during data set backup, DFSMShsm calculates the number of days since the backup copy creation date and compares the result to the retention days in the backup version record. If the time since the backup version was created is more than the retention days specified, then the backup version is deleted. This process includes both the active backup copies and the retained backup copies. DFSMShsm deletes all backup versions that have met the specified retention days, regardless of what parameters (DELETEDIFBACKUP, CATALOGEDDATA, or UNCATALOGEDDATA) are provided with the NONSMSVERSION keyword for the EXPIREBV command. If the retention days of the backup version are not met and the other additional EXPIREBV criteria are met, EXPIREBV processing will skip all backup copies that have not met their retention days, regardless of the existing EXPIREBV criteria. If RETAINDAYS was not specified for the backup copy, then only the existing EXPIREBV criteria are applied.

Expiring retained copies

EXPIREBV processing for non-SMS managed data sets and SMS managed data sets is the same. The process will check retention days for each retained backup copy. The retained copy will be identified as an expired version if the retained copy has met its retention period and, therefore, will be deleted during EXPIREBV execute processing.

5.1.3 Commands related to the RETAINDAYS syntax

As a result of this new data set backup retention period feature, the following DFSMShsm commands now have new parameters to accommodate this feature.

BACKDS command syntax

Figure 5-1 shows the BACKDS command syntax.

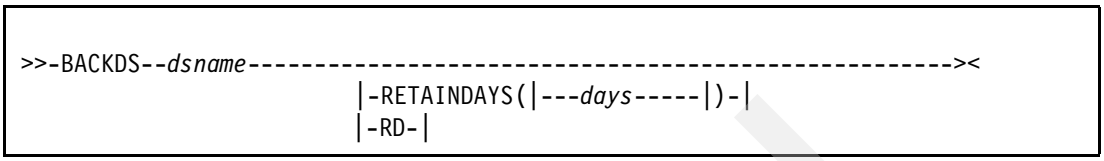


Figure 5-1 BACKDS command syntax

RETAINDAYS is a new, optional keyword. If RETAINDAYS is specified for the command, the number of retention days is a required parameter that must be an integer number in the range of 0 to 50000. Any retention days of 99999 will be managed as a “never expire” value. A retention days value of 0 will indicate that the data set backup version:

1. Could be expired within the same day that it was created if the maximum number of data set backup copies is reached
2. Will be kept as an active copy before roll off occurs
3. Will never be managed as a retained copy

Example 5-7 shows the BACKDS command with the RETAINDAYS value being run against MHLRES.JCL.CNTL.

Example 5-7 Sample BACKDS command

```
BACKDS MHLRES4.JCL.CNTL RD(15)
```

The ARC1605I message will be generated if you enter an invalid RETAINDAYS value, as shown in Example 5-8.

Example 5-8 An example of ARC1605I when an invalid RETAINDAYS value is entered

```
ARC1001I BACKDS MHLRES4.INIT.D119 RD(50001) COMMAND FAILED, RC=0005, REAS=0042
ARC1605I COMMAND HAD PARSE ERROR
COMMAND REQUEST 00000124 SENT TO DFSMSHSM
***
```

RETAINDAYS is only allowed with Cataloged Data Sets. An ARC1378I error message issued with uncataloged data sets, as shown in Example 5-9.

Example 5-9 Sample output of ARC1378I

```
ARC1001I MHLRES4.INIT.D119  BACKDS FAILED, RC=0078, REAS=0004
ARC1378I UNCATALOGUED DATA SET COULD NOT BE PROCESSED FOR BACKUP
COMMAND REQUEST 00000217 SENT TO DFSMSHSM
***
```

HBACKDS command syntax

Figure 5-2 shows the HBACKDS command syntax.

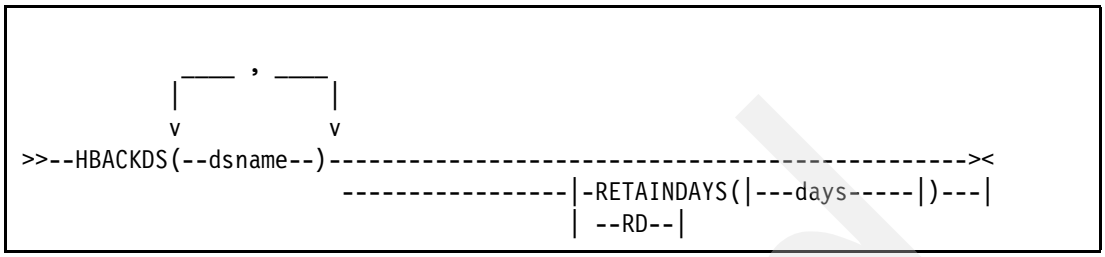


Figure 5-2 HBACKDS command syntax

The HBACKDS RETAINDAYS command syntax is enhanced to enable RETAINDAYS(days) to be specified for one or multiple data sets. If multiple data sets are specified, the RETAINDAYS value applies to all specified data sets. If a partially qualified data set is specified, the RETAINDAYS value applies to all of the data sets that meet the filter criteria. The same range checking under the BACKDS command also applies to HBACKDS.

ARCHBACK command syntax

Figure 5-3 shows the ARCHBACK command syntax.

Figure 5-3 ARCHBACK command syntax

DSN specifies the address of a field that contains the fully qualified name of the data set to be processed. For dsnaddr, use the address of the 44-byte data area that contains the data set name. The data set name must be left-justified and padded with blanks. This data set can be either SMS-managed or non-SMS-managed.

RETAINDAYS is a new, optional keyword of the ARCHBACK macro. RETAINDAYS specifies the address of a field that contains the retention days to assign to the new backup version that will be created via the ARCHBACK macro. For rdaddr, substitute the address of a 2-byte data area that contains the value of RETAINDAYS. The same range checking described under the (H)BACKDS command also applies to ARCHBACK.

ARCINBAK: Inline backup

ARCINBAK will be updated to add the optional RETAINDAYS keyword so that the inline backup allows customers to request a backup of a data set in a batch environment with the new RETAINDAYS keyword. The same range checking under the (H)BACKDS command also applies to ARCINBAK.

The new JCL that invokes ARCINBAK with a RETAINDAYS value of 999 is shown in Example 5-10.

Example 5-10 Sample JCL for ARCINBAK

```
//MHLRES4B JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES4
//*
//STEP1  EXEC PGM=????????
//SYSPRINT  DD  SYSOUT=A
```

```
//DSET1  DSN=MHLRES4.TEST.RETAINED,DISP=OLD
//DSET2
//DSET3   ..
//*
//STEP2  EXEC  PGM=ARCINBAK, PARM='TARGET(TAPE),
       CC(PREFERRED,PHYSICALEND),
       RETAINDAYS(999)'
 ..
//ARCPRINT  DD SYSOUT=A
//ARCSNAP DD SYSOUT=A
```

(H)LIST command syntax

Figure 5-4 shows (H)LIST command syntax.

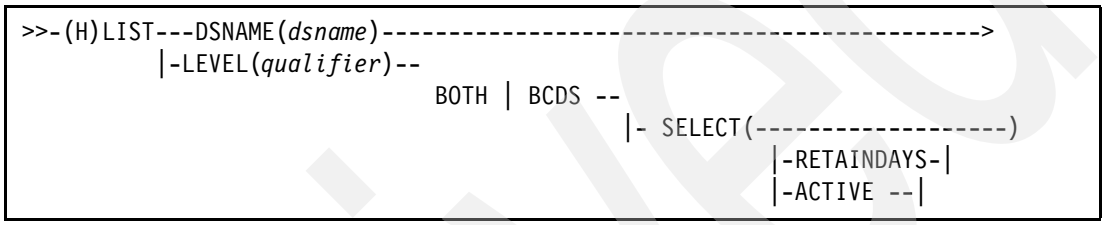


Figure 5-4 (H)LIST command syntax

Example 5-11 shows the LIST command for the MHLRES4.JCL.CNTL data set.

Example 5-11 Sample LIST command

```
LIST DSNAME(MHLRES4.JCL.CNTL) BCDS SELECT(RETAINDAYS)

DSN=MHLRES4.JCL.CNTL                              BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T124419.MHLRES4.JCL.J9225          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:44:12  CAT=YES  GEN=000  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00007

  BDSN=HSM.BACK.T434119.MHLRES4.JCL.J9225          BACKVOL=VT0062   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:41:39  CAT=YES  GEN=***  VER=***  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

 TOTAL BACKUP VERSIONS = 0000000002

 ARC0140I LIST COMPLETED,        6 LINE(S) OF DATA OUTPUT
 ***
```

RECOVER command syntax

Figure 5-5 shows the RECOVER command syntax.

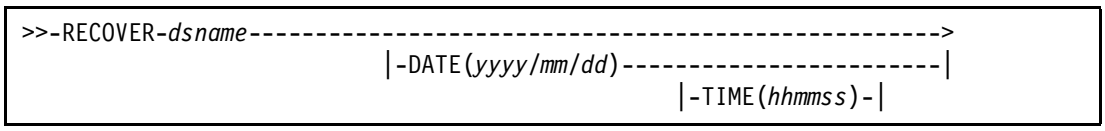


Figure 5-5 (H)RECOVER command syntax

The RECOVER dsname command syntax will be enhanced to enable TIME to be specified along with the DATE parameter. A fully qualified data set name is required for the command.

Example 5-12 shows the command to recover the MHLRES4.JCL.CNTL data set, which was backed up on 09/08/13 at 19:44:12.

Example 5-12 RECOVER command sample

```
Display:
BDSN=HSM.BACK.T124419.MHLRES4.JCL.J9225          BACKVOL=SBXHS5   FRVOL=MHLS1A
BACKDATE=09/08/13  BACKTIME=19:44:12  CAT=YES  GEN=000  VER=004  UNS/RET= NO
RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00007

Command to RECOVER:
RECOVER MHLRES4.JCL.CNTL DATE(2009/08/13) TIME(194412)

ARC1007I RECOVER REQUEST 00000030 SENT TO DFSMSHSM
ARC0778I DATA SET MHLRES4.JCL.CNTL WAS RECOVERED FROM A BACKUP MADE AT
ARC0778I (CONT.) 19:44:12 ON 2009/08/13
ARC1000I MHLRES4.JCL.CNTL RECOVER PROCESSING ENDED
```

HRECOVER command syntax

Figure 5-6 shows the HRECOVER command syntax.

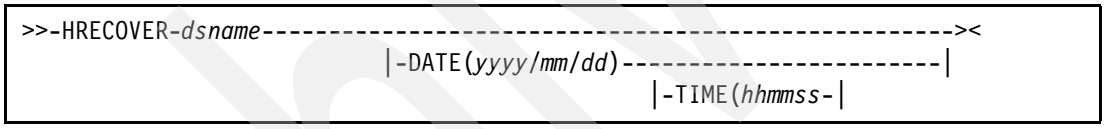


Figure 5-6 HRECOVER command syntax

The HRECOVER dsname command syntax will be enhanced to enable TIME to be specified with the DATE parameter. A partially qualified data set can also be specified for the command. If this is the case, then the DATE and TIME will be applied to all of the data sets that meet the filter criteria.

TIME is a new, optional keyword added to the (H)RECOVER command:

- If the TIME keyword is specified, the DATE keyword must also be specified or the (H)RECOVER command will fail. For *hhmmss*, substitute the time when the backup version to be recovered was created. The range for hours is 00-23, and for minutes and seconds is 00-59.
- If TIME is not specified, DFSMShsm does not consider this value when selecting the DATE to recover.
- If TIME is specified, DFSMShsm recovers the backup copy with the specified date and time. If the backup version is a retained backup copy, either GEN or VER will no longer be valid. You must use the LIST command to determine which backup copy was created at the specific DATE and TIME, and then use it to perform a recovery by passing these two new parameters.
- If only DATE is specified, DFSMShsm recovers the most recent backup copy, as it does currently if more than one backup copy exists with that DATE.
- If DATE and TIME are specified, then the exact backup copy will be recovered, which applies to both the active and retained copy.

ARCHRCOV command syntax

Figure 5-7 shows the ARCHRCOV command syntax.

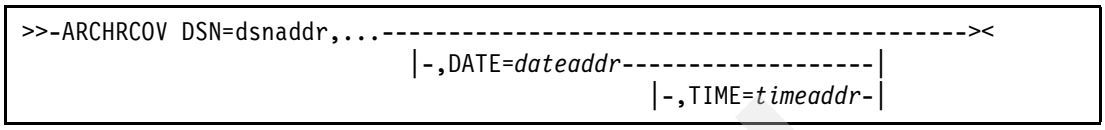


Figure 5-7 ARCHRCOV command syntax

TIME is a new, optional keyword of the ARCHRCOV macro. If TIME is specified without DATE, the macro invocation will fail.

TIME specifies the address of a field containing the time of creation of a backup version within a specified DATE that you want to recover. For *timeaddr*, substitute the address of a field containing the time of a specified date of the backup version you want to recover in using the *hhmmss* format, where the first two digits are hours, the middle two digits are minutes, and last two digits are seconds. The range for hours is 00-23, and for minutes and seconds is 00-59. The backup version that is recovered is the newest one created on or before the specified date and time. GEN and VER parameters will only be applied for the active backup copies. GEN and VER are not valid for retained copies. You can use the ARCXTRCT macro to extract data from DFSMShsm to obtain this backup copy creation date and time for backup versions that exist. This parameter applies to both SMS-managed and non-SMS-managed data sets. You can also use a LIST DSNAME BCDS/BOTH command to check if any retained copies exist, and if necessary, recover the data set from the retained backup copy.

(H)BDELETE command syntax

Figure 5-8 shows the (H)BDELETE command syntax.

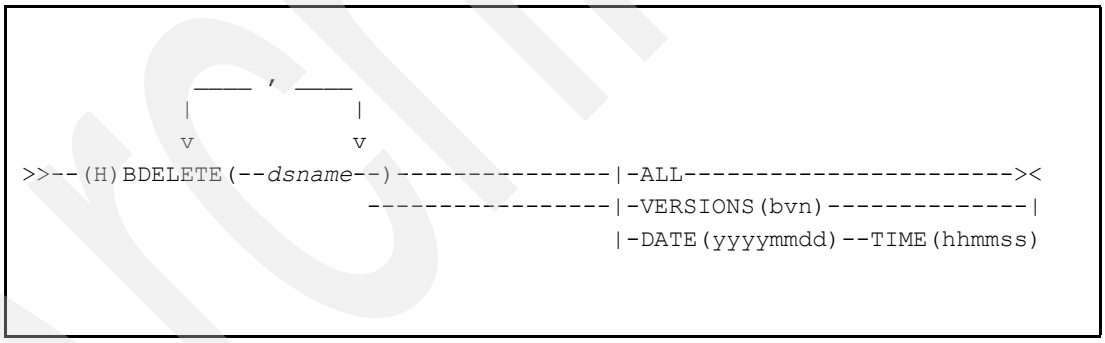


Figure 5-8 (H)BDELETE command syntax

The (H)BDELETE (dsname) command syntax will be enhanced to enable ALL to be specified for one or more multiple data sets. If multiple data sets are specified, the ALL keyword will apply to all specified data sets.

With HBDELETE, if multiple data sets are specified, the keyword ALL will apply to all specified data sets. If a partially qualified data set is specified, the keyword ALL will apply to all of the data sets that meet the filter criteria.

ALL is a new keyword added to the (H)BDELETE command. The ALL keyword is mutually exclusive with the VERSIONS and DATE keywords. ALL is required to delete all backup versions of the specified data set(s), including both active and retained copies. The (H)BDELETE command will fail with a parse error if ALL, VERSION, or DATE is not specified.

Example 5-13 shows samples of the BDELETE command run against the MHLRES4.JCL.CNTL data set.

Example 5-13 BDELETE command samples

```
BDELETE MHLRES4.JCL.CNTL ALL

BDELETE MHLRES4.JCL.CNTL VERSION(1)

BDELETE MHLRES4.JCL.CNTL DATE(2009/07/31) TIME(150454)
```

ARCHBDEL command syntax

Figure 5-9 shows the ARCHBDEL command syntax.

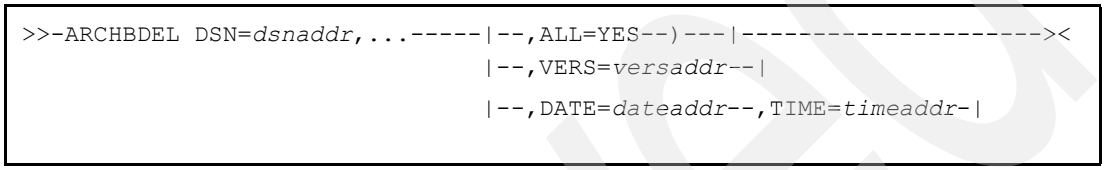


Figure 5-9 ARCHBDEL command syntax

ALL is a new parameter of the ARCHBDEL macro. When ALL=YES is specified, all backup versions are deleted.

DATE is a new parameter of the ARCHBDEL macro. DATE specifies the address of a field that contains the backup date to assign to the backup version that will be deleted via the ARCHBDEL macro. For *dateaddr*, substitute the address of a data area that contains the backup date. The field referenced by *dateaddr* must contain seven digits in the form of *yyyyddd*.

TIME is a new parameter of the ARCHBDEL macro. TIME can only be specified when DATE is also specified. TIME specifies the address of a field that contains the backup time to assign to the backup version that will be deleted via the ARCHBDEL macro. For *timeaddr*, substitute the address of a data area that contains the backup time. The field referenced by *timeaddr* must contain six digits in the form of *hhmmss*.

The DATE and TIME keywords must be specified together, or the macro invocation will fail.

ALL, VERS, and DATE keywords are mutually exclusive, and one of them must be specified for the command. The ARCHBDEL invocation will fail if none of the ALL | VERSION | DATE keywords are specified.

The VERS parameter can only be issued against active backup copies. To delete a retained copy, use the DATE and TIME keywords. You can use the LIST command or ARCXTRCT macro to extract data from DFSMShsm to obtain the backup version creation date and time for backup versions that exist.

FIXCDS syntax and MCBR record

A new CDS record MCBR (record type Z) is created to cater for retained copies. A new FIXCDS syntax is shown in Figure 5-10.

```
FIXCDS Z dsname INDEX(n)
FIXCDS Z X'hex Z key'
```

Figure 5-10 New FIXCDS command syntax for record type Z

Where:

- The INDEX parameter must be used to indicate the nth entry of the MCBR record set.
- The hex Z key is a unique HEX value derived from the MCB key through a special algorithm.

A sample command is shown in Figure 5-11.

```
FIXCDS Z MHLRES4.JCL.CNTL INDEX(1)
FIXCDS Z X'C4C6C8E250B572C0B1018C1C82FFFF'
```

Figure 5-11 FIXCDS Z sample command

5.1.4 BCDS sub-records type

BCDS contains several sub-record types. Data set backup information is kept in BCDS sub-record types, which include MCB, MCC, and a new MCBR sub-record. The following characteristics should be noted:

- The backup control data set data set record (MCB) describes a data set that has been backed up, and identifies all valid backup versions.
- The backup control data set backup version record (MCC) describes a backup version of a data set.
- A new sub-record type Z (MCBR) has been created in BCDS to describe a retained copy version of a data set.

Relationships between MCB, MCC, and MCBR records

We use the MHLRES4.JCL.CNTL data set as an example to illustrate the relationships between MCB, MCC, and MCBR records. This data set has a maximum number of backup versions set to five.

MCB record

You can list all the backup versions belong to a data set by running the FIXCDS B MHLRES4.JCL.CNTL command, Example 5-14 shows that this data set has five valid backup versions for MHLRES4.JCL.CNTL.

Example 5-14 FIXCDS B command output

```
MCH=   01D02000 C4A1EFC0 01FB024B C4A1EC47 A7B2948E
  *    D      .D                   *
 +0000  E2C2E7C8 E2F5FFFF 00000000 00000000 19571482 0109225F 02000C80 00908800
  *SBXHS5                         *
 +0020  00000011 00006C98 0000000F FFFFA000 00060005 0005FFFF 0109225C 00000000
  *                               *
 +0040  00000000 00000000 00000000 00000000 C8E2D44B C2C1C3D2 4BE3F1F4 F5F7F1F9
```

```
 *               HSM.BACK.T145719*
+0060  4BD4C8D3 D9C5E2F4 4BD1C3D3 4BD1F9F2 F2F54040 40404040 40404040 D4C8D3E2
 *.MHLRES4.JCL.J9225          MHLS*
+0080  F1C1C200 0109225F 00000006 00000000 C8E2D44B C2C1C3D2 4BE3F0F4 F5F7F1F9
 *1AB             HSM.BACK.T045719*
+00A0  4BD4C8D3 D9C5E2F4 4BD1C3D3 4BD1F9F2 F2F54040 40404040 40404040 D4C8D3E2
 *.MHLRES4.JCL.J9225          MHLS*
+00C0  F1C1C200 0109225F 00000005 00000000 C8E2D44B C2C1C3D2 4BE3F1F2 F4F4F1F9
 *1AB             HSM.BACK.T124419*
+00E0  4BD4C8D3 D9C5E2F4 4BD1C3D3 4BD1F9F2 F2F54040 40404040 40404040 D4C8D3E2
 *.MHLRES4.JCL.J9225          MHLS*
+0100  F1C1C280 0109225F 00000004 00070000 C8E2D44B C2C1C3D2 4BE3F5F6 F4F2F1F9
 *1AB             HSM.BACK.T564219*
+0120  4BD4C8D3 D9C5E2F4 4BD1C3D3 4BD1F9F2 F2F54040 40404040 40404040 D4C8D3E2
 *.MHLRES4.JCL.J9225          MHLS*
+0140  F1C15200 0109225F 00000003 00000000 C8E2D44B C2C1C3D2 4BE3F5F4 F4F2F1F9
 *1A              HSM.BACK.T544219*
+0160  4BD4C8D3 D9C5E2F4 4BD1C3D3 4BD1F9F2 F2F54040 40404040 40404040 D4C8D3E2
 *.MHLRES4.JCL.J9225          MHLS*
+0180  F1C15200 0109225F 00000002 00000000
 *1A                            *
ARC0197I TYPE B, KEY MHLRES4.JCL.CNTL, FIXCDS DISPLAY SUCCESSFUL
COMMAND REQUEST 00000260 SENT TO DFSMSHSM
***
```

The same information can be listed by running LIST DSN(MHLRES4.JCL.CNTL) BCDS SELECT(ACTIVE), as shown in Example 5-15.

Example 5-15 LIST DSN(MHLRES4.JCL.CNTL) BCDS SELECT(ACTIVE) output

```
DSN=MHLRES4.JCL.CNTL                              BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T145719.MHLRES4.JCL.J9225          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:57:14  CAT=YES  GEN=000  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T045719.MHLRES4.JCL.J9225          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:57:04  CAT=YES  GEN=001  VER=005  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T124419.MHLRES4.JCL.J9225          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:44:12  CAT=YES  GEN=002  VER=004  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00007

  BDSN=HSM.BACK.T564219.MHLRES4.JCL.J9225          BACKVOL=THS016   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:42:54  CAT=YES  GEN=003  VER=003  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T544219.MHLRES4.JCL.J9225          BACKVOL=THS016   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:42:50  CAT=YES  GEN=004  VER=002  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 TOTAL BACKUP VERSIONS = 0000000005

 ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
```

```
COMMAND REQUEST 00000261 SENT TO DFSMSHSM
***
```

MCC record

Using FIXCDS C HSM.BACK.T124419.MHLRES4.JCL.J9225 command (Example 5-16) shows information about a particular backup version of MHLRES4.JCL.CNTL. In this case, it lists all the information about the HSM.BACK.T124419.MHLRES4.JCL.J9225 backup version.

Example 5-16 FIXCDS C command output

```
MCH=   017C2400 C4A1ECD6 55C3DE0E C4A1ECD6 4125374E
 * @  D  0 C  D  0                *
+0000  D4C8D3D9 C5E2F44B D1C3D34B C3D5E3D3 40404040 40404040 40404040 40404040
 *MHLRES4.JCL.CNTL                *
+0020  40404040 40404040 40404040 E2C2E7C8 E2F50004 3030200F 19441293 0109225F
 *            SBXHS5              *
+0040  02000C80 00900004 00000011 00006C98 0000000F 00002000 8800D4C8 D3E2F1C1
 *                          MHLS1A*
+0060  00000000 3030200F 00000000 0109225F 00000000 00000000 00000000 00000000
 *                                *
+0080  00000000 00000000 00004040 40404040 40404040 40404040 40404040 40404040
 *                                *
+00A0  40404040 40404040 0008E2E3 C1D5C4C1 D9C44040 40404040 40404040 40404040
 *          STANDARD              *
+00C0  40404040 40404040 0008E2E3 C1D5C4C1 D9C44040 40404040 40404040 40404040
 *          STANDARD              *
+00E0  40404040 40404040 0000000F 0008C4A1 B7314C84 C48E0000 00000000 00000000
 *              D     D           *
+0100  00000000 00000000 00000000 00008000 000F0000 00000000 00000000 00000000
 *                                *
+0120  00000000 00000000 00000000 01000007 00000000 00000000 00000000
 *                                *
ARC0197I TYPE C, KEY HSM.BACK.T124419.MHLRES4.JCL.J9225, FIXCDS DISPLAY
ARC0197I (CONT.) SUCCESSFUL
COMMAND REQUEST 00000262 SENT TO DFSMSHSM
***
```

MCBR record

Using FIXCDS Z MHLRES4.JCL.CNTL INDEX(1), Example 5-17 shows that the MHLRES4.JCL.CNTL data set has one retained copy version named HSM.BACK.T434119.MHLRES4.JCL.J922.

Example 5-17 FIXCDS Z command output

```
MCH=   19902B00 00000000 00000000 00000000 00000000
 *                                *
+0000  E2C2E7C8 E2F5FFFF 00000000 00000000 19571482 0109225F 02000C80 00908800
 *SBXHS5                          *
+0020  00000011 00006C98 0000000F FFFFA000 00060001 0001FFFF 0109225C 00000000
 *                                *
+0040  00000000 00000000 00000000 00000000 C8E2D44B C2C1C3D2 4BE3F4F3 F4F1F1F9
 *               HSM.BACK.T434119*
+0060  4BD4C8D3 D9C5E2F4 4BD1C3D3 4BD1F9F2 F2F54040 40404040 40404040 D4C8D3E2
 *.MHLRES4.JCL.J9225          MHLS*
```

```
+0080  F1C15280 0109225F 00000001 00050000 00000000 00000000 00000000 00000000
 *1A                               *
+00A0  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
.
.
.
.
.
+1920  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
 *                                 *
+1940  00000000 00000000 00000000 00000000
 *                                 *
ARC0197I TYPE Z, KEY MHLRES4.JCL.CNTL, FIXCDS DISPLAY SUCCESSFUL
COMMAND REQUEST 00000263 SENT TO DFSMSHSM
***
```

As you can see, a MCB record can have multiple MCC records and multiple MCBR records. Each MCBR record would have a corresponding MCC record (a one to one relationship).

The maximum number of MCC records is the maximum number of backup versions (# BACKUPS (DS EXISTS)).

The maximum number of MCBR records for a data set can be unlimited. Figure 5-12 shows the relationship between MCB and MCBR records.

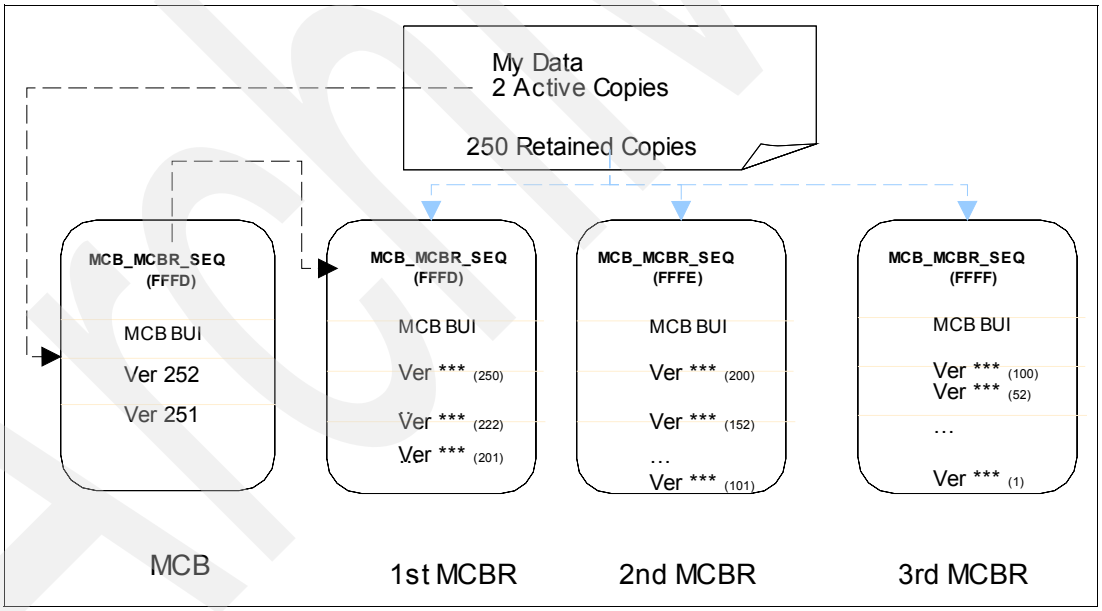


Figure 5-12 Relationship between MCB and MCBR record set

An internal relationship between MCB, MCC, and MCBR is shown in Figure 5-13.

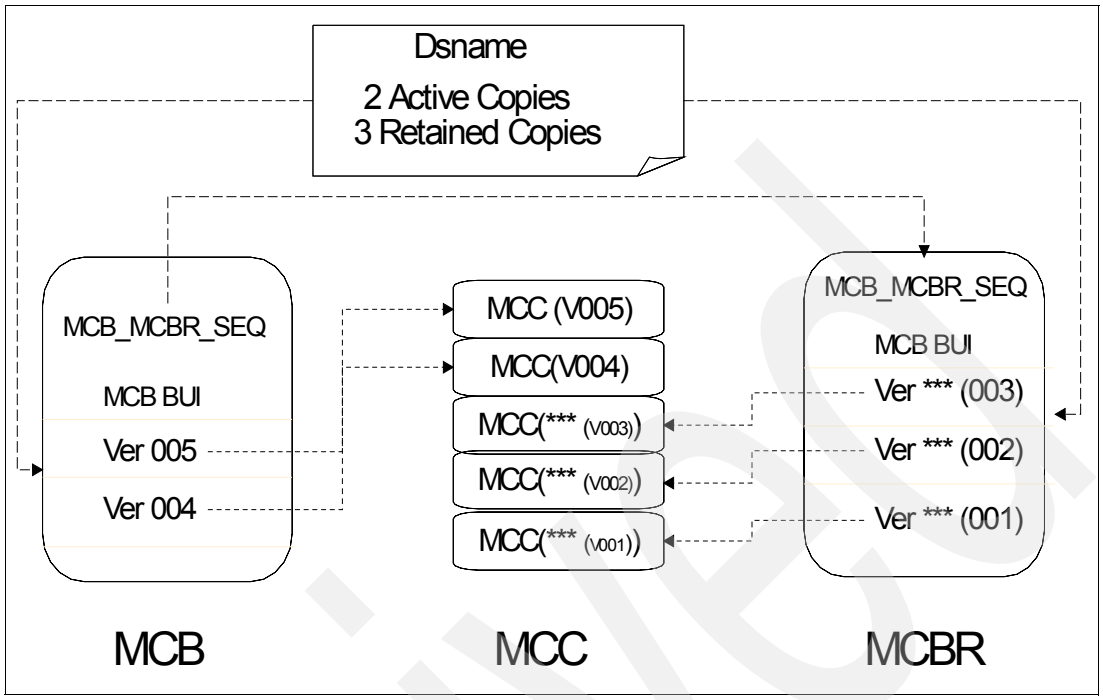


Figure 5-13 Relationships between MCB, MCC, and MCBR

5.1.5 Changes in ISMF

The following ISMF panels have been modified.

HBACKDS panel

On ISMF, option 1, Data Set, choose the data set for which you want to run HBACKDS or HRECOVER. This panel will work for both, as shown in Figure 5-14.

```
 Panel  Defaults  Utilities  Scroll  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                       DATA SET SELECTION ENTRY PANEL             Page 1 of 5
Command ===>

For a Data Set List, Select Source of Generated List  . . 2  (1 or 2)

  1  Generate from a Saved List         Query Name To
       List Name  . .                   Save or Retrieve

  2  Generate a new list from criteria below
       Data Set Name . . . MHLRES4.JCL.CNTL
       Enter "/" to select option      Generate Exclusive list
       Specify Source of the new list  . . 2    (1 - VTOC, 2 - Catalog)
       1 Generate list from VTOC
          Volume Serial Number . . .            (fully or partially specified)
          Storage Group Name . . . .            (fully specified)
       2 Generate list from Catalog
          Catalog Name . . .
          Volume Serial Number . . .            (fully or partially specified)
          Acquire Data from Volume . . . . . . . N  (Y or N)
          Acquire Data if DFSMShsm Migrated  . . N  (Y or N)
Use ENTER to Perform Selection; Use DOWN Command to View next Selection Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 5-14 Chosen data set was MHLRES4.JCL.CNTL

Next step, issue the HBACKDS command on the right side of the panel, as shown in Figure 5-15.

```
 DATA SET LIST
Command ===>                                                Scroll ===> HALF
                                                          Entries 1-1 of 1
Enter Line Operators below:                               Data Columns 3-5 of 42

     LINE                                  ALLOC      ALLOC     % NOT
     OPERATOR         DATA SET NAME        SPACE      USED      USED
    ---(1)----  ------------(2)------------ ---(3)---  ---(4)---  -(5)-
    HBACKDS     MHLRES4.JCL.CNTL           ---------  ---------    ---
    ----------  ------  -----------  BOTTOM OF  DATA  -----------  ------  ----
```

Figure 5-15 HBACKDS command

The Retain Days field is added to the panel, as shown in Figure 5-16.

```
 Panel  Utilities  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              HBACKDS ENTRY PANEL
 Command ===>

 Optionally Specify one or more for
  Data Set: MHLRES4.JCL.CNTL

   Target . . . . . . . . . . . . .          (D=DASD, T=TAPE or blank)

   Concurrent copy  . . . . . . . .          (P, R, S, VP, VR, CP, CR or blank)
     For P, R, VP, VR, CP or CR,
                 enter CC option             (P=PHYSICAL, L=LOGICAL or blank)

   Wait for Completion  . . . . . . N        (Y or N)

   Data Set Password . . . . . . .           (if password protected)

   Retain Days . . . . . . . . . .           (0 to 50000, or 99999 or blank)

   Note: Dataset Password ignored when in ADMIN mode.

 Use ENTER to Perform Hbackds;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 5-16 HBACKDS ISMF panel

The Retain Days value has a range of 0 to 50000 or a unique value of 99999. A value of 99999 will be treated as NEVER EXPIRED.

A new message, INVALID RETAIN® DAYS, will be issued if its entered value is not valid.

HRECOVER ISMF panel

To reach the HRECOVER panel, see the menu shown in Figure 5-17.

```
------------------------------------------------------------------------------
                                 DATA SET LIST
Command ===>                                                 Scroll ===> HALF
                                                       Entries 1-1 of 1
Enter Line Operators below:                            Data Columns 3-5 of 42

     LINE                                 ALLOC      ALLOC     % NOT
     OPERATOR          DATA SET NAME      SPACE      USED      USED
    ---(1)----  ------------(2)------------ ---(3)---  ---(4)---  -(5)-
     HRECOVER   MHLRES4.JCL.CNTL          ---------  ---------    ---
    ----------  ------  -----------  BOTTOM OF  DATA  -----------  ------  ----
```

Figure 5-17 HRECOVER command

A new TIME field has been added to the first panel of ISMF HRECOVER, as shown in Figure 5-18. This panel only shows the active backup version.

You can choose the backup version to recover by entering Y at column RECOVER (Y/N).

```
 Panel  Utilities  Scroll  Help
 ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                             HRECOVER ENTRY PANEL      RETAINED BKUP CPY EXISTS
 Command ===>

 Specify Y to recover a Backup Version for
  Data Set: MHLRES4.JCL.CNTL
                               Recover                                  Recover
 Version  Date    TIME   Gen#  (Y/N)  |  Version  Date    TIME   Gen#   (Y/N)
 --------------------------------------|----------------------------------------
 006     2009225  195714  00     N    |  005     2009225  195704  01      N
 004     2009225  194412  02     N    |  003     2009225  194254  03      N
 002     2009225  194250  04     N    |
                                      |
                                      |
                                      |
                                      |
                                      |
                                      |
                                      |

 Use ENTER to Continue; Use DOWN to select other versions;
 Use HELP Command for Help; Use END Command to Cancel the HRecover.
```

Figure 5-18 New TIME field in HRECOVER panel

In this example, the MHLRES4.JCL.CNTL data set has retained copies, as indicated by the `RETAINED BKUP CPY EXISTS` message on the top right corner of the panel. You must use the (H)LIST DSN(data set name) BCDS SELECT(RETAINDAYS) command to list the retained copies versions, as shown in Example 5-18.

Example 5-18 LIST command for Retained Copies output

```
DSN=MHLRES4.JCL.CNTL                              BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T124419.MHLRES4.JCL.J9225          BACKVOL=SBXHS5   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:44:12  CAT=YES  GEN=000  VER=004  UNS/RET= NO
RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00007

  BDSN=HSM.BACK.T434119.MHLRES4.JCL.J9225          BACKVOL=THS016   FRVOL=MHLS1A
  BACKDATE=09/08/13  BACKTIME=19:41:39  CAT=YES  GEN=***  VER=***  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00005

 TOTAL BACKUP VERSIONS = 0000000002

 ARC0140I LIST COMPLETED,        6 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000270 SENT TO DFSMSHSM
 ***
```

Use the HRECOVER command with the DATE and TIME parameter to recover the data set, as shown in Example 5-19.

Example 5-19 HRECOVER sample command for Retained Copies version

```
HRECOVER MHLRES4.JCL.CNTL DATE(09/08/13) TIME(194412)
```

5.1.6 Security

DFSMShsm uses the following set of RACF profiles to protect commands:

- STGADMIN.ARC.*command:* This is a profile for a specific DFSMShsm storage administrator command.
- STGADMIN.ARC.*command.parameter:* This profile protects a specific DFSMShsm administrator command with a specific parameter.
- STGADMIN.ARC.ENDUSER.*h_command:* This profile protects a specific DFSMShsm user command.
- STGADMIN.ARC.ENDUSER.*h_command.parameter:* This profile protects a specific DFSMShsm user command with a specific parameter.

A new facility class name is defined to protect the (H)BACKDS command RETAINDAYS parameter. This new feature gives users the freedom to create backup versions that are to be kept beyond what the Management Class allows.

Therefore, it is important to restrict which users can use the RETAINDAYS keyword. It is also important to maintain the extra retained backup versions that the Storage Administrator might want to control. Users must have READ access authority to the profile in order to use the command.

The Storage Administrator must ensure that the RACF rule in Table 5-1 is defined. Define this resource with the UACC(NONE) authority.

Table 5-1 RACF resource to control RETAINDAYS parameter

Facility class resource	Access authority
STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS	UACC(NONE)

Use the RACF PERMIT command to set which users can use the RETAINDAYS parameter, as shown in Figure 5-19.

```
PERMIT STGADMIN.ARC.ENDUSER.HBACKDS.RETAINDAYS CLASS(FAC) ID(userid) ACC(READ)
```

Figure 5-19 RACF command to allow users to issue the RETAINDAYS parameter in HBACKDS command

Storage administrators must have READ access authority to the following profiles in order to use the command or command and parameter:

- STGADMIN.ARC.*command*
- STGADMIN.ARC.*command.parameter*

5.1.7 New and updated messages

The following messages have been added or modified in V1.11 for DFSMShsm.

New messages

The following messages are new for z/OS V1.11.

ARC1068I

A new message that is related to data set recovery, as shown in Figure 5-20.

```
ARC1068I INVALID PARAMETER SPECIFIED FOR RECOVERY OF {DATA SET=dsname |
VOLUME=volser}, RC=return code
```

Figure 5-20 New ARC1068I message

ARC1378I

Figure 5-21 shows the new ARC1378I message.

```
ARC1378I UNCATALOGUED DATA SET COULD NOT BE PROCESSED FOR BACKUP
```

Figure 5-21 ARC1378I message

Updated messages

The following messages have been updated in V1.11.

ARC1334I

REAS=19 has been added, as shown in Figure 5-22.

```
ARC1334I - BACKUP VERSION NOT CREATED.
REAS=19: internal parm error
```

Figure 5-22 New REASON code for ARC1334I

ARC1381I

New RC=60 and RC=62 have been added, as shown in Figure 5-23.

```
ARC1381I BACKUP OF DATA SET=dsname TO NEWNAME=newdsname FAILED, RC=retcode
RC=60 RETAINDAYS was specified with NEWNAME and DATE. Creating a new version
with the specified date and retention period would result in a backup copy that
has already expired.
RC=62 RETAINDAYS was specified with NEWNAME and DATE, and the specified date is
earlier than the date when the most recent retained backup copy was created.
This would result in creating a backup version within the set of existing
retained copies, which is not allowed.
```

Figure 5-23 New RCs added for ARC1381I

ARC1605I

A new REAS=42 has been added, as shown in Figure 5-24.

```
ARC1605I COMMAND HAD PARSE ERROR
REAS=42 The number of RETAINDAYS is a required parameter that must be an
integer in the range of 0 to 50000, or 99999. RETAINDAYS values between the
range of 50001 and 99998 are not accepted. A RETAINDAYS value of 99999 is
managed as a 'never expire' value.
```

Figure 5-24 New REAS=42 for ARC1605I message

ARC0185I

A new RC=20 has been added, as shown in Figure 5-25

```
ARC0185I {HBDELETE | BDELETE} COMMAND FAILED, DSN=dsname, BDSN=backupdsname,
FROM VOL=original-volser, BACKUP DATE=date-version-was-made, CAT={YES | NO},
VER=version-number, RETURN CODE=return-code, REASON CODE=reason-code
RC=20 An ARCBZKEY error has occurred obtaining the key for the BCDS Z record.
See additional information in previously issued TRAP message.
```

Figure 5-25 New RC=20 for ARC0185I message

ARC0189I

Record type Z has been included in the message, as shown in Figure 5-26.

```
ARC0189I NO DFSMShsm CONTROL DATA SET C RECORD FOUND FOR BACKUP VERSION
backupdsname OF {CATALOGED | UNCATALOGED} DATA SET dsname, BACKUP VERSION ENTRY
DELETED FROM DFSMSHSM BACKUP CONTROL DATA SET [B | Z] RECORD
```

Figure 5-26 New ARC0189I format

5.1.8 Migration and coexistence considerations

This section contains migration and coexistence considerations.

Migration considerations

There is a new CDS record MCBR (record type Z or '2B'X in hex format) that caters to retained copies.

As the number of backup versions for each data set can be unlimited as well as new CDS record type Z created, the BCDS Size Work Sheet includes the estimated number of retained copies in addition to the number of current active backup versions that is used with the VERSION parameter of the SETSYS command or is specified in the Management Classes. Refer to the BCDS Size Work Sheet in Appendix A, "DFSMShsm Work Sheets", in *DFSMShsm Implementation and Customization Guide,* SC35-0418 in order to estimate the size of BCDS.

Prerequisites for installation

OA26327 is required for multiple release systems.

FSR

FSR types 7, 19, and 20 include:

- FSRF_BACKDS_RD_SPCD: The version with RETAINDAYS specified.
- FSRF_BACKDS_NEVER_EXP: The version that never expires.
- FSR_BACKDS_RETAINDAYS: The RETAINDAYS value is specified at the time of backup.

You can use RMM to generate reports on those newly updated FSR fields.

DCOLLECT backup data set information

The DCOLLECT type B record contains the RETAINDAYS value:

- UBF_RETAIN_SPCD: Whether the RETAINDAYS keyword was used to create the version.
- UBF_NEVER_EXP: The version never expires when set ON.
- UB_RETAINDAYS: The RETAINDAYS value specified at the time of backup.

Coexistence considerations

This enhancement is not supported on systems that have a version earlier than z/OS V1.11 DFSMS.

The coexistence APAR - OA26327 allows DFSMShsm to issue warning messages to recommend functions, for example, BACKDS, BDELETE, RECOVER, RECYCLE, FREEVOL, AUDIT, and EXPIREBV (from lower level systems), to be run on V1.11 or higher when encountering a retained backup copy.

If users fall back to lower level systems (pre-V1.11) from V1.11, they need to delete all of their Retained Copies. Failing to perform this action might cause data set backup to fail, as the Retained Copies will not roll off in the lower level systems.

The following HSM functions are affected when used on lower level systems:

- Backing up a data set on a lower level will only roll off a non-retention backup copy.

  Here we use the MHLRES.JCL.CNTL data set as our example. Its maximum number of backup versions is five.

  In Example 5-20, only backup copy version 16 is a non-retention version (RETDAYS=*****), and it is the most recent backup version.

  Example 5-20 LIST output of active copies on z/OS V1.11 prior to creating a new backup version on a lower level system

```
DSN=MHLRES4.JCL.CNTL                                  BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T334715.MHLRES4.JCL.J9209          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/28  BACKTIME=15:47:33  CAT=YES  GEN=000  VER=016  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T444015.MHLRES4.JCL.J9209          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/28  BACKTIME=15:40:44  CAT=YES  GEN=001  VER=014  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T364015.MHLRES4.JCL.J9209          BACKVOL=SBXHS5
FRVOL=MHLS1A
```

```
  BACKDATE=09/07/28  BACKTIME=15:40:36  CAT=YES  GEN=002  VER=013  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T401915.MHLRES4.JCL.J9209          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/28  BACKTIME=15:19:40  CAT=YES  GEN=003  VER=010  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=004  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

 TOTAL BACKUP VERSIONS = 0000000005

 ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000106 SENT TO DFSMSHSM
 ***
```

After a new backup version is created (VER=017) on a lower level system (z/OS V1.10), the non-retention backup version (VER=016) is rolled off (no longer shown in the output), as shown in Example 5-21. All other backup versions that have a RETDAYS value other than the value of ***** (retained copies) remain intact.

Example 5-21 LIST output of active copies on z/OS V1.11 after a new backup version is created on a lower level system (z/OS V1.10)

```
DSN=MHLRES4.JCL.CNTL                                  BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T515315.MHLRES4.JCL.J9209          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/28  BACKTIME=15:53:51  CAT=YES  GEN=000  VER=017  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T444015.MHLRES4.JCL.J9209          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/28  BACKTIME=15:40:44  CAT=YES  GEN=001  VER=014  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T364015.MHLRES4.JCL.J9209          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/28  BACKTIME=15:40:36  CAT=YES  GEN=002  VER=013  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T401915.MHLRES4.JCL.J9209          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/28  BACKTIME=15:19:40  CAT=YES  GEN=003  VER=010  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

  BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5
FRVOL=MHLS1A
  BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=004  VER=006  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=00015

 TOTAL BACKUP VERSIONS = 0000000005
```

```
ARC0140I LIST COMPLETED,       12 LINE(S) OF DATA OUTPUT
COMMAND REQUEST 00000107 SENT TO DFSMSHSM
***
```

The same LIST command on a lower level (z/OS V1.10) is shown in Example 5-22.

Example 5-22 LIST output of active copies on z/OS V1.10 after a new backup version is created on a lower level system (z/OS V1.10)

```
DSN=MHLRES4.JCL.CNTL                                  BACK FREQ = ***  MAX VERS=***

 BDSN=HSM.BACK.T515315.MHLRES4.JCL.J9209          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/28  BACKTIME=15:53:51  CAT=YES  GEN=000  VER=017  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=***

 BDSN=HSM.BACK.T444015.MHLRES4.JCL.J9209          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/28  BACKTIME=15:40:44  CAT=YES  GEN=001  VER=014  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=***

 BDSN=HSM.BACK.T364015.MHLRES4.JCL.J9209          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/28  BACKTIME=15:40:36  CAT=YES  GEN=002  VER=013  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=***

 BDSN=HSM.BACK.T401915.MHLRES4.JCL.J9209          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/28  BACKTIME=15:19:40  CAT=YES  GEN=003  VER=010  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=***

 BDSN=HSM.BACK.T221320.MHLRES4.JCL.J9204          BACKVOL=SBXHS5   FRVOL=MHLS1A
 BACKDATE=09/07/23  BACKTIME=20:13:22  CAT=YES  GEN=004  VER=006  UNS/RET= NO
 RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=***

ARC0140I LIST COMPLETED,       11 LINE(S) OF DATA OUTPUT
COMMAND REQUEST 00000026 SENT TO DFSMSHSM
***
```

Eventually, when all five active copies are backup versions with RETAINDAYS specified, then the backup action for that data set on the lower system will fail, as shown in Example 5-23.

Example 5-23 Failed BACKDS command output

```
ARC1001I MHLRES4.JCL.CNTL  BACKDS FAILED, RC=0034, REAS=0020
ARC1334I BACKUP VERSION NOT CREATED
COMMAND REQUEST 00000027 SENT TO DFSMSHSM
***
```

- Autobackup processing fails when attempting to move a retained backup copy from ML1 to tape.
- EXPIREBV skips backup versions with retain days specified.

- RECOVER DATE handles only active backup versions. If the RECOVER DATE command is issued and the version is not found and exists potentially as a retained copy, a warning message (Example 5-24) is issued to inform the user that retained backup copies can only be recovered on release V1.11 and up.

Example 5-24 Sample message: Recovering a RETAINED backup version on a lower level system

```
ARC1069I THE VERSION TO RECOVER MAY BE A RETAINED COPY
ARC1001I MHLRES4.JCL.CNTL RECOVER FAILED, RC=0028, REAS=0004
ARC1128I BACKUP/DUMP COPY DOES NOT EXIST
COMMAND REQUEST 00000031 SENT TO DFSMSHSM
***
```

- BDELETE handles only active backup versions. If the BDELETE DATE command is issued and the version is not found and exists potentially as a retained copy, a warning message (Example 5-25) is issued.

Example 5-25 BDELETE DATE command sample output

```
ARC0185I BDELETE COMMAND FAILED, DSN=MHLRES4.JCL.CNTL, BDSN=***, FROM VOL=***,
ARC0185I (CONT.) BACKUP DATE=***, CAT=***, VER=***, RETURN CODE=42, REASON
ARC0185I (CONT.) CODE=0
ARC0182I NO BACKUP VERSION(S) DELETED FOR MHLRES4.JCL.CNTL
COMMAND REQUEST 00000034 SENT TO DFSMSHSM
***
```

- AUDIT DIRCTL VOLUMES, DSCNTL, and BCDS will not process retained backup versions. When AUDIT encounters an MCC record, it checks to see whether a corresponding MCB record exists. If the MCB does not exist, then AUDIT checks for the existence of the record that indicates that this is a retained copy.
- Chapter 67, "Using the AUDIT Command", in *DFSMShsm Storage Administration Reference,* SC26-0422 will be updated with the changes made to the various AUDIT functions
- RECYCLE (including FORCE) fails when attempting to delete a retained backup copy.
- FREEVOL BVOL AGE(DAYS), where DAYS is 0, only moves all active backup versions of a data set to the tape. FREEVOL ML1BACKUPVERSIONS moves all backup versions off the migration volumes.
- DELVOL PURGE only deals with the active backup copies. If the backup version is a retained copy, a warning message will be issued and the DELVOL PURGE command will fail.

FIXCDS might run into an MCC record that belongs to a retained copy. FIXCDS C 'DSNAME' DISPLAY GEN(0) and PATCH GEN(0) are not impacted because the path will go through the MCB record first.

5.2 DFSMShsm Migration Level 1 enhancements

In DFSMS in z/OS V1.11, migrate and backup data sets larger than 64 K tracks in size may reside on Migration Level 1 (ML1) DASD as large format sequential (LFS) data sets. OVERFLOW volumes can now be utilized for migration and backup copies for data sets of a installation specified size.

5.2.1 Existing environment

Currently, all DFSMShsm data sets on ML1 volumes are single volume physical sequential data sets.

The ML1 selection algorithm fills all ML1 volumes evenly. This results in all ML1 volumes being filled to virtually the same percentage (the default is 80%).

This might cause problems when larger data sets must be migrated or backed up to ML1 volumes. Even though there might be enough space across multiple ML1 volumes to accommodate the large data set, there might not be a single ML1 volume that has enough space for the data set. The migration or backup might fail (unless a ML1 overflow is defined and has enough free space).

Moreover, data sets that have a size greater than 65534 tracks can only be migrated or backed up to tape.

Data set size estimation

During the migration/backup process for data sets to ML1 volumes, DFSMShsm calculates the data set's primary allocation size on the ML1 volume using the following algorithm:

- If the unit type of the selected volume is the same as the unit type of the source volume, DFSMShsm uses the allocated space size and used space size of the source data set to calculate the primary and secondary allocation of the target migration/backup copy. Each type of data set can have a different type of allocation methodology.
  - For VSAM, PS, and PDS (DSS data mover) with zero secondary allocation:
    - If the allocated size is greater than 2500 tracks, then the primary allocation size on the ML1 volume is equal to the allocated size.
    - If the allocated size is less than or equal to 2500 tracks, then the allocation size is 10 percent greater than the allocated size.
  - For all other data sets:
    - If the allocated size is greater than 2500 tracks, then the primary allocation size on the ML1 volume is equal to the used size.
    - If the allocated size is less than or equal to 2500 tracks, then the primary allocation size is 10 percent greater than the used size.
  - For all other data sets, secondary allocation is one tenth the size of the primary allocation.
- If the unit type of the selected volume is not the same as the source volume, DFSMShsm recalculates the primary allocation using track size and optimal blocksize for the selected volume.

If compaction is in use and the data set is eligible for compaction, then the calculated primary allocation is halved, but the secondary allocation remains the same.

ML1 volume selection for migration data set

DFSMShsm does not take into account the size of the copy in the migration volume selection process. DFSMShsm selects the ML1 volume by using the following criteria:

- The least number of users
- The most free space without consideration to volume fragmentation

Sometimes DFSMShsm cannot allocate a migration copy on a ML1 volume because of the following errors:

- There is not enough space on a ML1 volume (RC=37).
- The VTOC or VTOC Index is full (RC=44).
- There are other dynamic allocation errors (RC=8, REAS=DYNALLOC).

If RC=37 or RC=44 has been detected during interval migration (IM) or primary space management (PSM), DFSMShsm retries migration and selects another ML1 volume by using the most free space priority. This action can happen if the least number of users is not zero. For non-PSM and non-IM migrations, DFSMShsm holds ML1 migration if an out of space condition has been detected and the maximum free space amount on the ML1 volumes pool is less than the MCVTSMDS value specified in ARCCVT (the default value is 150 of 2 K blocks). Migration is also held if there was a dynamic allocation error or there was a VTOC or VTOC index error experienced.

ML1 volume selection for a backup data set

DFSMShsm uses ML1 volumes to backup data sets by using one of the following functions:

- Inline backup
- HBACKDS macro
- BACKDS macro
- ARCHBACK macro

DFSMShsm does not take into account the size of the copy in the migration volume selection process. DFSMShsm selects the ML1 NOOVERFLOW volume by using the following criteria:

- The least number of users
- The most free space without consideration to volume fragmentation

DFSMShsm differentiates between the conditions `There is not enough space on a ML1 volume` and `The VTOC or VTOC Index is full` and uses RC=37 for both of them. If RC=37 has been detected during backup, DFSMShsm retries the backup and selects another ML1 NOOVERFLOW volume with the most free space. If the second retry is not successful and a ML1 OVERFLOW volume exists, the backup will be retried a third time by selecting the OVERFLOW volume with the most free space. This is true for all data sets in any environment that has OVERFLOW volumes allocated.

DFSMShsm uses a ML1 OVERFLOW volume for backup versions if:

- The subparameter OVERFLOW was specified on the ADDVOL command for a ML1 volume.
- The data set being backed up is larger than 500 tracks.
- An out-of-space failure has occurred when using the least-usage and most-free-space selections that do not include the OVERFLOW volumes.

Note: Except for data sets that qualify to be migrated to a SDSP that resides on an OVERFLOW volume, data sets migrating to ML1 volumes are not targeted to OVERFLOW volumes. Migrating data flows to OVERFLOW volumes if an SDSP is defined in the OVERFLOW volume.

5.2.2 New z/OS 1.11 environment

In z/OS 1.11, the following enhancements have been made:

- Data sets whose expected size after compaction is greater than 64 K tracks can be migrated or backed up to ML1.
- DFSMShsm allows ML1 OVERFLOW volumes to be selected for migration processing in addition to their current use for data set backup processing.

Large Format Sequential data set

DFSMShsm allocates migration/backup copies for data sets whose expected size of the migration copy or backup version is equal or greater than 58 K tracks as a large format physical sequential (LFS) data set. The maximum size of a LFS migration/backup copy can be 16,777,214 tracks, which is larger than what the maximum size that a volume can be on V1.11. A data set that resides on an ML1 are single volume physical sequential and the maximum volume size of an EAV in V1.11 is 3,940,020 tracks. The maximum allowable migration/backup copy size is equal to the maximum size limit constraint for the largest volume available.

Table 5-2 describes the size of the migration or backup copy data set and its corresponding migration/backup copy format.

Table 5-2 ML1 data set format allocation based on data set expected size

Data set expected size (in tracks)	Migration/Backup copy format
<58K	Basic Physical Sequential
>=58K	Large Format Sequential

Overflow volumes

There is a new SETSYS command that regulates what a data set's expected size must be to have it allocated on the prefer ML1 OVERFLOW volume selection for migration or backup copies.

Figure 5-27 show the new SETSYS command syntax.

```
SETSYS ML1OVERFLOW(DATASETSIZE(dssize) THRESHOLD(threshold))
```

Figure 5-27 New SETSYS command to regulate ML1 OVERFLOW volumes

Where:

- DATASETSIZE(dssize) indicates the smallest data set size in kilobytes that should target ML1 Overflow volumes.
  - The default value is 2000000 (about 36 K tracks).
  - The dssize should be less than the average free space on a standard ML1.
- THRESHOLD(threshold) indicates the minimum average used space for all ML1 Overflow volumes before migration from ML1 Overflow volumes to ML2 begins.
  - The standard ML1 begins this migration when any volume hits its threshold.
  - The default value is 80%.

If the calculated size of the migration/backup copy is less than 36 K tracks, DFSMShsm uses the existing algorithm to select the ML1 volume for the migration copy or backup version.

DFSMShsm prefers a NOOVERFLOW ML1 volume with the maximum amount of free space and the least number of users.

If the actual size of the source data set is greater than 36 K tracks, DFSMShsm uses the new algorithm to select the ML1 OVERFLOW volume for the migration copy or backup version. DFSMShsm takes in account the expected size of migration copy or backup version. When DFSMShsm calculates the expected size of a copy, it takes in account the allocated and used size of the target data set and compaction (if the data set is eligible for compaction). DFSMShsm selects the ML1 OVERFLOW volume with the least amount of space available that still meets the requirement for the expected migration/backup copy size plus extra 10% (the least required amount of free space).

In the new volume selection algorithm for OVERFLOW volumes, there will be two separate priorities for how we select a migration / backup volume:

1. Expected compaction size of the data set (2:1 compression + 10% of data set size)
2. Actual data set size

Volume selection will be chosen by using the following criteria:

1. The OVERFLOW volume with the least number of users that would be able to fit the expected compaction size data set or actual size data set with the least amount of free space available on the volume.

   If there are no volumes in the ML1 OVERFLOW volume pool that meet this volume selection criteria, then you should apply the next criteria.
2. The OVERFLOW volume with the most amount of free space that meets the expected compaction data set size (priority 1 only).
3. The OVERFLOW with the most amount of free space (no priority, but still attempt to fit it into an OVERFLOW pool first)

If out of space (RC=37) or VTOC errors (RC=44) is detected during the migration or backup processes, DFSMShsm retries the migration or backup and selects another volume. The maximum amount of retries is four.

If a ML1 OVERFLOW volume is not defined to DFSMShsm or there is not enough room on the OVERFLOW volume, the request to migrate or backup a large data set to ML1 is attempted on a NOOVERFLOW volume. If the request cannot be fulfilled on a ML1 OVERFLOW or ML1 NOOVERFLOW volume, then the corresponding migration or backup ARC1208I or ARC1308I message is issued with a modified RC=10 message text. This text indicates that the migration or backup data set's size is larger than the available allocation room on the OVERFLOW or NOOVERFLOW volumes. Migration continues as normal and this data set remains on L0 DASD. DFSMShsm holds the ML1 migration if an out of space condition has been detected for both the NOOVERFLOW and OVERFLOW volumes and the maximum free space amount on the ML1 volumes pool is less than the MCVTSMDS value. The MCVTSMDS value is the minimum free space amount on the ML1 volumes pool allowable for an allocation (the default is 300 KB).

DFSMShsm will also be updated to process ML1 volumes differently when the error conditions VTOC is full and VTOC index is full are encountered. DFSMShsm continues to mark these volumes full (as is done currently by using the MVTFVFUL flag), but now eliminates these volumes from the ML1 volume selection process. DFSMShsm no longer is able to select ML1 volumes with full VTOC errors for further processing until the error is corrected.

Migration and backup ML1 volume selection algorithm example

The following example illustrates how DFSMShsm selects an ML1 volume for a migration or backup data set.

The current setup has:

- Two ML1 overflow volumes:
  - SBXHS4 with total free space of 105,344 tracks
  - SBXHS6 with total free space of 150,033 tracks
- One ML1 no overflow volume (SBXHS5) with a total free space of 119,360 tracks
- MHLRES4.ML1ENH data set with total size of 64,500 tracks
- SETSYS ML1OVERFLOW(DATASETSIZE(6000000)), which is equivalent to 108K tracks

A migrate command is issued to migrate MHLRES4.ML1ENH to the ML1 volume. Because SETSYS ML1OVERFLOW(DATASETSIZE(6000000)) is set, DFSMShsm selects the ML1 no overflow volume, which is SBXHS5, as shown in Example 5-26.

Example 5-26 LIST command output for MHLRES.ML1ENH after it is migrated to a ML1 volume

```
DSN=MHLRES4.ML1ENH                                  MIGVOL=SBXHS5 DSO=PS  SDSP=NO
  LAST REF=09/07/28 MIG=09/07/28 TRKS=0064500 2K BLKS= 1758651 TIMES MIG= 001
  16K BLKS=******  LAST MIGVOL=******

ARC0140I LIST COMPLETED,        3 LINE(S) OF DATA OUTPUT
COMMAND REQUEST 00000165 SENT TO DFSMSHSM
***
```

We now recall the MHLRES4.ML1ENH data set back to the primary volume and reset SETSYS ML1OVERFLOW(DATASETSIZE(2000000)), which is equivalent to 36 K tracks.

The environment is now as follows:

- Two ML1 overflow volumes:
  - SBXHS4 with a total free space of 105,344 tracks
  - SBXHS6 with a total free space of 150,033 tracks
- One ML1 no overflow volume SBXHS5 with a total free space of 119,360 tracks
- The MHLRES4.ML1ENH data set with total size of 64,500 tracks
- The MHLRES4.ML1ENH2 data set with total size of 64,500 track
- SETSYS ML1OVERFLOW(DATASETSIZE(2000000)), which is equivalent to 36K tracks

A migrate command is issued to migrate MHLRES4.ML1ENH to the ML1 volume. Because SETSYS ML1OVERFLOW(DATASETSIZE(2000000)) is set, DFSMShsm selects overflow ML1 volume with the least free space that accommodates the data set, which is SBXHS4, as shown in Example 5-27.

Example 5-27 LIST command output for MHLRES.ML1ENH after it migrated to ML1 volume

```
DSN=MHLRES4.ML1ENH                                  MIGVOL=SBXHS4 DSO=PS  SDSP=NO
  LAST REF=09/07/28 MIG=09/07/28 TRKS=0064500 2K BLKS= 1758651 TIMES MIG= 002
  16K BLKS=******  LAST MIGVOL=******

ARC0140I LIST COMPLETED,        3 LINE(S) OF DATA OUTPUT
COMMAND REQUEST 00000169 SENT TO DFSMSHSM
***
```

At this point:

- ML1 overflow volume SBXHS4 has 40,208 tracks of remaining free space.
- ML1 overflow volume SBXHS6 has 150,033 tracks of remaining free space.
- ML1 no overflow volume SBXHS5 has 119,360 tracks of remaining free space.

Run a BACKDS command against MHLRES4.ML1ENH2. DFSMShsm selects the ML1 overflow volume SBXHS6 to accommodate the backup copy, as this volume is the only ML1 overflow volume with enough free space. The result is shown in Example 5-28.

Example 5-28 LIST command output for MHLRES.ML1ENH2 after creating its backup version on ML1 volume

```
DSN=MHLRES4.ML1ENH2                                  BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T425519.MHLRES4.ML1ENH2.J9209      BACKVOL=SBXHS6   FRVOL=MLD10C
  BACKDATE=09/07/28  BACKTIME=19:55:42  CAT=YES  GEN=000  VER=001  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 TOTAL BACKUP VERSIONS = 0000000001

 ARC0140I LIST COMPLETED,        4 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000175 SENT TO DFSMSHSM
 ***
```

At this point:

- The ML1 overflow volume SBXHS4 has 40,208 tracks of remaining free space.
- The ML1 overflow volume SBXHS6 has 84,897 tracks of remaining free space.
- The ML1 no overflow volume SBXHS5 has 119,360 tracks of remaining free space.

Another BACKDS command issued against MHLRES4.ML1ENH2 will make DFSMShsm select ML1 overflow volume SBXHS6 as a candidate, as it still has enough free space to accommodate the new backup copy version. The result is shown in Example 5-29.

Example 5-29 LIST command output for MHLRES.ML1ENH2 after creating its second backup version on the ML1 volume

```
DSN=MHLRES4.ML1ENH2                                  BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***
```

```
  BDSN=HSM.BACK.T301120.MHLRES4.ML1ENH2.J9209      BACKVOL=SBXHS6   FRVOL=MLD10C
  BACKDATE=09/07/28  BACKTIME=20:11:30  CAT=YES  GEN=000  VER=002  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T425519.MHLRES4.ML1ENH2.J9209      BACKVOL=SBXHS6   FRVOL=MLD10C
  BACKDATE=09/07/28  BACKTIME=19:55:42  CAT=YES  GEN=001  VER=001  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 TOTAL BACKUP VERSIONS = 0000000002

 ARC0140I LIST COMPLETED,        6 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000177 SENT TO DFSMSHSM
 ***
```

At this point:

- The ML1 overflow volume SBXHS4 has 40,208 tracks of remaining free space.
- The ML1 overflow volume SBXHS6 has 19,761 tracks of remaining free space.
- The ML1 no overflow volume SBXHS5 has 119,360 tracks of remaining free space.

Another BACKDS command issued against MHLRES4.ML1ENH2 will make DFSMShsm select the ML1 no overflow volume SBXHS5 as a candidate, as all ML1 overflow volumes cannot satisfy the requested space. The result is shown in Example 5-30.

Example 5-30 LIST command output for MHLRES.ML1ENH2 after creating its third backup version on ML1 volume

```
DSN=MHLRES4.ML1ENH2                                  BACK FREQ = ***  MAX ACTIVE B
ACKUP VERSIONS = ***

  BDSN=HSM.BACK.T282120.MHLRES4.ML1ENH2.J9209      BACKVOL=SBXHS5   FRVOL=MLD10C
  BACKDATE=09/07/28  BACKTIME=20:21:28  CAT=YES  GEN=000  VER=003  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T301120.MHLRES4.ML1ENH2.J9209      BACKVOL=SBXHS6   FRVOL=MLD10C
  BACKDATE=09/07/28  BACKTIME=20:11:30  CAT=YES  GEN=001  VER=002  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

  BDSN=HSM.BACK.T425519.MHLRES4.ML1ENH2.J9209      BACKVOL=SBXHS6   FRVOL=MLD10C
  BACKDATE=09/07/28  BACKTIME=19:55:42  CAT=YES  GEN=002  VER=001  UNS/RET= NO
  RACF IND=NO  BACK PROF=NO  NEWNM=NO  NOSPH=*** GVCN=*** RETDAYS=*****

 TOTAL BACKUP VERSIONS = 0000000003

 ARC0140I LIST COMPLETED,        8 LINE(S) OF DATA OUTPUT
 COMMAND REQUEST 00000179 SENT TO DFSMSHSM
 ***
```

5.2.3 Secondary space management considerations

Secondary space management (SSM) checks data space utilization on ML1 volumes. If the limit (high threshold) for the percentage of occupied space (THRESHOLD) is reached or exceeded for at least one ML1 volume, then SSM starts ML1 to ML2 migration of all eligible data sets. If at least one ML1 volume has been marked full (full VTOC or VTOC index), SSM will start ML1 to ML2 migration even if the THRESHOLD value is not reached for all ML1 volumes. DFSMShsm marks the ML1 volume full, if the number of available DSCBs is equal or less then 5. The flag will be reset if the number of available DSCBs becomes greater then 5 again. This process attempts to correct the VTOC errors encountered on these volumes. The successfully processed volumes that were previously marked unavailable for ML1 volume selection because of the VTOC errors will then become re-eligible for ML1 volume selection again.

Note: In some cases, SSM cannot resolve the problem of the full VTOC errors. One instance of this situation is when not enough data sets are eligible for ML1 to ML2 migration (SMS set by Management Class LEVEL 1 DAYS NON-USAGE or NSMS set by the SETSYS MIGRATIONLEVEL1DAYS(xx) command. The FREEVOL command can be used to move migration copies to another ML1 volume or backup versions to another backup volume.

ABARS considerations

ABARS processes Large Format Sequential ML1 copies in V1.11 for ABACKUP's and ARECOVER's processing.

Note: An ARECOVER of ML1 data set with a migration copy larger than 64 K tracks will be failed in downlevel systems.

LIST command syntax

A new parameter has been added to list ML1 OVERFLOW/NOOVERFLOW volumes, as shown in Figure 5-28.

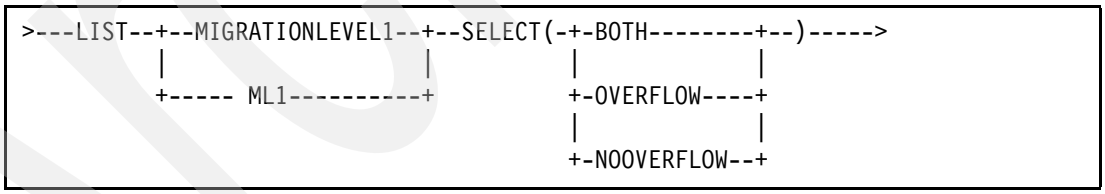


Figure 5-28 LIST command syntax for ML1 OVERFLOW/NOOVERFLOW volumes

Where:

- BOTH: Lists all ML1 volumes. This is the default value.
- OVERFLOW: Lists all OVERFLOW ML1 volumes.
- NOOVERFLOW: Lists all NOOVERFLOW ML1 volumes.

5.2.4 New and updated system messages

The messages shown in this section have been added or modified.

New messages

The message shown in this section is new.

ARC1030I

The new ARC1030I message is shown in Figure 5-29.

```
ARC1030I DATA SET dsname OF nnnn TRACKS ELIGIBLE FOR MIGRATION/BACKUP TO AN
OVERFLOW VOLUME HAS BEEN REDIRECTED UP TO A NOOVERFLOW VOLUME
```

Figure 5-29 New ARC1030I

Explanation: A data set with a size of nnnn tracks or nnnnK tracks that is eligible for migration or backup to an OVERFLOW ML1 volume has been redirected up to a NOOVERFLOW ML1 volume.

Updated messages

The messages shown in this section have been updated.

ARC0103I

New text has been added to the ARC0103I message to warn you about invalid SETSYS parameters, as shown in Figure 5-30.

```
ARC0103I INVALID SETSYS PARAMETER {ML1OVERFLOW DATASETSIZE | ML1OVERFLOW
THRESHOLD}
```

Figure 5-30 New parameters added to ARC0103I

The ML1OVERFLOW DATASETSIZE: DATASETSIZE parameter is outside the range of 1-704642988.

The ML1OVERFLOW THRESHOLD:THESHOLD parameter is outside the range of 1-100.

ARC0272I

When the QUERY SETSYS command is issued, a new field, ML1OVERFLOW, is added to the ARC0272I message, as shown in Figure 5-31.

```
ARC0272I PRIMARY SPACE MGMT CYCLE LENGTH=ndays
DAY(S),}, TODAY IS DAY=n, CYCLE START DATE=yy/mm/dd,
ARC0272I SECONDARY SPACE MGMT CYCLE LENGTH=ndays
DAY(S),}, TODAY IS DAY=n, CYCLE START DATE=yy/mm/dd,
ML1OVERFLOW(DATASETSIZE=dssizeK, THRESHOLD=threshold)
```

Figure 5-31 The new ML1OVERFLOW in the ARC0272I field

ARC0757I

New text (MIGRATION) has been added to the ARC0757I message, as shown in Figure 5-32.

```
ARC0757I [BACKUP | MIGRATION] OF A DATA SET OF tracks-used TRACKS TO ML1 VOLUME
volser FAILED FOR INADEQUATE SPACE
```

Figure 5-32 New ARC0757I format

Explanation: A data set migration or backup was requested and the available space on the volume (including ML1 overflow volumes, if they exist) was inadequate. Sufficient space was not available on the VOLSER; the ML1 was volume selected.

5.2.5 Migration considerations

Consider the following migration items:

- You should do an analysis of data sets that are larger than 64 K tracks and determine whether the installation wants them to migrate or back up to ML1 DASD. Subsequent SMS or NSMS adjustments for movement to DASD / tape are based on this analysis.
- You should evaluate the current use of ML1 OVERFLOW volumes:
  - Determine how the installation uses ML1 OVERFLOW and NOOVERFLOW volumes.
  - Determine how the installation uses ML1 OVERFLOW and NOOVERFLOW volumes.
  - Adjust the SETSYS ML1OVERFLOW parameters as needed.

5.2.6 Coexistence considerations

OA26330 provides new function to enables ML1 enhancements coexistence support on lower level installations. Down-level DFSMShsm installations (pre-V1.11) will be able to:

- Recall, recover, and arecover data sets from V1.11.
- Make DFSMShsm LFS migration copies or backup versions from NOOVERFLOW or OVERFLOW volumes.
- The following processing will fail when making a LFS migration copy or backup version:
  - L1 $\rightarrow$ L2 DASD migration
  - L1 $\rightarrow$ L1 migration copy moving
  - L1 $\rightarrow$ backup daily volume backup version moving AUDIT MEDIACONTROLS VOLUMES(volser). FIX command processing will support LFS data sets.

5.3 DFSMShsm fast replication enhancements

The following enhancements have been made in z/OS V1.11:

- Allows FRRECOV DSNAME to delete or move data sets.

  Pre-V1.11, FRRECOV DSNAME relies upon the current catalog information for a data set to locate the backup copy within the volume copies of a copy pool backup version. Data sets that have since moved or been deleted do have reliable catalog information to be able to locate the backup copy.

  If catalog information is captured at the time of an FRBACKUP request, that information can be used to locate the backup copy of the data set within a copy pool backup version and to recatalog the data set properly after recovery. The current catalog information for the data set is not required to be consistent with the information at the time of the FRBACKUP. This allows users to use FRRECOV for deleted or moved data sets.
- Automates Catalog unallocation during volume FRRECOV.

  When a catalog is allocated by the catalog address space (CAS), the volume on which the catalog resides is also allocated. FRRECOV requires volumes that are to be restored to be unallocated. In this release, DFSMShsm automatically unallocates the catalogs.

- ALLOWPPRCP settings can be used in the SMS copy pool definition.
- Initializes volumes that have been withdrawn from a FlashCopy relationship.

5.3.1 FRRECOV DSNAME enhancement

In pre-z/OS V1.11 DFSMS, users cannot using FRRECOV for data sets that have been deleted or moved because there was not a catalog entry that was representative of the state of the data set at backup time. In V1.11, DFSMShsm now supports this feature. The solution is to capture the catalog information of all data sets backed up during FRBACKUP.

The ISMF panel has been updated to allow up to 10 catalogs names to be specified, as shown in Figure 5-33.

```
Panel  Utilities  Scroll  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              COPY POOL DEFINE                 Page 2 of 5
Command ===>

SCDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name : NEWCPOOL

To DEFINE Copy Pool, Specify:
  Catalog Name ==>
               ==>
               ==>
               ==>
               ==>
               ==>
               ==>
               ==>
               ==>
               ==>

  Capture Catalog Information for Data Set Recovery . . . N   (R, P or N)

Use ENTER to Perform Verification; Use UP/DOWN Command to View previous Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 5-33 ISMF new copy pool panel

However, all of the catalogs specified in the copy pool 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 in which it is defined.

You can set the “Capture Catalog Information for Data Set Recovery” field set to Required or Prefer or No (R|P|N):

- Specifying R (required) has the effect of invalidating the backup if not all the catalog information could be captured.
- When P (preferred) is specified and not all the catalog information could be captured, the backup version will be valid if all the volumes were successfully backed up.
- Specifying NO on this new field will disable the recovery of moved or deleted data sets. If the field is not set, it will remain blank and be considered the same as N.

At FRBACKUP time, Catalog Search Interface (CSI) is invoked for each catalog specified in the copy pool definition, and the information is saved in a data set called Catalog Information Data Set (CIDS), which is created on an ML1 volume, but will not be treated as a migrated data set.

There is one CIDS data set per copy pool version. These data sets are managed by DFSMShsm and will be deleted when their associated version has been withdrawn, rolled off, or deleted. They will not be eligible for SDSPs or ever migrate to ML2.

The naming convention for the CIDS data sets is:

```
Bprefix.HSMCIDS. Dyyddd.Thhmmss.Cnumseq
```

Where:

- Bprefix is the prefix you specified with the BACKUPPREFIX parameter of the SETSYS command. Otherwise, BPREFIX is the DFSMShsm default UID.
- HSMCIDS, T, V, D, and C are constants.
- hhmmss is the time in hours, minutes, and seconds that the Fast Replication backup was taken.
- yyddd is the year and day the backup was taken.
- numseq is the qualifier that ensures the data set name is unique. num is a three digit number from 000 to 255 that will be increased by 1 in the event of a data set name collision. seq is a two digit number that is generated based on the sequence of catalog names in the list returned via the SMS copy pool information retrieval.

By default, DFSMShsm allocates a CIDS with 50 primary and 50 secondary cylinders. However, a user can change the default CIDS data set size by using the PATCH command:

- To change the default primary allocation, use the code in Example 5-31, which changes the primary allocation from 50 cylinders to 100 cylinders.

  Example 5-31 PATCH command to change the primary allocation from 50 cylinders to 100 cylinders

```
PATCH .FRGCB.+30C X'00000064'  VERIFY(.FRGCB.+30C X'00000032')
```

- To change the default secondary allocation, use the code in Example 5-32.

  Example 5-32 PATCH command to change the secondary allocation from 50 cylinders to 100 cylinders

```
PATCH .FRGCB.+310 X'00000064'  VERIFY(.FRGCB.+310 X'00000032')
```

When the FRRCOV DSNAME command is issued, DFSMShsm will check if it exists for the FRBACKUP version. CIDS is scanned for supported data sets and:

- If a match is found, the current catalog is checked against the catalog information found in the CIDS. If different, information from the CIDS is used. In this case, the moved data set will be re-allocated and re-cataloged to the state at which it existed at backup time.
- If a match is not found, the data set must be currently cataloged and moved to the same volumes to recover the data set.

If a Storage Group is defined in several copy pools, when those copy pools are backed up, the data sets on those volumes will have several backup copies. To recover those data sets, you must specify which copy version you want to recover by using the keyword FROMCOPYPOOL(copypoolname). In this release, this keyword is also required when a fully qualified deleted data set is specified.

Wild card support

In pre-z/OS V1.11 DFSMS, users are allowed to use wild card for the FRRECOV command. However, in V1.11, you cannot use a wild card in the FRRECOV command to recover deleted or uncataloged data sets. In order to recover them, you must specify a fully qualified data set name.

When recovering all data sets belonging to a copy pool, use wild cards first for all cataloged data sets and then specify fully qualified data sets for deleted data sets.

LIST command

LIST COPYPOOL will be updated to list all data sets that were backed up with a specified version. The new DATASETS keyword will be one of the mutually exclusive required parameters of the COPYPOOL(cpname) keyword. You can use the list output to find data sets that were backed up but have been deleted. Data sets will be listed in alphabetical order by catalog.

An example output of LIST COPYPOOL(xx) DATASET(GENERATION(0)) showing all the catalog information of all data sets during FRBACKUP is shown in Example 5-33.

Example 5-33 LIST COPYPOOL sample output

```
********************************* Top of Data ********************************

-- DFSMShsm CONTROL DATA SET -- COPY POOL -- LISTING -- AT 18:59:46 ON 09/07/30

COPYPOOL=CP2                          ,VER=003,GEN=000,CATINFO=P
CATALOG INFORMATION DATA SET NAME=HSM.HSMCIDS.D09211.T185924.C001            ,
TKN(C)=C''
TKN(H)=X''

CATALOG NAME = UCAT.VSBOX01

DATA SET NAME
$DSN810.APPLY.OUTPUT
$DSN810.SDSNLINK
$DSN910.SDSNLINK
.........................
LOGR.IGWTV065.IGWLOG.SYSLOG.SC65
LOGR.IGWTV065.IGWSHUNT.SHUNTLOG.A0000000

TOTAL NUMBER OF DATA SETS =       5829
******************************** Bottom of Data ******************************
```

AUDIT command

AUDIT COPYPOOLCONTROLS is updated to verify that each CIDS has the corresponding copy pool.

The AUDIT DIRECTORYCONTROLS VOLUMES(volser) can be used to audit an ML1 volume. When it encounters a catalog information data set on an ML1 volume, DFSMShsm verifies if the data set is cataloged and determines if the volume in the catalog entry matches the volume being processed. If there are no errors, DFSMShsm determines if the catalog information data set corresponds to a valid copy pool backup version.

FREEVOL command

The FREEVOL command for ML1 volumes has been modified to move CIDS in a similar manner as VTOC copy data sets are moved currently.

FREEVOL processing move CIDS on the freed volume to other migration level 1 volumes, regardless of the target level of the FREEVOL command.

FRRECOV command

A new ALLOWPPRC parameter has been added, as shown in Example 5-34.

Example 5-34 FRRECOV syntax

```
>>--FRRECOV------COPYPOOL---(cpname)----------------------------------------------------->
               | |      |              |         _Y_    |                                |
               | ---CP---              |        |   |   |                                |
               |                       --VERIFY(---N---)--                               |
               |                                                                         |
               |         _,______                                                        |
               |        |        |                                                       |
               |        V        |                                                       |
               |-DSNAME(---dsname---)----------------------------------------------------|
               |                      |         | |              | |                | |
               |                      --REPLACE-- --| Option A |--- --| Option B |---- |
               |-TOVOLUME(volser)--------------------------------------------------------|
                                  |                |
                                  --| Option A |----

>---------------------------------------------------------------------------------------->
  |                     |  |                                               |
  --DATE(yyyy/mm/dd)-----  --ALLOWPPRC---(---NO----------------------------)--
  |                     |                  |                            |
  --GENERATION(gennum)---                  ----PRESERVEMIRRORNO-----------
  |                     |                  | |                  |       |
  --TOKEN(token)---------                  | --PMNO--------------       |
  |                     |                  | |                  |       |
  --VERSION(vernum)------                  | --YES---------------       |
                                           |                            |
                                           ----PRESERVEMIRRORPREFERRED----
                                           | |                        | |
                                           | --PMPREF------------------- |
                                           ----PRESERVEMIRRORREQUIRED-----
                                             |                        |
                                             --PMREQ--------------------
```

Where:

- PMREQ (PR): Specifies that a Metro Mirror primary volume may become a FlashCopy target, and the Metro Mirror pair must not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, the FlashCopy should not be attempted. If the target volume is not a Metro Mirror primary volume, this parameter has no effect.
- PMPREF (PP): Specifies that a Metro Mirror primary volume may become a FlashCopy target and it would be preferable that the Metro Mirror pair does not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, the FlashCopy operation is still attempted. If the intended FlashCopy target volume is not a Metro Mirror primary volume, this parameter has no effect.

- PMNO (PN): Specifies that a Metro Mirror primary volume may become a FlashCopy target, and the FlashCopy should be performed without considering a preserve mirror operation. If the target volume is not a Metro Mirror primary volume, this parameter has no effect.
- NO: Specifies that a Metro Mirror primary volume should not become a FlashCopy target. This is the default.

Updated ISMF panels

The panels in this section have been modified to support DFSMShsm Fast Replication enhancement.

DEFINE and ALTER COPY POOL panels

New fields for ALLOWPPRC have been added to the DEFINE/ALTER panel, as shown in Figure 5-34:

- FRBACKUP to PPRC Primary Volumes: Indicates DFSMShsm will target PPRC Primary volumes if they are available for FRBACKUP processing.
- FRRECOV to PPRC Primary Volumes: Indicates that DFSMShsm will target PPRC Primary volumes if they are available for FRRECOV processing.

```
Panel  Utilities  Scroll  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              COPY POOL DEFINE                  Page 1 of 5
Command ===>

SCDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name : TEST

To DEFINE Copy Pool, Specify:
  Description ==>
              ==>
  Auto Dump . . . N   (Y or N)        Dump Sys/Sys Group Name . . .
  Dump Class  . .                     Dump Class  . .
  Dump Class  . .                     Dump Class  . .
  Dump Class  . .

  Number of Recoverable DASD Fast
   Replicate Backup Versions  . . . . 2       (0 to 85 or blank)
  FRBACKUP to PPRC Primary Volumes allowed . . .     (NO, PN=PMNO, PP=PMPREF,
                                                      PR=PMREQ or blank)
  FRRECOV to PPRC Primary Volumes allowed  . . .     (NO, PN, PP, PR or blank)
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 5-34 New fields in the COPY POOL DEFINE/ALTER panel

Where:

– PMREQ (PR): Specifies that a Metro Mirror primary volume may become a FlashCopy target, and the Metro Mirror pair must not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, the FlashCopy should not be attempted. If the target volume is not a Metro Mirror primary volume, this parameter has no effect.

– PMPREF (PP): Specifies that a Metro Mirror primary volume may become a FlashCopy target and that it is preferable that the Metro Mirror pair does not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, the FlashCopy operation should be attempted. If the intended FlashCopy target volume is not a Metro Mirror primary volume, this parameter has no effect.

– PMNO (PN): Specifies that a Metro Mirror primary volume may become a FlashCopy target, and the FlashCopy should be performed without considering a preserve mirror operation. If the target volume is not a Metro Mirror primary volume, this parameter has no effect.

– NO: Specifies that a Metro Mirror primary volume should not become a FlashCopy target. This is the default.

– Blank, none specified.

COPY POOL DISPLAY panels

New fields for ALLOWPPRC for FRBACKUP and FRRECOV have been added to the COPY POOL DISPLAY panel, as shown in Figure 5-35.

```
 Panel  Utilities  Scroll  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              COPY POOL DISPLAY                 Page 1 of 5
Command ===>

CDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name: CP2

Description : COPY POOL 2

Auto Dump  . . . : N
Dump Class . . . :
Dump Class . . . :
Dump Class . . . :
Dump Class . . . :
Dump Class . . . :
Dump Sys/Sys Group Name :
Number of Recoverable DASD Fast Replicate Backup Versions : 2
FRBACKUP to PPRC Primary Volumes allowed :
FRRECOV to PPRC Primary Volumes allowed :


Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 5-35 COPY POOL DISPLAY panel 1

The second panel shows ten Catalog Names and the capture data set catalog information field, as shown in Figure 5-36.

```
Panel  Utilities  Scroll  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                             COPY POOL DISPLAY                Page 2 of 5
Command ===>

CDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name: CP2

Catalog Name : UCAT.VSBOX01
Catalog Name :
Catalog Name :
Catalog Name :
Catalog Name :
Catalog Name :
Catalog Name :
Catalog Name :
Catalog Name :
Catalog Name :
Capture Catalog Information for Data Set Recovery : P



Use UP/DOWN Command to View previous Panel;
Use HELP Command for Help; Use END Command to Exit.
```

Figure 5-36 COPY POOL DISPLAY panel 2

Also, the COPY POOL LIST panel has been updated to show FCTOPPRC FRBACKUP (column 14), FCTOPPRC FRRECOV (column 15), and CATINFO (column 16), as shown in Figure 5-37.

```
Panel  List  Utilities  Scroll  Help
--------------------------------------------------------------------------------
                                COPY POOL LIST
Command ===>                                                Scroll ===> PAGE
                                                            Entries 1-1 of 1
                                                            View in Use
CDS Name : SYS1.SMS.SCDS

Enter Line Operators below:

     LINE                                    FCTOPPRC  FCTOPPRC
     OPERATOR          COPY POOL NAME        FRBACKUP  FRRECOV   CATINFO
    ---(1)----  -------------(2)------------- --(14)--  --(15)--  --(16)---
                CP2                           ------    ------    PREFERRED
    ----------  ------------------------------ ------  BOTTOM  OF  DATA
------
```

Figure 5-37 COPY POOL LIST new fields

NaviQuest support

NaviQuest is a testing and reporting tool that speeds and simplifies the tasks that are associated with DFSMS initial implementation and ongoing ACS routine and configuration maintenance. NaviQuest provides the following functions:

- Familiar ISPF panel interface functions
- Fast, easy, and bulk test case creation
- ACS routine and DFSMS configuration testing automation
- Storage reporting assistance
- Additional tools to aid with storage administration tasks
- Batch creation of data set and volume listings
- Printing of ISMF LISTs
- Batch ACS routine translation
- Batch ACS routine validation

In z/OS V1.11, the NaviQuest JCL for the DEFINE, ALTER, DISPLAY copy pool will be modified to support the new enhancements. Example 5-35 shows the JCL to define or alter the copy pool.

Example 5-35 Sample NaviQuest JCL to define or alter the copy pool

```
//MHLRES4  JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES4
 //*
 //STEP1   EXEC ACBJBAOB,
 //        TABL2=MHLRES4.ISPTABL
 //SYSUDUMP DD  SYSOUT=*
 //SYSTSIN  DD *
 PROFILE PREFIX(IBMUSER)
 ISPSTART CMD(ACBQBAP1 +
 DEFINE/ALTER +
 SCDS(TEST.CDS) +
 COPYPOOL(CPPOOL1) +
 DESCR(DEFINE COPY POOL CPPOOL1) +
 NUMVERS(2) +
 AUTODUMP(N) +
 DMPSYSNM() +
 DUMPCLAS() +
 CATNAME() +
 CATINFO() +
 FCTOPPRCB(N) +
 FCTOPPRCR(N) +
 POOLSGNM() +
 UPDHLVLSCDS() +
 ) +
 BATSCRW(132) BATSCRD(27) BREDIMAX(3) BDISPMAX(999999)
 /*
```

New messages

The following messages have been added in z/OS V1.11.

ARC1812I

Whenever an operation is requested against a CIDS and there is an error, the ARC1812I message will be issued, as shown in Figure 5-38.

```
ARC1812I OPERATION=op FOR CATALOG INFORMATION DATA SET=dsn FOR COPY POOL=cpname
HAS COMPLETED WITH RETURN CODE=rc AND REASON CODE=rsn
```

Figure 5-38 ARC1812I

ARC1838I

A new message is generated when a volume withdrawn from FlashCopy initializes, as shown in Figure 5-39.

```
ARC1838I INITIALIZATION ATTEMPT OF COPY POOL BACKUP STORAGE GROUP VOLUME volser
HAS COMPLETED, RC= rc,RSN = rsn
```

Figure 5-39 ARC1838I

Updated messages

The following messages have been modified in z/OS V1.11.

ARC1803E

The ARC1803E message is issued after fast replication backup or recovery has completed and lists volumes that were not backed up or recovered successfully. The ARC1803E message will be modified to also specify a failing DSS message number along with each volume name. It will not always be possible to associate a DSS error message with a failed volume. In those cases, a DSS error is reported for that volume as ADR****, as shown in Figure 5-40.

```
ARC1803E THE FOLLOWING numvols VOLUME(S) FAILED DURING FAST REPLICATION {BACKUP
| RECOVERY} OF COPY POOL cpname

volser : volser indicates the name of the volume that failed
ADRnnns: ADR**** indicates that an associated failing message could not be
identified.
```

Figure 5-40 ARC1803E

ARC0550I

ARC0550I has been updated to indicate the CIDS data set encountered during the FREEVOL process, as shown in Figure 5-41.

```
ARC0550I count DATA SETS ON VOLUME volser, WERE NOT PROCESSED BY FREEVOL

Fast Replication catalog information data sets, identified by the second
qualifier of HSMCIDS, cannot be moved by a pre-V1.11 DFSMShsm.
```

Figure 5-41 ARC0550I

ARC0734I

A new keyword (ACTION=MOVCIDS) has been added, as shown in Figure 5-42.

```
ARC0734I ACTION=action FRVOL=volser1 TOVOL=volser2 | TRACKS=tracks |
RC=retcode, REASON=reascode, AGE=days, DSN=dsname

MOVCIDS – Move a catalog information data set to a migration level 1 volume
during the processing of a FREEVOL command.
```

Figure 5-42 ARC0734I

ARC0814I

New return codes (18, 19, 20, 21, and 24) have been added, as shown in Figure 5-43.

```
ARC0814I LIST {USER | TTOC | DATA SETS | VOLUME | BACKUPVOLUME | PRIMARYVOLUME
| DUMP VOLUMES | DUMP CLASSES | AGGREGATE | COPYPOOL} PROCESSING TERMINATING
EARLY, RC=retcode
```

Figure 5-43 ARC0814I

Where:

RC18 LIST COPYPOOL DATASETS was requested to write output to the terminal. The TERMINAL parameter is not supported for the DATASETS keyword.

RC19 Copy pool records indicate that catalog information was not captured for this backup version.

RC20 Copy pool records indicate that catalog information was captured for this backup version, but the Fast Replication Catalog Information Data Set was not found.

RC21 The version for the specified GENERATION or TOKEN could not be found.

RC24 An internal error occurred; LIST could not continue.

ARC1605I

A new return code (42) for the FRBACKUP command has been added, as shown in Figure 5-44.

```
ARC1605I COMMAND HAD PARSE ERROR
```

Figure 5-44 ARC1605I

Where:

RC42 The number of RETAINDAYS is a required parameter that must be an integer in the range of 0 to 50000, or 99999. RETAINDAYS values between the range of 50001 and 99998 are not accepted. A RETAINDAYS value of 99999 is managed as a ‘never expire’ value.

ARC1802I

A new message field (CAPTURE CATALOG RC=) has been added, as shown in Figure 5-45.

```
ARC1802I FAST REPLICATION {BACKUP | PREPARE | RECOVERY} HAS COMPLETED FOR {COPY
POOL cpname | VOLUME volser}, AT time ON date, FUNCTION RC=retcode, MAXIMUM
VOLUME RC=max vol retcode, {CAPTURE CATALOG RC=capture catalog retcode}
```

Figure 5-45 ARC1802I

ARC1806E

A new return code (70) has been added, as shown in Figure 5-46.

```
ARC1806E FAST REPLICATION {BACKUP | RECOVERY | *****} HAS FAILED FOR {COPY POOL
cpname | VOLUME volser | *****}, RC=retcode
```

Figure 5-46 ARC1806E

Where:

RC70 A capture catalog was requested as REQUIRED in the copy pool definition. The catalog capture function did not complete successfully.

ARC1866I

New return codes (48 and 50) have been added, as shown in Figure 5-47.

```
ARC1866I FAST REPLICATION {RECOVERY} HAS FAILED FOR {DATA SET dsname},
RC=retcode
```

Figure 5-47 ARC1866I

Where:

RC48 The REPLACE keyword was not specified on the FRRECOV command.

RC50 The FROMCOPYPOOL keyword was not specified for a data set that is not currently cataloged.

5.3.2 Automating catalog unallocation

When a catalog is allocated by the catalog address space (CAS), the volume on which the catalog resides is also allocated. FRRECOV requires volumes that are to be restored to be unallocated. Often, using FRRECOV on a copy pool fails because this manual action has been overlooked.

With the new enhancement, all catalogs associated with a copy pool will be specified in the copy pool definition, DFSMShsm automatically unallocates the specified catalogs prior to the recovery of a copy pool. If the unallocation of any catalogs is unsuccessful, FRRECOV processing will fail. The catalog reallocates the catalogs when used.

When catalog names are specified in the SMS copy pool definition, FRRECOV requests that the catalog address space to unallocate the catalogs. DFSMShsm retrieves the copy pool definition from SMS. If the copy pool definition indicates that the catalogs should be unallocated, DFSMShsm issues an “F CATALOG,UNALLOCATE(catname)” for each catalog name that is specified in the copy pool definition at the time of the FRRECOV.

In the event that an unallocate request has not completed, has failed, or the catalog has been re-allocated, the user will continue to observe FRRECOV failures.

This enhancement is not applicable to the FRRECOV DATASET command. because volume restores are not performed for data set recovery.

5.3.3 ALLOWPPRC settings in COPY POOL definition

In pre-z/OS V1.11 DFSMS, you can select whether to allow Fast Replication backup and recover copy pool volumes to put PPRC primary volumes in a duplex pending state via the FRBACKUP and FRRECOV commands. The ALLOWPPRCP(YES|NO) keyword will remain and be expanded to the ALLOWPPRCP(NO|PMNO|PMPREF|PMREQ) keyword so that the SMS copy pool PPRCP settings may be overridden.

```
Panel  Utilities  Scroll  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              COPY POOL DEFINE                  Page 1 of 5
Command ===>

SCDS Name  . . : SYS1.SMS.SCDS
Copy Pool Name : TEST

To DEFINE Copy Pool, Specify:
  Description ==>
              ==>
  Auto Dump . . . N   (Y or N)        Dump Sys/Sys Group Name . . .
  Dump Class  . .                     Dump Class  . .
  Dump Class  . .                     Dump Class  . .
  Dump Class  . .

  Number of Recoverable DASD Fast
   Replicate Backup Versions  . . . . 2       (0 to 85 or blank)
  FRBACKUP to PPRC Primary Volumes allowed . . .     (NO, PN=PMNO, PP=PMPREF,
                                                      PR=PMREQ or blank)
  FRRECOV to PPRC Primary Volumes allowed  . . .     (NO, PN, PP, PR or blank)
Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 5-48 New fields to support ALLOWPPRC in copy pool definition

Where:

- PMREQ (PR): Specifies that a Metro Mirror primary volume may become a FlashCopy target, and the Metro Mirror pair may not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, the FlashCopy should not be attempted. If the target volume is not a Metro Mirror primary volume, this parameter has no effect.

- PMPREF (PP): Specifies that a Metro Mirror primary volume may become a FlashCopy target and it would be preferable that the Metro Mirror pair does not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, the FlashCopy operation should still be attempted. If the intended FlashCopy target volume is not a Metro Mirror primary volume, this parameter has no effect.
- PMNO (PN): Specifies that a Metro Mirror primary volume may become a FlashCopy target, and that the FlashCopy should be performed without considering a preserve mirror operation. If the target volume is not a Metro Mirror primary volume, this parameter has no effect.
- NO: Specifies that a Metro Mirror primary volume should not become a FlashCopy target. This is the default.

5.3.4 Initializing FlashCopy withdraw volume

DFSMShsm withdraws volume FlashCopy relationships automatically in a NOCOPY environment and also allows users to withdraw relationships for a copy pool version via the FRBACKUP command. DFSMShsm invokes SDM to complete this task. When a withdraw is invoked in this manner, the track by track copy is stopped. It is possible that the pointer to the VTOC starting location and the physical location of the VTOC itself might become out of sync. In this situation, the VTOC becomes inaccessible, resulting in an error when the VTOC is subsequently accessed. The volume must be re-initialized to create a new VTOC. Be certain that the following tasks occur:

1. Copy pool backup Storage Group volumes that might have a corrupted VTOC will be re-initialized.
2. Prior to withdraw processing, DFSMShsm performs an obtain for the Format 4 of the source volume. From the Format 4, the tracks on which the VTOC can reside can be determined.
3. SDM is then invoked to query those tracks on the target volume to determine if the tracks on which the VTOC resides are in an active FlashCopy relationship.
4. If the VTOC tracks are in a relationship, DFSMShsm invokes ICKDSF to re-initialize the volume, which also withdraws any Flash Copy relationships.
5. You still have to vary the volume offline and back online for other systems in the sysplex that access the volume as shared. The ARC1838I message notifies users that this action should be taken, as shown in Figure 5-49.

```
ARC1838I INITIALIZATION ATTEMPT OF COPY POOL BACKUP STORAGE GROUP VOLUME volser
HAS COMPLETED, RC= rc,RSN = rsn
```

Figure 5-49 ARC1838I message

5.3.5 Migration considerations

- Users have to note which catalogs are currently within each copy pool and update their copy pool definition with the catalog names.
- Automation of the unallocation of the copy pool catalogs prior to a copy pool recover is no longer needed.
- Stop using the YES option in the ALLOWPPRCP keyword of the FRBACKUP and FRRECOV commands and use options PMNO, PMREQ, and PMPREF instead.
- Ensure that there is adequate free space on ML1 volumes to accommodate CIDS.

5.3.6 Coexistence considerations

Keep the following coexistence considerations in mind:

- A pre-z/OS V1.11 DFSMShsm backs up a copy pool that has been defined in V1.11, but fails the command if the SMS copy pool definition indicates that the catalog information should be captured. The ARC1806E RC74 message is issued whenever the SMS copy pool definition specifies REQUIRED for capturing catalog information at backup time.
- Catalogs listed in the copy pool definition are not unallocated during copy pool recovery when using a pre-V1.11 DFSMShsm.
- FREEVOL processing on a pre-V1.11 DFSMShsm does not process catalog information data sets. When a catalog information data set is encountered during FREEVOL processing, the FREEVOL function fails. The ARC0550I message informs the user that the catalog information data sets found on the volume could not be moved and the FREEVOL function should be run on a V1.11 or later system to ensure that the data sets are processed.

5.3.7 Coexistence maintenances

Consider the following coexistence maintenances:

OA24697

This is a new function for DFSMShsm Fast Replication users in a mixed sysplex environment where a V1.11 system co-exists with V1.10 and V1.9 releases.

OA24698

This is a new function related to ISMF for DFSMShsm Fast Replication in a mixed sysplex environment with a V1.9 system.

OA26328

This is new function toleration for DFSMShsm Fast Replication users in a mixed sysplex environment where a V1.11 system co-exists with V1.10 and V1.9 releases.

5.4 DFSMShsm and Extended Address Volume (EAV)

With z/OS V1.10 and higher releases, z/OS has added support for DASD volumes having more than 65,520 cylinders.

The capacity of storage volumes has increased steadily over the years within the constraints of the 3390 storage device architecture. To expand the capacity of DASD storage volumes beyond 65520 cylinders, z/OS had to extend the track address format, which is why the term Extended Address Volume (EAV) denotes a volume of more than 65,520 cylinders.

The new DASD volume maximum for V1.10 and V1.11 is 262,668 cylinders.

In z/OS 1.11, the EAV support in DFSMShsm is enhanced to include sequential extended format data sets.

In order to allow users to control the migration of non-VSAM data sets to EAS, a new data set level attribute is defined. Each data set has a new attribute, EATTR, to indicate whether the data set can support extended attributes (that is, Format 8 and Format 9 DSCBs). By definition, a data set with extended attributes can reside in EAS on an EAV. This attribute may be specified for non-VSAM data sets as well as VSAM data sets.

This attribute allows for the following values:

- EATTR = NS: Extended attributes not specified.
- EATTR = NO: No extended attributes. Allocation may select a volume with or without extended attributes (DCEEADSCB=OFF). There should be no preference based on this attribute. However, when created, the data set has a F1 DSCB and cannot have extents in the EAS. This is the default for non-VSAM data sets.
- EATTR = OPT: Extended attributes are optional. The data set can have extended attributes and can reside in EAS. This is the default for VSAM data sets.
- EATTR = REQ: Extended attributes are required. The data set must have extended attributes and can reside in EAS. This option is not be valid in V1.11, but DFSMShsm supports this value in volume selection.

5.4.1 SMS volume selections

For SMS volume selection on recalls, recovery or ARECOVER's DFSMShsm pass the DATACLASS value to DSS and volume selection is performed by SMS. The only exception is for PDS data sets; the DFSMShsm volume selection logic for SMS PDS data sets is consistent with the non-SMS volume selection logic.

5.4.2 Non-SMS volume selections

DFSMShsm updates the volume selection logic to take into account the EATTR specified for a particular data set upon recall, recover, or arecover.

5.4.3 Recall, recover, and ARECOVER functions for non-VSAM data sets

DFSMShsm enables all the EAV volume spaces to be used when they are defined as ML1 and ML2 volumes.

For non-SMS data sets, recall and recover volume selection will be modified to select the appropriate type of volume based on a data set's EATTR attribute as follows:

- EATTR = NS: The data set has no EAS data set attributes specified. The default data set preferences for each data set type are used as a model for the EATTR value. The default for VSAM data sets is OPT, and the default for all non-VSAM data sets is NO. The preference for volumes corresponds to the appropriate default EATTR values for each data set.
- EATTR = NO: The data set has no preference for any volume. The volume with the most amount of free space is selected (either EAV or non-EAV), but only the track-managed free space is used for determining the free space calculations of an eligible EAV. If none are available, then the function fails.

  The data set is allocated with a Format 1 DSCB, indicating that it is not eligible to have extents in the EAS. If an EAV is chosen for allocation and the backup or migration copy of this data set has Format 8 or Format 9 DSCBs, then the data set will be allocated with a Format 1 DSCB.
- EATTR = OPT: The data set gives preference to EAV volumes first if the size of the allocation is equal to or greater than the BreakPointValue (BPV) specified. If the requested size of the data set allocation is smaller than the BPV, then EAVs and non-EAVs will have equal preference. If an EAV is chosen for allocation and the backup or migration copy of this data set has Format 8 or Format 9 DSCBs, then the data set will be allocated with Format 8 or Format 9 in the track-managed space
- “EATTR = REQ: The data set only gives preference to EAVs. The data set is allocated on the EAV with the most amount of available free space. If the allocation size of the data set is greater than or equal to the BPV, then the data set is allocated in the cylinder-managed space. If the allocation size of the data set is smaller than the BPV, then the data set is allocated in the track-managed space. In either scenario, the data set is Format 8 or Format 9.

If a data set is being directed to a pre-allocated data set on a recovery or ARECOVER, the EATTR value of the pre-allocated data set takes preference over the existing EATTR value of that data set. If no EATTR value exists for the pre-allocated data set, then the EATTR = NS value is assumed.

5.4.4 DFSMShsm Journal data set

Large and extended format sequential, PDSE, and VSAM data sets can occupy an entire EAV with multiple extents. Beginning in z/OS V1.11, DADSM allows a single extent to span the track-managed and cylinder-managed space. Therefore, it is also possible for a data set to occupy the entire volume with a single extent. This means that the DFSMShsm Journal can potentially occupy an entire volume. The DFSMShsm Journal can be defined as a LFS Journal (with CONTIG specified) and reside on an entire EAV volume in a single extent, because DADSM allows data sets to straddle the EAS line in V1.11.

5.4.5 Autodump considerations

With the introduction of large capacity volumes such as EAV, users who use dump stacking should review DUMP CLASS and its Dump Stack figure.

A Storage Group of different capacity volumes (Mod3, Mod9, Mod27, Mod54, and EAV) might require the DUMP window to be extended in order for DFSMShsm to complete the full volumes dump.

A dump task with the same number of dump stacking, where all the volumes to be dumped are high capacity volumes (such as mod27, Mod54, or EAV), requires more time to complete than a dump task that only dumps all mod3 volumes using the same stacking figure.

Group the same capacity volumes into the same dump class and assign the appropriate dump stacking figure (the higher the volume's capacity, the smaller dump stacking figure) to avoid the differential between dump tasks. You do not need to extend the dump window.

5.4.6 Coexistence considerations

z/OS V1.10 and V1.9 DFSMShsm need coexistence to handle the additional EAS-Eligible data sets and the new EATTR data set attribute.

APAR OA27146 provides DFSMShsm toleration for EAV, which applies to all users of z/OS DFSMShsm in an HSMplex environment with V1.11 DFSMShsm and V1.10 or V1.9 DFSMShsm levels.

Consider the following items:

- Non-VSAM data sets that have Format 8 or Format 9 DSCBs will be visible to the system, but not be eligible for migration, backup, or ABACKUP processing. When one of these data sets is encountered, DFSMShsm fails it. The ARC1220I message for migration and the ARC1320I message for backup with RSN21 or ARC0734I with RC20 and RSN21 are issued.

  Example 5-36 shows the error message when we try to migrate EAV data set CVERNON.EAVTEST.EFSAM0 on V1.10 level system.

  Example 5-36 ARC1220I message

```
ARC1001I CVERNON.EAVTEST.EFSAM0 MIGRATE FAILED, RC=0020, REAS=0021
ARC1220I DATA SET NOT ELIGIBLE FOR MIGRATION
***
```

  Example 5-37 shows the error message when we try to back up the EAV data set CVERNON.EAVTEST.EFSAM0 on V1.10 level system.

  Example 5-37 ARC1320I message

```
ARC1001I CVERNON.EAVTEST.EFSAM0  BACKDS FAILED, RC=0020, REAS=0021
ARC1320I DATA SET NOT ELIGIBLE FOR BACKUP
COMMAND REQUEST 00000072 SENT TO DFSMSHSM
***
```

- Recall/Recovery of non-VSAM data sets that have Format 8 or Format 9 DSCBs will be completed on the track managed space, so DFSMSdss restores them, ADR556W is issued, and DFSMShsm issues the ARC0784I message. However, the recall fails if there is no track managed space volume available to satisfy the space allocation request.

In Example 5-38, the CVERNON.EAVTEST.EFSAM0 data set originally resided on EAV volume GKDD65, which is the only volume in Storage Group EAVGK. When we try to recall the data set on a V1.10 level system, DFSMShsm tries to allocate the data set on a volume with track managed space, but none of them are available to Storage Group EAVGK. Therefore, DFSMShsm fails the recall, as shown in Example 5-38.

Example 5-38 Failed recall due to no available track managed space volume

```
ARC1001I CVERNON.EAVTEST.EFSAM0  RECALL FAILED, RC=0069, REAS=0709
 ARC1169I RECALL/RECOVER FAILED DUE TO AN ERROR IN DFDSS
  RECALL REQUEST 00000073 SENT TO DFSMSHSM
 ***
```

- Recover REPLACE of non-VSAM data sets fails if a pre-allocated non-VSAM data set is found with a Format 8or Format 9 DSCB. The ARC1158I message is issued. The ARC1001I or ARC0734I messages give the data set name and reason code 51. If a pre-allocated non-VSAM data set is found with an Format 1 DSCB, the EATTR value is passed to DFSMSdss to preserve the value.

  Example 5-39 shows the ARC1158I message when we try to recover the CVERNON.EAVTEST.EFSAM0 data set to replace the existing data set, which has Format 8 or Format 9 DSCBs on a V1.10 level system.

Example 5-39 ARC1158I message

```
ARC1001I CVERNON.EAVTEST.EFSAM0 RECOVER FAILED, RC=0058, REAS=0051
ARC1158I RECOVER DATA SET FAILED
COMMAND REQUEST 00000074 SENT TO DFSMSHSM
***
```

6

VSAM and Record Level Sharing enhancements

In this chapter, we discuss the enhancements to VSAM and Record Level Sharing. These enhancements include:

- The new VSAM data trap
- The VSAM trace improvements
- The VSAM SMB performance improvement
- The catalog health check feature, which is used to detect IMBED and REPLICATE

6.1 VSAM data trap

The VSAM data trap is a RAS enhancement for VSAM record management. VSAM Record Management checks all the data component CIs before they are written to DASD. If they are in error, they are not written. Data CIs are checked for integrity and if there is any corruption, an SVC dump is captured and a message is issued to the console to notify the user that a problem was encountered with the key-sequenced data set (KSDS). Data component CIs are checked for records out of sequence, records in the free space area, or any problems with RDF or CIDF fields.

This enhancement has been provided to resolve the problem where a VSAM KSDS data component might fail, which will cause the VSAM index to fail. If the index trap has failed, this might result in multiple data collection outages. Usually, the data is collected too late after the initial error and is usually not sufficient to diagnose what happened to the data component. Using the data trap, VSAM can capture a data failure for certain types of data component corruption.

The VSAM Data Trap default is OFF. The trap can be enabled or disabled by running a command. If performance is adversely impacted, you should disable the trap.

6.1.1 Enabling, disabling, and displaying the trap status

The default data trap setting is off. Here are the MVS commands that you can use to turn the data trap on or off or to verify its status:

```
V SMS,MONDS(IGWVSAM.BASE.DATA.TRAP),ON|OFF
D SMS,MONDS(IGWVSAM.BASE.VSAM.DEBUG.FEATURES)
```

6.1.2 VSAM KSDS data trap features

The VSAM KSDS data trap comes in two phases. The first phase includes the following features and limitations:

- The trap is only for KSDS.
- The trap will not test spanned records or when there can only be one record in a CI.
- The trap cannot test for sequence errors between control intervals.
- The trap tests the key sequence in the CI as it is being written to DASD.
- It might cause performance issues, especially during WRTBFR processing.
- The trap can be turned on and off.

In the second phase, the following features were added:

- VSAM Data Trap now can test for data key sequence errors between control intervals (CIs).
- VSAM Data Trap only processes those data set component names that are identified by users.

A new control block, PLHDTRAP, is obtained during OPEN processing in IDA0192Y for every placeholder (PLH).

The data trap code is in VSAM module IDA019RV and is called from VSAM module IDAM19R3, which is the physical I/O driver.

When IDAM19R3 is entered, there are several checks that are made, including:

- A data trap was activated by a user.
- It is not a catalog data set.
- There is a data component.
- There is a keyed data set.
- Data work areas are already built.
- There is a write request.
- There are no spanned records before a call to IDA019RV is made.

There are further checks in the trap code in IDA019RV. The trap test to see if the I/O is for a write of a data component of a KSDS. It is bypassed if it is not a KSDS or if it is a KSDS with spanned records. The trap checks the data CI in the buffer by analyzing the CI Definition Field (CIDF) values and checks to see if there are any records in the free space between the last record and the last RDF entry. The trap obtains the AMDSB from the current AMB and will determine the key length and the key offset in the records for the data set. These values are kept in the trap control block and will be used to compare the key sequence of the data records in the CI.

The trap locate the first Record Definition Field (RDF) entry and steps through the RDF entries to locate the individual data records. The RDF is either going to be a single entry for variable length records or it will be an RDF pair for equal length records. It is possible to have both types of RDF entries in a CI, so the RDF is analyzed for the type of entry. For equal length records, there is a count of the equal length records and the trap must keep track of the count while stepping through the data records.

When the trap encounters a key sequence error consisting of consecutive equal keys or the keys are not in ascending sequence, the IDAI1003E message is generated to the system console and a SVC dump is taken for problem analysis. The message is similar to the INDEX TRAP message IDAI1002E and identifies the failing data set.

When a corrupted data CI is encountered, the bad data buffer is not written to DASD. Information for the console message and the parameters for the SVC dump are contained in the trap control block. The storage address pointer to the trap control block is in the data AMB for the individual data sets.

6.1.3 Messages

A console message is issued when the VSAM KSDS data trap hits, and the message identifies the failing data set. The message is also saved in the job output.

These messages are not documented in the z/OS V1.11 message manuals, but will be included in the manuals for z/OS V1.12.

6.2 VSAM trace enhancement

VSAM record management tracing and VSAM internal tracing have been enhanced to improve reliability, availability, and serviceability (RAS) for VSAM. VSAM tracing usability is expanded by adding new hook points within various VSAM processing modules for record management. Serialization of the tracing routine has also been improved. Improvements have been made to the IPCS trace formatted output and some minor enhancement of the VSAM internal (VSAM footstep) tracing has been made to improve first failure data capture.

In the past, VSAM record management tracing has been enabled by using the VSAM AMP parameters in JCL. This situation meant that for data sets like catalogs that have no DD JCL statement associated with them a usermod had to be applied to a specific data set. This required that the program being loaded initially, which is not very practical.

6.2.1 New hook points

New hook points have been added to VSAM record management trace to enable tracing of CI reclaim and compression processing in VSAM. The new hook points, their module location, and point in processing are shown in Table 6-1.

Table 6-1 New hook points

Trace Point	Module	Description
20	IDAVCCMS	Before data record compression
21	IDAVCCMS	After data record compression
22	IDAVCCMS	Before data record decompression
23	IDAVCCMS	After data record decompression
26	IDA019SC	Start of a CI reclaim
27	IDA019SC	After completion of a CI reclaim

Use the HOOK subparameter of the TRACE subparameter of the JCL AMP DD parameter to select which hook points are to be used in a VSAM record management trace.

Attention: Users who have defined their own VSAM record management trace points must remove them for this support. See *z/OS DFSMSdfp Diagnosis,* GY27-7618 for details about removing a user define hook point.

6.2.2 New control blocks traced

This support provides the ability to trace the following control blocks under certain circumstances:

- The compression work area (CMWA) is traced when byte 1 bit 4 is in PARM1/PARM2 of the VSAM record management trace subparameter and hook points 20-23 are being traced. Example on page 213 shows the setting of the HOOK and PARM1 appropriate values to trace the CMWA. The necessary values are in bold font.

  Example 6-1 Tracing the CMWA and compression hook points

```
//KSDS01 DD DSN=VSAM.DATA.SET,DISP=SHR,
//     AMP=('TRACE=(PARM1=70100180112 2',
//     'HOOK=(20,21,22,23)')
```

- The RPL ECB is traced when RPLECB points to an ECB user.
- The ECB used in the IOMB is traced when the user specifies that the IOMB should be traced.

6.2.3 Invocation

Invocation of the VSAM record management trace requires that Generalized Trace Facility (GTF) to be active and that the required hook points are specified in the TRACE subparameter of the AMP JCL parameter. To accomplish this task, perform the following steps:

1. Code GTF parameters in a member. You can use the USR parm to trace all GTF user records with a TRACE statement, for example:

   ```
   TRACE=USR
   ```

   You could also use USRP to be more specific and stop the collection of any other USR events, for example:

   ```
   TRACE=USRP
   USR=FF5
   ```

2. Start GTF. Ensure that the GTF common storage buffer is at least 1024 KB so that it can handle the trace records. Use the BLOK parameter with the START GTF command:

   ```
   S GTF,,,(MODE=INT,BLOK=10M,SA=10M,SD=10M),MEMBER=xxxx,DSN=NULLFILE
   ```

   This command starts an internal GTF that can be formatted in an SVC dump in which up to 10 MB of trace records will be available. In this case, you need to specify both the SADMP(SA) and SDUMP (SD) parameters, and the SDUMP value cannot be larger than the specified SADMP value. Alternatively, for external output, use:

   ```
   S GTF,,,(BLOK=10M),MEMBER=xxxx
   ```

   Do not use the NOPROMPT parameter, as it causes the default of 40 K to be used for BLOK.

3. Add the TRACE subparameter to the AMP keyword in JCL to the DD statement for the VSAM KSDS to be traced, as shown in Example . This example traces all the new hook points in Table 6-1 on page 212, using a PARM1 that traces the Compression Management Work Area (CMWA) (in bit 3 of byte 1, X'10' is on) in addition to other control blocks.

`PARM1=701001801122` specifies that the following items are traced:

- ACB
- AMB
- AMBL
- CMWA
- PLH
- RPL
- User's key
- User's record
- Index and data control blocks

In addition, a validity check is made on all control blocks, and any that fail are traced.

Complete details about using VSAM record management tracing are described in *z/OS DFSMSdfp Diagnosis,* GY27-7618.

6.2.4 Interactive Problem Control System enhancements

Minor enhancements were made to the Interactive Problem Control System (IPCS) Trace Formatter to support the changes made in VSAM Record Management tracing.

Minor enhancements for the main trace header include:

- Changing the Hook field to display the 2 byte Hook ID
- Changing the Count Field to display 4 bytes instead of 2 bytes
- Adding a description for the newly added Hook points and Control blocks

These changes are shown in bold font in Figure 6-1.

```
**
       ****************************************       VSAM RECORD MANAGEMENT TRACE       ************************************
**
       **        HOOK=0017  IDAVCCMS             AFTER DATA RECORD IS DECOMPRESSED.
**
       **        COUNT=000007D3,CVERNON9,G       ,VSAMTEST,CVERNON.KHHO.KSDS.DATA
**
       **        REQUEST = GET     MOV,   ,SYN,KEQ,FKY,   ,FWD,ARD,NUP   REQUEST SUCCESSFUL
**
       ********************************************************************************************************************************
**

       ********* HOOK=0017, COUNT=000007D3 ******** ACB ************************************
       00015080 0000 A010004C 2529A088 00000000 96000001   00020000 00000000 80000008 00000000 *...<...h....o
.................*
       000150A0 0020 00000000 00000000 00540011 007B2EF0   D3000000 00000000 00000000 00000000 *.............OL
.............*
       000150C0 0040 00000000 00000000 00000000                                               *............
*

       ********* HOOK=0017, COUNT=000007D3 ******** DATA AMB ********************************
       2529A5B8 0000 400000F0 007B6C08 2529C950 2529C4D4   24E85040 2529CE40 10000008 2529A508 * ..0..%...I&..DM.Y& ...
......v.*
       2529A5D8 0020 C1D4C200 00000000 00000011 007C96D8   13027100 7FFCB080 2529CF58 7F641800 *AMB...........oQ....".......
```

Figure 6-1 IPCS formatting of VSAM record management trace entries for hook=22

The reason for increasing the Count field from 2 bytes to 4 bytes is to eliminate the confusion where the count number has suddenly changed from COUNT=FFFF to COUNT=0001 when there are more than 65,535 trace entries.

6.2.5 Migration actions

Users who have defined their own VSAM record management trace points must remove them. See *z/OS V1R11 Migration*, GA22-7499 for details about removing a user defined hook point.

6.3 VSAM SMB performance improvement

Performance of System Managed Buffering (SMB) Direct Optimized Access Bias had been affected adversely when a VSAM data set continued to grow. This is due to the original allocation for index buffer space becoming increasingly deficient as the data set size increases, This problem is avoided in the data buffer space by using the SMBVSP subparameter of the JCL AMP parameter, but for index buffer space, the only way to adjust the index buffer space to a more appropriate allocation was to close and reopen the data set. Changes have been made in z/OS V1R11 VSAM to avoid the necessity of closing and reopening the data set.

6.3.1 SMBVSP parameter processing changes

Before V1R11, when SMBVSP was used to specify the amount of storage for SMB Direct Optimized Access Bias, the value was used by VSAM OPEN to calculate the number of data buffers (BUFND). The number of index buffers (BUFNI) in contrast was calculated by VSAM OPEN based on the current high used CI, that is, it was based upon the data set size at open time.

With VSAM V1R11, VSAM OPEN calculates the BUFNI to be allocated using 20% of the value of SMBVSP, or the data set size, if the calculation using it actually yields a higher BUFNI. The usage in regard to calculating BUFND remains unchanged.

As KSDS grows, provisions can be made for a better storage allocation for both data and index buffers by using the SMBVSP parameter.

6.3.2 SMBVSP specification

SMBVSP is only a parameter for SMB Direct Optimized Access Bias. It can be specified in the SMS DATACLAS construct or through JCL, as shown in Example 6-2.

Example 6-2 SMSVSP specification in JCL

```
//STEP000 EXEC PGM=TVC15
//SYSPRINT DD SYSOUT=*
//KSDS0001 DD DSN=PROD.KSDS,DISP=SHR,AMP=('ACCBIAS=DO,SMBVSP=2048M')
```

For more details about SMB, see *z/OS V1R11 DFSMS Using Data Sets*, SC26-7410. For further details about invoking SMB, specifying DO (Direct Optimized), and SMBVSP values in a DATACLASS construct, see *z/OS V1R11 DFSMS Storage Administration Reference*, SC26-7402. For specification of JCLs, see *z/OS V1R11 JCL Reference*, SA22-7597.

6.4 Catalog detection of IMBED and REPLICATE

Support for the use of IMBED and REPLICATE parameters to define new catalogs and VSAM KSDSs was dropped in z/OS V1.5. While it is no longer possible to define a new catalog using IMBED or REPLICATE, the support for existing catalogs defined with these features still continues. For more information, refer to teh Software Announcement 204-180, 2004, which can be found at the following address:

http://www-01.ibm.com/common/ssi/cgi-bin/ssialias?subtype=ca&infotype=an&appname=iSource&supplier=897&letternum=ENUS204-180

Remove these features from any catalogs defined with them because they no longer provide a performance gain due to the storage cache features of modern DASD controllers. Using these features wastes DASD space and can, in fact, decrease performance, and even has caused outages in some cases. IBM announced in US Software Announcement 207-175, 2007 that there are plans to remove these attributes during DFSMSdss logical RESTORE and DFSMShsm RECALL operations.

In this release, the Catalog has implemented a health check to detect and identify catalogs that have these attributes.

6.4.1 IBM Health Checker for z/OS

The IBM Health Checker needs to be set up and ran to take advantage of the new check. You can use the F HZSPROC,DISPLAY command to check its status, as shown in Figure 6-2. This panel shows you the HZSPDATA data set in use and the HZSPRMxx member suffix in use.

```
F HZSPROC,DISPLAY
HZS0203I  13.54.09 HZS INFORMATION  585
POLICY(*NONE*)
OUTSTANDING EXCEPTIONS: 25
  (SEVERITY NONE: 0   LOW: 13   MEDIUM: 10   HIGH: 2)
ELIGIBLE CHECKS: 108  (CURRENTLY RUNNING: 0)
INELIGIBLE CHECKS: 25   DELETED CHECKS: 0
ASID: 0034   LOG STREAM: NOT DEFINED
  LOG STREAM WRITES PER HOUR: 114
  LOG STREAM AVERAGE BUFFER SIZE: 3586 BYTES
HZSPDATA DSN: WTSCPLX2.SC70.HZSPDATA
HZSPDATA RECORDS: 68
PARMLIB: 00
```

Figure 6-2 Display Health Checker status

For details about setting up the Health Checker, refer to Chapter 2, “Setting up IBM Health Checker for z/OS”, in *IBM Health Checker for z/OS User’s Guide,* SA22-7994.

After everything is set up, start the IBM Health Checker for z/OS using the START HZSPROC command or, if any parmlib members are being used, use the START HZSPROC,HZSPRMxx command.

After the IBM Health Checker for z/OS is up and running, the exit routine adds the check to the system. If the check is successfully added, it starts running at the interval specified in the exit routine or the HZSPRMxx parmlib member unless the system disables it.

6.4.2 HZSPRMxx parmlib member

IBM supplies an HZSPRM00 member, although it contains only comments that explain the statements and their syntax. The catalog check requires no input from a parmlib member. You only need to add statements if you want to change the default settings of the catalog check in some way.

6.4.3 New HZSPROC commands

By using the MVS MODIFY command (F for short), commands can be passed to the Health Checker address space.

F HZSPROC,DISPLAY,CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE),DETAIL is available and its output is needed to report a problem to IBM. The DISPLAY includes all of the values supplied in the check definition and modified by the installation. It also includes the reason the check was written and the severity of the check. Figure 6-3 shows information about the check, but nothing about the detected catalogs.

```
F HZSPROC,DISPLAY,CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE),DETAIL
HZS0201I 12.29.33 CHECK DETAIL     436
CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE)
 STATE: ACTIVE(ENABLED)              STATUS: EXCEPTION-LOW
 EXITRTN: IGGOCLHX
 LAST RAN: 08/23/2009 16:28    NEXT SCHEDULED: 08/24/2009 16:28
 INTERVAL: 24:00
 EXCEPTION INTERVAL: SYSTEM
 SEVERITY: LOW
 WTOTYPE: INFORMATIONAL
 SYSTEM DESCCODE: 12
 THERE ARE NO PARAMETERS FOR THIS CHECK
 REASON FOR CHECK:  IMBED and REPLICATE attributes are no longer
                    allowed on the define of a new catalog
 MODIFIED BY: N/A
 DEFAULT DATE: 20080619
 ORIGIN: HZSADDCK
 LOCALE: HZSPROC
 DEBUG MODE: OFF  VERBOSE MODE: NO
```

Figure 6-3 Health Checker DISPLAY CHECK output

6.4.4 Health Checker output

The system programmer should be prepared to evaluate the output contained in the different messages issued.

In the case of IGGHC104E, a decision should be made when to perform an EXPORT/IMPORT of the affected catalogs to remove the IMBED and REPLICATE attributes.

If an error is detected at any time during the processing of CSI, the HZS1002E message is issued. If a decision is made to not continue with the check, the HZS1002I V is issued along with IGGHC107I. Diagnostic information is returned in HZS1002E and can help diagnose the problem. If a decision is made to continue with the check after the error, IGGHC108I is issued and gives a brief description of the error.

HZS1001E and HZS1217E are issued when the check detects a bad parameter. This will stop the check execution.

If the check needs to be deactivated, the system programmer may create a policy statement in the HZSPRMxx parmlib member. For details, see Chapter 4, “Managing Checks”, in *IBM Health Checker for z/OS User’s Guide*, SA22-7994.

6.4.5 Messages and displays

The IGGHC104E exception message and IGGHC103I, IGGHC107I, IGGHC108I, and IGGHC109I informational messages are new messages. When an exception is found, the IGGHC105R and IGGHC106I messages are used to generate a report of affected catalogs in the message buffer in addition to IGGHC104E message. For an example of the IGGHC104E message, see Figure 6-4.

```
HZS0001I CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE): 838
IGGHC104E The CATALOG_IMBED_REPLICATE check has detected one or more
catalogs defined with the IMBED and/or REPLICATE attributes.
```

Figure 6-4 IGGHC104E message

When the IGGHC104E message is issued to the console, as shown in Figure 6-4, the system programmer or storage administrator needs to enquire which catalogs have been found with either the IMBED or REPLICATE attributes. This task can be accomplished by running an HZSPRNT job, as shown in Figure 6-5.

```
//IDCAMS  EXEC PGM=HZSPRNT,
//*            PARM='CHECK(*,*)'      list all checks
//             PARM='CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE)'
//SYSOUT   DD  SYSOUT=*,DCB=LRECL=256
```

Figure 6-5 HZSPRNT JCL to list catalogs in the IGGHC106I message

By running the HZPRNT job, a report of the check details contained in the Health Checker message buffer is produced, as shown in Figure 6-6 on page 219. This report can be used by the system programmer or storage administrator to create EXPORT/IMPORT jobs to remove the attributes from the identified catalogs.

```
********************************************************************************
*                                                                              *
* HZSPRINT (HBB7750-07288)  2009/08/24 14:31                                   *
*                                                                              *
* HZSU001I Check messages                                                      *
* Sysplex: SANDBOX    System: SC70                                             *
*                                                                              *
* Filter: CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE)                            *
*                                                                              *
********************************************************************************

********************************************************************************
*                                                                              *
* Start: CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE)                             *
*                                                                              *
********************************************************************************

CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE)
START TIME: 08/23/2009 16:28:57.498736
CHECK DATE: 20080619  CHECK SEVERITY: LOW

IGGHC106I
Following catalog(s) were inspected by the health check

-------------------------------------------------------------------------
 CATALOG NAME                                ATTRIBUTE  COMP AVAILABILITY
-------------------------------------------------------------------------
CATALOG.DB2ICF2.VTOTCAT                       IMBED     DATA  YES
CATALOG.DB2ICF2.VTOTCAT                       IMBED     INDEX YES
CATALOG.DB2ICF2.VTOTCAT                       REPLICATE INDEX YES
CATALOG.HCD.USERCAT                           IMBED     DATA  YES
CATALOG.HCD.USERCAT                           IMBED     INDEX YES
...
MCAT.OS3R4V01.VTOTCAT                         IMBED     DATA  YES
MCAT.OS3R4V01.VTOTCAT                         IMBED     INDEX YES
MCAT.OS3R4V01.VTOTCAT                         REPLICATE INDEX YES
MCAT.SANDBOX.VSBOX11                          IMBED     DATA  YES
MCAT.SANDBOX.VSBOX11                          IMBED     INDEX YES
MCAT.SANDBOX.VSBOX11                          REPLICATE INDEX YES
UCAT.DB2V01                                   --------- ----- NO
UCAT.MQWF                                     IMBED     DATA  YES
UCAT.MQWF                                     IMBED     INDEX YES
UCAT.MQWF                                     REPLICATE INDEX YES

IGGHC109I IMBED/REPLICATE HEALTH CHECK SUMMARY REPORT:
A total of 88 catalogs were inspected by the Health Check
64 catalogs did not have either IMBED or REPLICATE attributes on them.
4 catalogs were not processed by the Health Check either because the
catalog volume was not available or the catalog entry was in error.
20 catalogs were detected with either IMBED and/or REPLICATE attributes
on them.
```

Figure 6-6 HZSPRNT output for CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE)

An alternative to running an HZSPRNT job is to use the Health Checker panels set up in SDSF, assuming they have been set up already. The SDSF CK option, or CK E for just exceptions, shows the checks, which can then be selected using an S, as shown in Figure 6-7, to get the same information in the HZSPRINT report shown in Figure 6-6 on page 219.

```
 Display  Filter  View  Print  Options  Help
-------------------------------------------------------------------------------
SDSF HEALTH CHECKER DISPLAY  SC70                      LINE 1-16 (133)
COMMAND INPUT ===>                                        SCROLL ===> CSR
NP   NAME                              CheckOwner       State             Status
     ASM_LOCAL_SLOT_USAGE              IBMASM           ACTIVE(ENABLED)   SUCCES
     ASM_NUMBER_LOCAL_DATASETS         IBMASM           ACTIVE(ENABLED)   SUCCES
     ASM_PAGE_ADD                      IBMASM           ACTIVE(ENABLED)   SUCCES
     ASM_PLPA_COMMON_SIZE              IBMASM           ACTIVE(ENABLED)   EXCEPT
     ASM_PLPA_COMMON_USAGE             IBMASM           ACTIVE(ENABLED)   SUCCES
S    CATALOG_IMBED_REPLICATE           IBMCATALOG       ACTIVE(ENABLED)   EXCEPT
     CEE_USING_LE_PARMLIB              IBMCEE           ACTIVE(ENABLED)   EXCEPT
     CNZ_AMRF_EVENTUAL_ACTION_MSGS     IBMCNZ           ACTIVE(ENABLED)   SUCCES
     CNZ_CONSOLE_MASTERAUTH_CMDSYS     IBMCNZ           ACTIVE(ENABLED)   SUCCES
     CNZ_CONSOLE_MSCOPE_AND_ROUTCODE   IBMCNZ           ACTIVE(ENABLED)   EXCEPT
     CNZ_CONSOLE_ROUTCODE_11           IBMCNZ           ACTIVE(ENABLED)   EXCEPT
     CNZ_EMCS_HARDCOPY_MSCOPE          IBMCNZ           ACTIVE(ENABLED)   SUCCES
     CNZ_EMCS_INACTIVE_CONSOLES        IBMCNZ           ACTIVE(DISABLED)  GLOBAL
     CNZ_OBSOLETE_MSGFLD_AUTOMATION    IBMCNZ           ACTIVE(ENABLED)   EXCEPT
     CNZ_SYSCONS_MSCOPE                IBMCNZ           ACTIVE(ENABLED)   SUCCES
     CNZ_SYSCONS_PD_MODE               IBMCNZ           ACTIVE(ENABLED)   SUCCES
 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=IFIND     F6=BOOK
 F7=UP        F8=DOWN      F9=SWAP nex F10=LEFT     F11=RIGHT    F12=RETRIEVE
```

Figure 6-7 SDSF CK: Health Checker panel

6.4.6 Other useful commands

The UPDATE command can be used to stop Health Checker from checking catalogs. The MVS command syntax is;

```
F HZSPROC,UPDATE,CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE),ACTIVE|INACTIVE
```

It is used to both stop and start the check, as shown in Figure 6-8.

```
F HZSPROC,UPDATE,CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE),INACTIVE
HZS0400I CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE): 454
UPDATE PROCESSING HAS BEEN COMPLETED
...
F HZSPROC,UPDATE,CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE),ACTIVE
HZS0400I CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE): 456
UPDATE PROCESSING HAS BEEN COMPLETED
```

Figure 6-8 Using UPDATE to stop and start the catalog check

The UPDATE command's full syntax is:

```
UPDATE
CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE)
```

```
ACTIVE | INACTIVE
SEVERITY(LOW)
INTERVAL(TIMER) HOURS(24)
REASON('IMBED and REPLICATE are no longer'
'allowed on the define of catalogs')
```

It can be used in the MVS MODIFY command, as shown in Figure 6-8 on page 220 or used in an HZSPRMxx parmlib member and then implemented using the F HZSPROC,REPACE PARMLIB=xx command, as shown in Figure 6-9.

```
F HZSPROC,REPLACE PARMLIB=(00)
IEE252I MEMBER HZSPRM00 FOUND IN CPAC.ZOSR1B.PARMLIB
HZS0403I REPLACE PARMLIB PROCESSING HAS BEEN COMPLETED
...
F HZSPROC,REPLACE PARMLIB=(CV)
IEE252I MEMBER HZSPRMCV FOUND IN SYS1.PARMLIB
HZS0400I CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE): 634
UPDATE PROCESSING HAS BEEN COMPLETED
HZS0403I REPLACE PARMLIB PROCESSING HAS BEEN COMPLETED
```

Figure 6-9 Changing the HZSPRMxx member in use

The REFESH command is a quick and convenient way to reset the catalog check back to its original defaults:

```
F HZSPROC,REFRESH,CHECK(IBMCATALOG,CATALOG_IMBED_REPLICATE)
```

7

SMS enhancements

In this chapter, we discuss the enhancements made to SMS in DFSMS V1.11. These enhancements include:

- Data set separation by volume
- SMS striping volume selection
- SMS health check

7.1 Data set separation by volume

Currently, when creating new data sets, SMS ranks all the candidate volumes based on some attributes and selects the volumes with the highest preference attributes. This algorithm normally works well if there are sufficient candidate volumes in the highest ranking pool. However, if the number of candidate volumes in the highest ranking pool is small and static, then data sets might be allocated on these volumes repeatedly.

Some applications, such as DB2, might experience performance degradation if its critical data sets (DB2 partitions) are on the same volume.

The current way to place SMS managed data sets on different volumes is to assign those data sets a Storage Class with a guaranteed space attribute and manually specify the VOLSER for each data set involved.

Pre-z/OS V1.11, users are able to separate data sets at the physical control unit (PCU) level to prevent single point of failure. In V1.11, this feature is expanded to provide a facility for the users to separate their critical data sets onto different extent pools and volumes for better performance.

During the allocation, SMS first tries to separate these user-specified data sets onto different extent pools. If separation by extent pools is not possible, SMS attempts to separate them onto different volumes.

7.1.1 Extent pool

An extent pool consists of one or several ranks. Two extent pools cannot be configured out of the same rank. Thus, I/O contentions are avoided when these critical data sets reside on different extent pools. Logical volumes are created from extents in an extent pool. I/O contentions will likely be reduced when data sets are allocated on different logical volumes than on the same volume.

7.1.2 IBM System Storage DS8000 storage architecture

In order to understand more about this feature, users should know the basic concepts of the IBM System Storage DS8000 storage architecture, as shown in Figure 7-1 on page 225:

- Disk Drive Module (DDM): A physical storage unit that consists of a single disk drive.
- Array site: A group of eight DDMs that are selected by the IBM System Storage DS8000 server algorithm in a storage facility image. An array site is managed by one storage facility image.
- Array: Each array site can be individually formatted by the user to use a specific RAID format. A formatted array site is called an array. The supported RAID formats are RAID 5 and RAID 10. The process of selecting the RAID format for an array is also called defining an array.
- Rank: A rank is defined by the user. The user selects an array and defines the storage format for the rank, which is either Count Key Data (CKD) or Fixed Block (FB) data. One rank will be assigned to one extent pool by the user.

- Extents: The available space on each rank is divided into extents. The extents are the building blocks of the logical volumes. The characteristic of an extent is its size, which depends on the specified device type when defining a rank:
  - For fixed block format, the extent size is 1 GB.
  - For CKD format, the extent size is 94 GB for model 1.
- Extent pools: An extent pool refers to a logical construct that is used to manage a set of extents. The user defines extent pools by selecting one to N ranks managed by one storage facility image. The user defines which storage facility image server (Server 0 or Server 1) manages the extent pool. All extents in an extent pool must be of the same storage type (CKD or FB). Extents in an extent pool can come from ranks defined with arrays of different RAID formats, but the same RAID configuration within an extent pool is recommended. The minimum number of extent pools in a storage facility image is two (each storage facility image server manages a minimum of one extent pool).

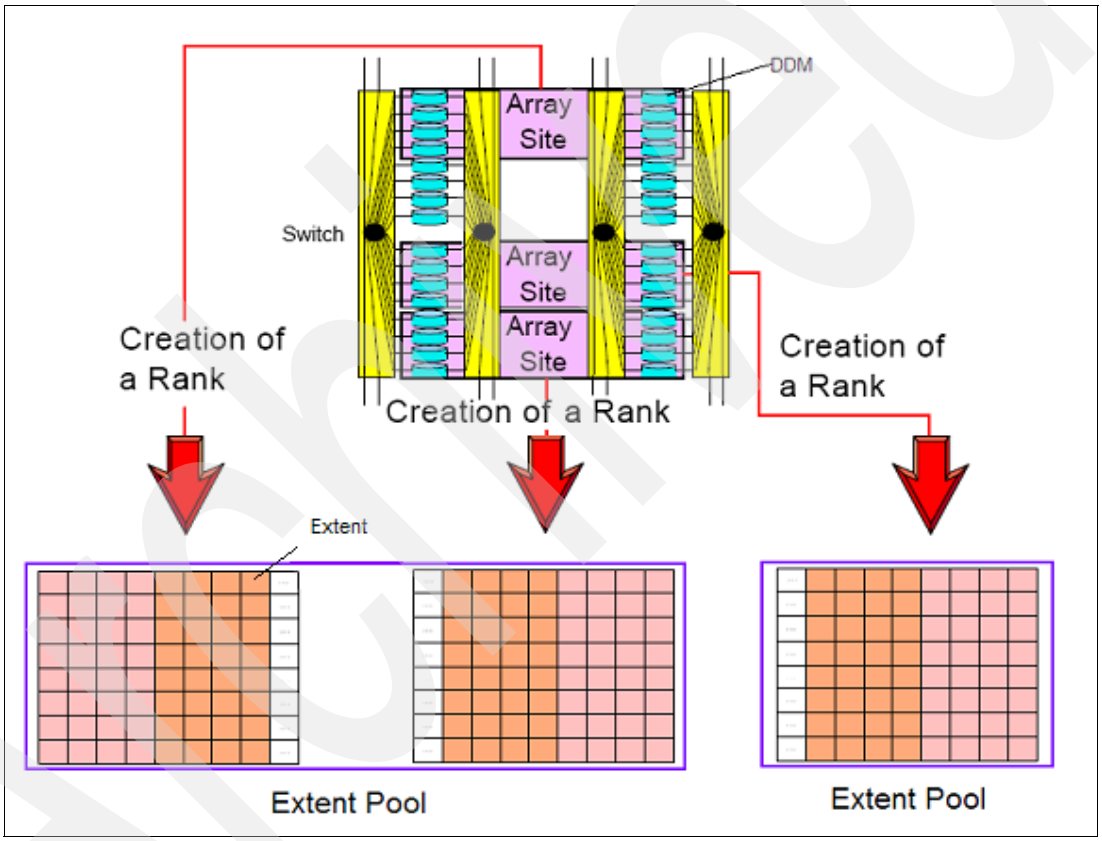


Figure 7-1 Relationship between array, rank, extent, and extent pool in the IBM System Storage DS8000 storage architecture

7.1.3 Data set separation profile

To use data set separation function, users must create a data set separation profile and specify the name of the profile to the CDS base configuration, as shown in Figure 7-2.

The profile data set has three requirements:

- The organization can be either sequential or partitioned.
- The record format must be fixed or fixed-block (RECFM=FB).
- It must be cataloged. Catalog the profile data set in the MASTERCAT to avoid a deadlock between SMS and CATALOG during the initial load of the program with contention in the SMS ASID, or a SYSZTIOT might occur.

SMS reads the data set separation profile when SMS initializes with an active configuration, when SMS restarts, and when a configuration is activated or switched.

All systems within the SMS complex share the same data set separation profile that is specified in the base configuration.

```
 Panel  Utilities  Scroll  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              SCDS BASE ALTER                 Page 1 of 2
 Command ===>

 SCDS Name . : SYS1.SMS.SCDS
 SCDS Status : VALID

 To ALTER SCDS Base, Specify:

  Description ===> BASE SMS CONFIG FOR OE
              ===>

  Default Management Class  . .          (1 to 8 characters)
  Default Unit  . . . . . . . .          (esoteric or generic device name)
  Default Device Geometry
    Bytes/Track . . . . . . . . 56664    (1-999999)
    Tracks/Cylinder . . . . . . 15       (1-999999)
  DS Separation Profile                  (Data Set Name)
    ==> 'MHLRES4.DATASET.SEP.PROFILE'


 Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
 Use HELP Command for Help; Use END Command to Save and Exit; CANCEL To Exit.
```

Figure 7-2 Define/Alter base configuration in include data set separation profile

A data set separation profile contains at least one data set separation group. Each data set separation group specifies whether separation is at the PCU or volume level, whether it is required or preferred, and includes a list of data set names to be separated from each other during allocation. The data set names listed in the separation groups must follow the naming convention described in *z/OS MVS JCL Reference*, SA22-7597. The wild card characters are supported starting at the third level qualifier (for example, A.B.*).

7.1.4 Data set separation profile pre-z/OS V1.11 command syntax

In previous releases of z/OS, the data set separation function is at the physical control unit level (PCU). Each data set separation group contains a FAILLEVEL keyword and a list of data set names that are separated from each other during allocation. The fail level keyword specifies whether the separation is mandatory or preferred, as shown in Figure 7-3.

```
SEPARATIONGROUP | SEP -
FAILLEVEL | FAIL ({PCU | NONE}) -
DSNLIST | DSNS | DSN (data-set-name[,data-set-name,...]);
```

Figure 7-3 Pre-z/OS V1.11 data set separation command syntax

Where:

- FAILLEVEL(PCU) indicates that separation on the PCU level is required. SMS fails the allocation if the requested data set cannot be separated on the PCU level from other data sets that are listed in the SEPARATIONGROUP.
- FAILLEVEL(NONE) indicates that separation on the PCU level is preferred. SMS allows the allocation if the requested data set cannot be separated on the PCU level from other data sets that are listed in the SEPARATIONGROUP. SMS allocates the data sets on the same PCU.

When this syntax is used, data set names listed in a data set separation group must not contain quotation marks or wildcards, and must follow the naming convention described in *z/OS MVS JCL Reference*, SA22-7597.

New data set separation profile command syntax

In z/OS V1.11, the syntax of the data set separation profile has been changed to support both PCU and VOLUME levels. The existing syntax that was provided for the data set separation by PCU will continue to function. A toleration fix is provided to lower releases to recognize the new syntax for PCU separation and ignore the new syntax for VOLUME separation. The new PCU syntax is supported on lower level releases with the exception of wild card characters.

The new command syntax is shown in Figure 7-4.

```
SEPARATIONGROUP | SEP (PCU | {VOLUME | VOL}) -
TYPE ({REQUIRED | REQ | R} | {PREFERRED | PREF | P}) -
DSNLIST | DSNS | DSN (data-set-name[,data-set-name,...]);
```

Figure 7-4 New data set separation command syntax

Where:

- SEPARATIONGROUP(VOLUME | VOL) identifies the separation level: PCU or VOLUME.
- TYPE(REQ | PREF) identifies the separation type: REQUIRED or PREFERRED.
- Wildcards are supported on DSN at the third qualifier (for example, A.B.* but not A.*).

Example 7-1 shows a data set separation profile that contains four data set separation groups.

Example 7-1 Data set separation groups example

```
SEP -
FAIL(NONE) -
```

```
DSNLIST(SMS.PROD.SCDS1, SMS.PROD.SCDS2);

SEP(PCU) -
TYPE(REQ) -
DSNLIST(SMS.PROD.ACDS, SMS.PROD.COMMDS);

SEP(VOL) -
TYPE(PREF) -
DSNLIST(A.B.C1, A.B.C2, A.B.C3);

SEP(VOL) -
TYPE(REQUIRED) -
DSNLIST(A.B.*);
```

7.1.5 Volume selection with data set separation by volume

SMS invokes data set separation by volume function when:

- Allocating a new SMS-managed data set or extending an existing SMS-managed data set to a new volume.
- The data set name that is being allocated matches one of the data set names specified in the DSNLIST of a VOLUME separation group.

SMS builds a candidate volume list that contains all the candidate volumes that are eligible for selection. If the current allocation is a candidate for separation, SMS calls the Catalog Search Interface (CSI) to derive a list of volumes occupied by data set names that are specified in the matched separation groups and contained in ICF catalogs.

After building the candidate volume list and the separation volume list, SMS determines whether a given candidate volume on the candidate volume list is in the same extent pool as any volume on the separation volume list. The same extent pool flag is set on the given candidate volume when it is in the same extent pool as at least one of the volumes in the separation volume list. The same extent pool flag is not set if a given candidate volume is on the same extent pool ID as volumes in the separation volume list but on a different PCU.

Each candidate volume is assigned a volume preference weight based on the attributes requested by the user via the JCL, Storage Class. Two new volume preference attributes, Separation By Extent Pool and Separation By Volume, are defined for each candidate volume. The volumes that are in different extent pools than volumes on the separation list meet the attribute of separation by extent pool and will be most preferred. The volumes that are not on the separation volume list but are in the same extent pools as any volume on the separation volume list meet the attribute of separation by volume. When TYPE(REQ) is specified, candidate volumes that are also on the separation volume list are rejected. When TYPE(PREF) is specified, candidate volumes that are also on the separation volume list will be ranked lower and less preferred for selection.

Here are the existing attributes in the descending order of importance and where the two new attributes fit into this hierarchy:

1. Separation by PCU: The volume does not reside in the same physical control unit that has allocated a data set from which this data set should be separated, as specified in the data set separation profile.
2. Separation by Extent Pool: The volume does not reside in the same extent pool that has allocated a data set from which this data set should be separated, as specified in the data set separation profile.

3. Separation by Volume: The volume does not contain a data set from which this data set should be separated, as specified in the data set separation profile.
4. Volume count: The volume resides in a Storage Group that has enough eligible volumes to satisfy the requested VOLUME COUNT.
5. Primary Threshold: The volume has sufficient space in the target addressing space for the allocation amount without exceeding the Storage Group HIGH THRESHOLD value.
6. Secondary Threshold: The volume has sufficient space for the allocation amount without exceeding the Storage Group HIGH THRESHOLD value.
7. Enabled status: The volume and its associated Storage Group SMS status are ENABLED.
8. Multi-tiered Storage Group: The volume resides in a Storage Group that will be selected in the order of specification.
9. End of Volume Extend: This is an end-of-volume extend, where an extend Storage Group is specified, and the volume does not reside in the specified extend Storage Group.
10. Non-Overflow Storage Group: The volume resides in a non-overflow Storage Group.
11. IART: The volume is mountable, and an IART value greater than zero was specified in the Storage Class.
12. Fast Replication: The volume is eligible for a fast replication request.
13. EAV: The volume has extended attributes (Format 8 and Format 9 DSCBs) or optionally resides in extended addressing space.
14. Accessibility: The volume resides in a control unit that supports ACCESSIBILITY, and the Storage Class ACCESSIBILITY value is PREFERRED, or the volume resides in a control unit that does not support ACCESSIBILITY, and the Storage Class ACCESSIBILITY value is STANDARD.
15. PAV: The volume supports the PAV specification in the Storage Class
16. Availability: The volume resides in a control unit that supports AVAILABILITY, and the Storage Class AVAILABILITY value is PREFERRED, or the volume resides in a control unit that does not support AVAILABILITY, and the Storage Class AVAILABILITY value is STANDARD.
17. Extended Format: The volume resides in a control unit that supports EXTENDED FORMAT and the Data Class IF EXT value is PREFERRED.
18. Millisecond Response Time: The volume provides the requested response time that is specified or defaulted to in the Storage Class DIRECT MSR or SEQUENTIAL MSR, or the volume provides a faster response time than what is specified or defaulted to in the Storage Class DIRECT MSR or SEQUENTIAL MSR.

Because the data set separation attributes are the most important attributes, volumes that meet this criteria will be ranked higher than volumes that do not, regardless of whether other preference attributes (thresholds, enabled, and so on) are met or not.

The separation by volume function is supported for all types of allocations, including striping allocations and allocations that use best-fit. SMS fails the allocation if the separation level is REQ but separation at the volume level cannot be achieved. SMS issues an IGD17372I message to the job log and the hardcopy log if the separation level is PREF but separation at the volume level cannot be achieved. IGD17372I is enhanced to indicate the separation levels (PCU, VOL, or both) that are specified for the data set.

During normal allocations, candidate volumes that satisfy all requested attributes are considered primary volumes. Primary volumes must meet the separation by extent pool attribute. SMS calls SRM to select the best volume from the pool of primary volumes. Volumes that do not satisfy all attributes are considered secondary volumes. All secondary volumes are ranked based on the relative importance of attributes that they satisfy. All volumes that have the same rank are considered to occupy one band. If no primary volumes are available, then SMS randomly selects volumes from the pool of secondary volumes in the highest band.

For striping allocations, volumes that meet the separation attributes will be ranked higher than the volumes that do not. SMS selects the best Storage Group and the volumes that have the highest preference weight from each unique controller to meet the target stripe count.

For best-fit allocations, SMS takes the separation attributes into consideration when ranking volumes. SMS uses the existing algorithm to select the best Storage Group and the volumes that have the highest preference weight.

7.1.6 Using data set separation with generation data groups and striping

Data set striping can use multiple controllers when multistriped. When using data set separation on the PCU level that references undistributed data sets, ensure that the Storage Group ACS routines select Storage Groups that contain a sufficient number of PCUs.

Generation data groups (GDGs) that are listed in the data set separation profile must specify the absolute generation and version number (for example, A.B.C.G0001V00). Data set separation does not support relative generation numbers and GDS base names.

When specifying separation of GDGs on the PCU level, ensure that the number of generations that reside on pool volumes does not exceed the number of available PCUs.

7.1.7 Environmental conditions affecting data set separation

Under the following environmental conditions, SMS might not separate two data sets that are specified for required or preferred separation:

- Allocation is not SMS-managed.
- Allocation is performed on a system in the SMS complex that runs a prior version of SMS that does not support data set separation.
- During switching or activation of an SMS configuration, the data set separation profile cannot be accessed or fails validation.
- A service that is allocating a temporary data set name does not provide the real data set name to SMS.
- Two data sets that are specified to be separated are allocated at the same time by two different tasks or two different systems. For performance reasons, SMS does not serialize data set allocations.
- A volume is varied online during allocation.
- An IODF change occurs during allocation.
- A data set name that is not listed in the data set separation profile is specified during DFSMShsm recover.
- A data set separation profile might have been modified after the configuration was activated. This can result in differences between the data set separation profile data set and the active copy of the data set separation profile.

- SMS does not perform data set separation during RENAME.
- SMS does not perform data set separation during DFSMShsm migration to level 1 or level 2 storage.
- SMS does perform data set separation during DFSMShsm recall.
- SMS does not perform data set separation during full volume image copy.

SMS might not issue the corresponding warning message IGD17372I in those cases where allocation was successful and data set separation was not honored.

7.1.8 Data set separation example

A data set separation rule is defined in Example 7-2

Example 7-2 Data set separation example

```
SEP(VOL) -
TYPE(REQ) -
DSNLIST(MHLRES4.SEP1.*,MHLRES4.SEP2.*);
```

It is defined in the MHLRES4.DATASET.SEP.PROFILE data set profile, which is in the CDS base shown in Example 7-3.

Example 7-3 Base CDS configuration panel

```
Panel  Utilities  Scroll  Help
 sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                              SCDS BASE ALTER                 Page 1 of 2
 Command ===>

 SCDS Name . : SYS1.SMS.SCDS
 SCDS Status : VALID

 To ALTER SCDS Base, Specify:

  Description ===> BASE SMS CONFIG FOR OE
              ===>

  Default Management Class  . .          (1 to 8 characters)
  Default Unit  . . . . . . . .          (esoteric or generic device name)
  Default Device Geometry
    Bytes/Track . . . . . . . . 56664    (1-999999)
    Tracks/Cylinder . . . . . . 15       (1-999999)
  DS Separation Profile                  (Data Set Name)
    ==> 'MHLRES4.DATASET.SEP.PROFILE'


 Use ENTER to Perform Verification; Use DOWN Command to View next Panel;
 Use HELP Command for Help; Use END Command to Save and Exit; CANCEL To Exit.
```

We then activate the new SMS configuration and use IEFBR14 to allocate four data sets, as shown in Example 7-4.

Example 7-4 Example of data set allocation

```
//MHLRES4S JOB (999,POK),MSGLEVEL=1,NOTIFY=MHLRES4
/*JOBPARM S=SC70
//*
```

```
//STEP010   EXEC PGM=IEFBR14
//TEST1     DD DISP=(,CATLG),DSN=MHLRES4.SEP1.VOL1,
//        SPACE=(TRK,(1,1)),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//TEST2     DD DISP=(,CATLG),DSN=MHLRES4.SEP2.VOL2,
//        SPACE=(TRK,(1,1)),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//TEST3     DD DISP=(,CATLG),DSN=MHLRES4.SEP1.TES1,
//        SPACE=(TRK,(1,1)),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
//TEST4     DD DISP=(,CATLG),DSN=MHLRES4.SEP2.TES2,
//        SPACE=(TRK,(1,1)),
//        DCB=(RECFM=FB,LRECL=80,BLKSIZE=0)
```

The result shows that four data sets were allocated on different volumes, as shown in Example 7-5.

Example 7-5 Data set separation by volume example

```
Menu  Options  View  Utilities  Compilers  Help
 ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
 DSLIST - Data Sets Matching MHLRES4.SEP*                        Row 1 of 4
 Command ===>                                                 Scroll ===> CSR

 Command - Enter "/" to select action                Message          Volume
 ------------------------------------------------------------------------------
          MHLRES4.SEP1.TES1                                               MLD10C
          MHLRES4.SEP1.VOL1                                               MHLS1E
          MHLRES4.SEP2.TES2                                               MHLS1A
          MHLRES4.SEP2.VOL2                                               MLD20C
 ****************************** End of Data Set list ****************************
```

7.1.9 New system message

The following message has been added in z/OS V1.11.

IGD089I

This message is generated at the time a new SMS source control data set (SCDS) has been activated or an active control data set (ACDS) has been switched. The control data set contains an unsupported data set separation group. Figure 7-5 shows the format of the new IGD089I message.

```
IGD089I DATA SET SEPARATION PROFILE dsn CONTAINING AN UNSUPPORTED DATA SET
SEPARATION GROUP WAS FOUND ON LINE nnn
```

Figure 7-5 The new IGD089I message

7.1.10 Updated system messages

The following messages have been modified in z/OS V1.11.

IGD17372I

This message has been update to indicate DATA SET SEPARATION BY (sep_level), as shown in Figure 7-6.

```
IGD17372I JOBNAME jobname PROGRAM NAME progname STEPNAME stepname DDNAME ddn.
DATA SET SEPARATION BY (sep_level) COULD NOT BE PROVIDED FOR DATA SET dsn. DATA
SET SEPARATION PROFILE profile.
```

Figure 7-6 IGD17372I

The sep_level separation level is PCU or VOL, or both.

IGD17389I

IGD17389I has been changed to add two new flags indicating whether a candidate volume meets the separation by extent pool and by volume.

```
IGD17389I volser ({S | N | R},preference,fail_flags,diagdata)...
```

Figure 7-7 IGD17389I

The following are the new message flags:

1 Meets the PCU separation criteria, if specified.

.1 Meets the extent pool separation criteria, if specified.

..1 Meets the VOLUME separation criteria, if specified.

7.1.11 Migration considerations

If you have the old syntax in your data set separation profile, consider updating the profile to use the current syntax. Toleration PTFs are available that provide support for the z/OS V1.11 syntax for PCU-level separation on lower level systems, with the exception of wildcard characters in data set names. With the PTFs, separation groups defined with syntax that is not supported, such as SEP(VOL) or wildcard characters in the data set name list, are ignored.

Table 7-1 shows example definitions that are equivalent to the old and new syntax.

Table 7-1 PCU separation level syntax example

Old syntax	New syntax
SEP - FAIL(PCU) - DSNLIST(A.B.C1, A.B.C2);	SEP(PCU) - TYPE(REQ) - DSNLIST(A.B.C1, A.B.C2);

7.1.12 Coexistence considerations

Consider the following coexistence items:

- The old syntax of PCU separation continues to function in z/OS V1.11.
- APAR OA25344 must be installed to lower level systems. It will:
  - Ignore the VOLUME separation group.
  - Accept both the old and new syntax for PCU separation.
  - Ignore wildcards characters in the PCU separation syntax.
- DFSMS issues the existing IGD06032I message to users when the new syntax is ignored during the validation of the SMS configuration
- DFSMS issues the new IGD089I message to the console when the new syntax is ignored during the activation of the SMS configuration.
- The data set separation function is deactivated if the new syntax is detected in z/OS V1.8 and earlier.

7.2 SMS striping volume selection

The existing implementation of the SMS striping volume selection has several restrictions that do not exist in conventional volume selection for non-striped allocations, such as:

- Volumes that are above the high allocation threshold are rejected outright.
- Enabled and quiesced volumes are treated equally.
- Normal and overflow Storage Groups are treated equally.
- The multi-tiered Storage Group function is not supported.
- The data set separation function is not supported.
- Volume preference attributes, such as availability, accessibility, PAV capability, and so on, that are used in conventional volume selection are ignored.

In z/OS V1.11, the SMS striping volume selection enhancement tries to make striping allocation function for both VSAM and non-VSAM as close as possible to the conventional volume selection.

This enhancement removes above unnecessary restrictions and enhances striping volume selection to support all volume preference attributes similar to non-striping volume selection and:

- Allows volumes that are above high allocation threshold to be eligible for selection as secondary volumes
- Prefers an enabled volume over a quiesced volume
- Prefers a normal Storage Group over an overflow Storage Group
- Supports a multi-tiered Storage Group function to honor a Storage Group sequence
- Prefers volumes that meet data set separation criteria
- Supports availability, accessibility, PAV, EAS, and other volume preference attributes that are used in non-striping volume selection
- Automatically activates Fast Volume Selection to improve performance when 100 volumes have been rejected by DADSM for insufficient space

7.2.1 New striping volume selection algorithm

Striping volume selection is invoked only in the following situations:

- During initial allocation: If the Data Class specifies an extended format as either REQUIRED or PREFERRED and the SDR value in the Storage Class is nonzero. If the SDR value is zero, a non striped data set in extended format is allocated, and conventional volume selection is used.
- During restore/recall processing: If the data set was a multi stripe data set when it was migrated or backed up. If the data set was single-striped when migrated or backed up, it follows the conventional volume selection path.

Striping volume selection is very similar to conventional volume selection. Volumes that are eligible for selection are classified as primary and secondary, and assigned a volume preference weight, based on preference attributes:

- Primary: For each controller, SMS randomly assigns a single volume that meets all of the requested preference attributes as the primary volume.
- Secondary: For each controller, SMS classifies all eligible volumes other than the primary volume as secondary volumes.

Preference weight

SMS calculates the average preference weight of each Storage Group using the preference weights of the volumes that will be selected if the Storage Group is selected for allocation. Then, SMS selects the Storage Group that contains at least as many primary volumes as the stripe count and has the highest average weight. If there are no Storage Groups that meet these criteria, the Storage Group with the largest number of primary volumes is selected. If multiple Storage Groups have the largest number of primary volumes, the one with the highest average weight is selected. If there are still multiple Storage Groups that meet the selection criteria, SMS selects one at random.

After selecting a Storage Group, SMS selects volumes by their preference weight. Primary volumes are preferred over secondary volumes, as they have a higher preference weight. Secondary volumes are selected when there are an insufficient number of primary volumes. If there are multiple volumes with the same preference weight, SMS selects one of the volumes at random.

Volumes that meet the requested MSR are preferred over volumes that do not meet the requested performance. A volume is considered to meet the requested performance if the volume's performance is within a predetermined range of the requested MSR.

The throughput of striped data sets is gated by the slowest device if the striped set includes devices of varying data delivery capabilities.

7.2.2 Rules for striping volume selection

For both guaranteed and non-guaranteed space allocations, the following key rules apply for striping volume selection:

- Storage Groups containing mixed device types are not considered.
- Target volumes:
  - An SDR of 0 results in a target stripe count of 1 for both guaranteed and non-guaranteed space requests, and conventional volume selection occurs. A nonzero SDR that is 1 when divided by the device value results in a target stripe count of 1, but striping volume selection is used instead of conventional volume selection.

  - For non-guaranteed space, the number of target volumes is computed by dividing the SDR that is specified in the Storage Class by a value of 3 for 3380 devices, by a value of 4 for 3390 devices, and rounding up the result, if required. For example, an SDR of 18 results in a target stripe count of 6 on a 3380 device and a target stripe count of 5 (after rounding up) on a 3390 device.
  - For guaranteed space, if the SDR value is 1 or greater, it is ignored, in which case, the target number of stripes is the greater of either the volume count that is specified or the number of specified volume serial numbers. All specified volumes must be in the same SMS Storage Group for a guaranteed space request. SMS assumes that the amount of space that the user wants is the target number of stripes times the specified primary space.
- All temporary data sets with a volume count greater than one are allocated as non-striped.
- The volume must be able to satisfy the primary space requested by the number of stripes.
- The maximum number of stripes (volumes) for VSAM data sets is 16. The maximum number of stripes for physical sequential (PS) data sets is 59.
- The maximum number of extents is five for each space allocation.
- The maximum number of extents per stripe is 123. For VSAM data sets, the maximum number of extents is:
  - Per volume is 123.
  - Per component is 7257.
- The minimum allocation is one track per stripe.
- The maximum allocation can exceed the 64 K track limit.
- All stripes must be able to satisfy the secondary space allocation (divided by the number of stripes) during extend processing or the allocation fails. The secondary space amount is divided by the number of stripes for both guaranteed and non-guaranteed space requests. OPEN, CLOSE, and EOV perform this calculation.
- Non-VSAM multistriped data sets cannot be extended to additional volumes. Striped VSAM data sets can be extended to additional volumes.
- Volume fragmentation information is not available to SMS at volume selection time. An allocation failure by DADSM because of fragmentation results in striping volume re-selection.
- Primary space
  - For non guaranteed space, the volume must be able to satisfy the primary space that is requested divided by the number of stripes. For example, if primary space is 15 MB and the number of stripes is three, the volume must be able to satisfy an allocation of 5 MB.
  - For guaranteed space, for requests that contain specified volume serial numbers, each stripe must be able to satisfy the requested primary space (15 MB in our example).
- Secondary space amount is divided by the number of stripes and rounded up for each volume.

The following rules are true for guaranteed space allocations:

- If you explicitly specify volume serial numbers with guaranteed space and the target number of stripes is equal to the number of volume serial numbers that you specify, SMS must allocate the primary space requested on each of these volumes. If this is not possible, the allocation fails.

- If you do not specify any volume serial numbers, then the target number of stripes is equal to the volume count. SMS tries to select the same number of volumes, but settles for less if this number is unavailable. If fewer stripes are allocated for the non-VSAM data sets, SMS increases the allocation per volume to compensate for the fewer stripes. For VSAM data sets, the primary space requested will be allocated on each stripe regardless of whether the target number of stripes is acquired or not.
- For VSAM data sets, if the number of guaranteed space volumes exceeds sixteen (which is the maximum number of stripes for a VSAM data component), the number of guaranteed space volumes will be reduced to sixteen with the remaining volumes becoming candidates for secondary space.
- If the target number of stripes is higher than the number of volume serial numbers you specify, SMS must select all the specified volumes plus additional ones. These nonspecific volumes are not mandatory and if none are available, SMS allocates the primary quantity on each of the specified volumes.

7.2.3 Requirements for data set striping

The following items are the requirements for data set striping:

- Volumes must be behind one of the following controllers:
  - Controllers that are ESCON®-attached and support concurrent copy
  - 3990-6 controllers
  - 3990-3 controllers that are Extended Platform and ESCON-attached
  - 3990-3 controllers that have the RAMAC support-level microcode
  - 9394 controller
  - 9343 controller with cache
  - IBM RAMAC Virtual Array
  - IBM Enterprise Storage Server®
- Volumes must be ENABLED or QUIESCED and varied ONLINE.

7.2.4 Migration and coexistence considerations

None.

7.3 SMS health check infrastructure

In z/OS V1.11, the Health Check infrastructure for SMS is implemented in preparation to support SMS future checks.

SMS Health Check provides the detection of potential configuration issues and helps the system administrator to modify the configuration to prevent potential problems. SMS Health Check utilizes the IBM Health Checker interface to allow a centralized health report for SMS.

The SMS Health checks are not provided in this release.

8

Utilities

In this chapter, we discuss changes to the utilities in DFSMS V1.11.

IDCAMS, a utility for managing data sets in the system catalogs, has been enhanced to allow more complicated data set name specifications.

In this chapter, we discuss:

- IDCAMS DELETE MASK|NOMASK general information in “IDCAMS DELETE MASK|NOMASK general information” on page 240
- IDCAMS DELETE in NO MASK mode in “IDCAMS DELETE in NOMASK mode” on page 240
- IDCAMS DELETE in MASK mode in “IDCAMS DELETE in MASK mode” on page 245

8.1 IDCAMS DELETE MASK|NOMASK general information

In DFSMS V1R11, the IDCAMS data set delete function is extended through the MASK operand to allow more flexibility in deleting data sets.

The data set(s) to be processed are specified in the same manner as previously, that is, by using the entryname operand, but the interpretation of that operand is altered by the NOMASK|MASK operand.

NOMASK is the default under which the DELETE function operates, as in prior releases, using the Generic File locate.

MASK causes the mode to switch to use the Catalog Search Interface (CSI) to locate data sets. This, or course, means that only cataloged data sets can be deleted using the MASK operand.

In this chapter, we provide examples that contrast the use of the NOMASK (default) operand for specification of data sets to be deleted with the MASK operand specification.

Note: These enhancements to the IDCAMS DELETE function are not available through the TSO interface to IDCAMS functions. Similar functions are, however, available through ISPF functions.

8.2 IDCAMS DELETE in NOMASK mode

In this section, we review briefly the original IDCAMS DELETE statement.

In Figure 8-1 we show the general form of the DELETE statement. We do not show the other optional operands.

```
DELETE entryname
or
DELETE (entryname1,entryname2....)
```

Figure 8-1 IDCAMS DELETE statement

The entryname parameter specifies the data set name to be deleted. It may by fully qualified or use a generic qualifier (*). The generic qualifier may be used in place of any qualifier in the data set name other than the first.

The * qualifier may be repeated in the entryname specification, but each time it is used, it represents the whole qualifier.

Multiple names may be specified in one DELETE statement, in which case the list of names must be enclosed in parentheses.

To illustrate this form of the IDCAMS DELETE command, the following data sets are used. For this example, we specify the same first and second qualifiers:

- MHLRES1.BBB.CCC.DDD
- MHLRES1.BBB.DDD.DDD
- MHLRES1.BBB.CCC.DDD.EEE
- MHLRES1.BBB.CCC

Four examples of the DELETE statement are used in our examples:

- DELETE MHLRES1.BBB.XXX.*

  This statement does not result in the deletion of data sets, because there are no data sets where the fourth qualifier matches XXX.

- DELETE MHLRES1.BBB.*

  The statement results in the deletion of the MHLRES1.BBB.CCC data set, because only MHLRES1.BBB.CCC is a three-part data set name where the first two parts explicitly match the selection.

- DELETE MHLRES1.BBB.*.DDD

  This statement results in the selection of the MHLRES1.BBB.CCC.DDD and MHLRES1.BBB.DDD.DDD data sets, because they are four-part names where the first, second, and fourth parts match the selection.

- DELETE MHLRES1.BBB.*.DDD.EEE

  This statement results in the deletion of the MHLRES1.BBB.CCC.DDD.EEE data set, because it is the only five-part name and the first, second, fourth, and fifth parts match the selection.

  In all of these cases, the * qualifier matches any qualifier.

In Figure 8-2, we show an example of a job that allocates data sets and then deletes them using various forms of the data set name specified in the entryname.

```
000100 //MHLRES1D JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
000200 // NOTIFY=&SYSUID,TIME=1440,REGION=6M
000300 /*JOBPARM L=999,SYSAFF=*
000400 //*
000500 //* JOB TO DEMONSTRATE USE OF GENERIC DELETE
000600 //*
000700 //ALLOC1 EXEC PGM=IEFBR14
000800 //DD1   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
000900 //      DSN=MHLRES1.BBB.CCC.DDD
001000 //DD2   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
001100 //      DSN=MHLRES1.BBB.DDD.DDD
001200 //DD3   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
001300 //      DSN=MHLRES1.BBB.CCC.DDD.EEE
001400 //DD4   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
001500 //      DSN=MHLRES1.BBB.CCC
001600 //DELETE1   EXEC PGM=IDCAMS
001700 //SYSPRINT  DD SYSOUT=*
001800 //SYSIN     DD  *
001900   LISTC LVL(MHLRES1.BBB)
002000   DELETE MHLRES1.BBB.XXX.*
002100   DELETE MHLRES1.BBB.*
002200   DELETE MHLRES1.BBB.*.DDD
002300   DELETE MHLRES1.BBB.*.DDD.EEE
002400   LISTC LVL(MHLRES1.BBB)
002500 //ALLOC2 EXEC PGM=IEFBR14
002600 //DD1   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
002700 //      DSN=MHLRES1.BBB.CCC.DDD
002800 //DD2   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
002900 //      DSN=MHLRES1.BBB.DDD.DDD
003000 //DD3   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
003100 //      DSN=MHLRES1.BBB.CCC.DDD.EEE
003200 //DD4   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
003300 //      DSN=MHLRES1.BBB.CCC
003400 //DELETE2   EXEC PGM=IDCAMS
003500 //SYSPRINT  DD SYSOUT=*
003600 //SYSIN     DD  *
003700   LISTC LVL(MHLRES1.BBB)
003800   DELETE (MHLRES1.BBB.XXX.*,MHLRES1.BBB.*)
003900   DELETE (MHLRES1.BBB.*.DDD,MHLRES1.BBB.*.DDD.EEE)
004000   LISTC LVL(MHLRES1.BBB)
```

Figure 8-2 IDCAMS example job to illustrate generic DELETE statements

The ALLOC step creates the data sets, and the DELETE step deletes them. The ALLOC and DELETE steps are repeated to illustrate the same function when using the DELETE statements with more than one entry name on each statement.

The DELETE steps include a listing of the catalog to show the before and after states.

In Figure 8-3, we show the output from the IDCAMS DELETE1 part of the example job shown in Figure 8-2 on page 242. The statistics part of the LISTC output has been suppressed.

```
IDCAMS  SYSTEM SERVICES                                            TIME:
15:40:36

  LISTC LVL(MHLRES1.BBB)
00190003
IDCAMS  SYSTEM SERVICES                                            TIME:
15:40:36
                            LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.DDD.DDD
     IN-CAT --- UCAT.VSBOX01
IDCAMS  SYSTEM SERVICES                                            TIME:
15:40:36

  DELETE MHLRES1.BBB.XXX.*
00200003
IDC3012I ENTRY MHLRES1.BBB.XXX. NOT FOUND
IDC3007I ** VSAM CATALOG RETURN CODE IS 8
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

  DELETE MHLRES1.BBB.*
00210003
IDC0550I ENTRY (A) MHLRES1.BBB.CCC DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DELETE MHLRES1.BBB.*.DDD
00220003
IDC0550I ENTRY (A) MHLRES1.BBB.CCC.DDD DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.DDD.DDD DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DELETE MHLRES1.BBB.*.DDD.EEE
00230003
IDC0550I ENTRY (A) MHLRES1.BBB.CCC.DDD.EEE DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  LISTC LVL(MHLRES1.BBB)
00240003
IDC3012I ENTRY MHLRES1.BBB NOT FOUND
IDC3007I ** VSAM CATALOG RETURN-CODE IS 8
IDC1566I ** MHLRES1.BBB NOT LISTED
```

Figure 8-3 IDCAMS generic delete (step DELETE1) example job output

In Figure 8-4, we show the output from the IDCAMS DELETE2 part of the example job shown in Figure 8-2 on page 242.

```
IDCAMS  SYSTEM SERVICES                                          TIME:
15:40:37

  LISTC LVL(MHLRES1.BBB)
00370003
IDCAMS  SYSTEM SERVICES                                          TIME:
15:40:37
                              LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.DDD.DDD
     IN-CAT --- UCAT.VSBOX01
IDCAMS  SYSTEM SERVICES                                          TIME:
15:40:37

  DELETE (MHLRES1.BBB.XXX.*,MHLRES1.BBB.*)
00380003
IDC3012I ENTRY MHLRES1.BBB.XXX. NOT FOUND
IDC3007I ** VSAM CATALOG RETURN CODE IS 8
IDC0550I ENTRY (A) MHLRES1.BBB.CCC DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

  DELETE (MHLRES1.BBB.*.DDD,MHLRES1.BBB.*.DDD.EEE)
00390003
IDC0550I ENTRY (A) MHLRES1.BBB.CCC.DDD DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.DDD.DDD DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.CCC.DDD.EEE DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  LISTC LVL(MHLRES1.BBB)
00400003
IDC3012I ENTRY MHLRES1.BBB NOT FOUND
IDC3007I ** VSAM CATALOG RETURN-CODE IS 8
IDC1566I ** MHLRES1.BBB NOT LISTED
```

Figure 8-4 IDCAMS generic delete (step DELETE2) example job output

In this case, the results are the same as for the DELETE1 step, illustrating that even though part of the first DELETE statement did not work, the remaining part did work just as though the two statements had been in separate statements. Note that the condition code for that DELETE statement was 8, reflecting the fact that part of the statement was not successful.

8.3 IDCAMS DELETE in MASK mode

In this section, we introduce the new filter capabilities that available when you use MASK mode.

When IDCAMS is used in MASK mode, the data set lookup is done by the CSI, and then the results are passed back to ICAMS to perform the deletion. When in the prior (NOMASK) mode, the catalog lookup and deletion are done by IDCAMS.

In Figure 8-5, we show the general form of the DELETE statement when using the MASK operand.

```
DELETE entryname MASK
```

Figure 8-5 IDCAMS DELETE statement using the MASK operand

When using MASK mode, the specification of the data set name may include filter keys to define the data sets to be processed.

Only one data set name may be specified on each DELETE statement.

The MASK operand must not be specified in conjunction with the following operands:

- TRUENAME (TRUENAME)
- Non-VSAM Volume record (NVR)
- VSAM Volume Record (VVR)
- PDSE/PDS member of a data set
- Library Entry (LIBRARYENTRY)
- Tape Volume entry (VOLUMEENTRY)

In practical terms, when using the MASK keyword, unless the data sets are cataloged in the MASTER catalog, the catalog that they are in must be specified by using the CATALOG operand.

8.3.1 Filter keys

When using the MASK operand, the single generic qualifier (*) that was available prior to DFSMS V1R11 has been replaced by three filter keys:

- The ** filter key represents one or more data set qualifiers, each of which have a length of 1 to 8. The filter key must be preceded by a period and followed by a period or a blank. If it is followed by a period, there must be another qualifier.
- The * filter key represents one data set qualifier with a length of 1 to 8.
- The % filter key represents one character, and within one part of a data set name there may between 1 and 8 instances of %, depending on the presence of other characters or numbers (numbers must only appear after an initial % or character).

We use the following data sets to illustrate this form of the IDCAMS DELETE command. We specify the same first and second qualifiers (MHLRES1.BBB) for data sets 1 to 4, and the same first and a different second qualifier (MHLRES1.MMM) for data sets 5 to 8 listed:

1. MHLRES1.BBB.CCC.DDD
2. MHLRES1.BBB.DDD.DDD
3. MHLRES1.BBB.CCC.DDD.EEE
4. MHLRES1.BBB.CCC

5. MHLRES1.MMM.AAC.BBC.EEE
6. MHLRES1.MMM.ABC.BBC.EEE
7. MHLRES1.MMM.ADC.BBDD.EEEE
8. MHLRES1.MMM.ADC.BCCD.EEEE

Four examples of the DELETE statement are used in our examples:

- DELETE MHLRES1.BBB.** MASK

  The statement results in the deletion of the MHLRES1.BBB.CCC.DDD, MHLRES1.BBB.DDD.DDD, MHLRES1.BBB.CCC.DDD.EEE, and MHLRES1.BBB.CCC data sets.

- DELETE MHLRES1.MMM.** MASK

  The statement results in the deletion of the MHLRES1.MMM.AAC.BBC.EEE, MHLRES1.MMM.ABC.BBC.EEE, MHLRES1.MMM.ADC.BBDD.EEEE, and MHLRES1.MMM.ADC.BCCD.EEEE data sets.

- DELETE MHLRES1.MMM.A%C.BBC.EEE

  The statement results in the deletion of the MHLRES1.MMM.AAC.BBC.EEE and MHLRES1.MMM.ABC.BBC.EEE data sets.

- DELETE MHLRES1.MMM.ADC.B%%D.EEEE

  The statement results in the deletion of MHLRES1.MMM.ADC.BBDD.EEEE and MHLRES1.MMM.ADC.BCCD.EEEE data sets.

Note: The filter keys are very powerful, so you must be more careful about using them than the generic qualifier. More data sets may be selected as a result of the CSI lookup than would have occurred using the NOMASK mode.

The examples we ran show how the range of data sets selected increases when using the MASK operand.

In Figure 8-6, we show the JCL for a job that uses MASK and key filter statements. The job allocates the data sets, lists them, deletes them, and then confirms deletion by again attempting to list them.

Because the data sets are cataloged in USERCAT, the CATALOG operand has to be used to locate them.

```
//MHLRES1M JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*
//* JOB TO DEMONSTRATE USE OF MASK
//*
//ALLOC1 EXEC PGM=IEFBR14
//DD1   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD
//DD2   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.DDD.DDD
//DD3   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD.EEE
//DD4   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC
//DD5   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.AAC.BBC.EEE
//DD6   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ABC.BBC.EEE
//DD7   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BBDD.EEEE
//DD8   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BCCD.EEEE
//DELETE1   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD  *
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
  DELETE MHLRES1.BBB.**           MASK CATALOG(UCAT.VSBOX01)
  DELETE MHLRES1.MMM.**           MASK CATALOG(UCAT.VSBOX01)
  DELETE MHLRES1.MMM.A%C.BBC.EEE  MASK CATALOG(UCAT.VSBOX01)
  DELETE MHLRES1.MMM.ADC.B%%D.EEEE MASK CATALOG(UCAT.VSBOX01)
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
```

Figure 8-6 IDCAMS example job (#1) to illustrate DELETE statements using MASK and filters

In Example 8-1, we show the output from the IDCAMS steps from the job shown in Figure 8-6. The statistics parts of the IDCAMS LISTC output has been suppressed.

Example 8-1 IDCAMS output from a job (#1) using MASK and filters to DELETE data sets

```
IDCAMS  SYSTEM SERVICES                                        TIME: 16:47:09

  LISTC LVL(MHLRES1.BBB)
IDCAMS  SYSTEM SERVICES                                        TIME: 16:47:09
                            LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC
     IN-CAT --- UCAT.VSBOX01
```

```
NONVSAM ------- MHLRES1.BBB.CCC.DDD
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.DDD.DDD
     IN-CAT --- UCAT.VSBOX01
IDCAMS  SYSTEM SERVICES                                        TIME: 16:47:09

  LISTC LVL(MHLRES1.MMM)
IDCAMS  SYSTEM SERVICES                                        TIME: 16:47:09
                            LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.AAC.BBC.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ABC.BBC.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ADC.BBDD.EEEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ADC.BCCD.EEEE
     IN-CAT --- UCAT.VSBOX01
IDCAMS  SYSTEM SERVICES                                        TIME: 16:47:09

  DELETE MHLRES1.BBB.**            MASK CATALOG(UCAT.VSBOX01)
IDC0550I ENTRY (A) MHLRES1.BBB.CCC DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.CCC.DDD DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.CCC.DDD.EEE DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.DDD.DDD DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DELETE MHLRES1.MMM.**            MASK CATALOG(UCAT.VSBOX01)
IDC0550I ENTRY (A) MHLRES1.MMM.AAC.BBC.EEE DELETED
IDC0550I ENTRY (A) MHLRES1.MMM.ABC.BBC.EEE DELETED
IDC0550I ENTRY (A) MHLRES1.MMM.ADC.BBDD.EEEE DELETED
IDC0550I ENTRY (A) MHLRES1.MMM.ADC.BCCD.EEEE DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DELETE MHLRES1.MMM.A%C.BBC.EEE   MASK CATALOG(UCAT.VSBOX01)
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.MMM.A%C.BBC.EEE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

  DELETE MHLRES1.MMM.ADC.B%%D.EEEE MASK CATALOG(UCAT.VSBOX01)
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.MMM.ADC.B%%D.EEEE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

  LISTC LVL(MHLRES1.BBB)
IDC3012I ENTRY MHLRES1.BBB NOT FOUND
IDC3007I ** VSAM CATALOG RETURN-CODE IS 8
IDC1566I ** MHLRES1.BBB NOT LISTED
  LISTC LVL(MHLRES1.MMM)
IDC3012I ENTRY MHLRES1.MMM NOT FOUND
IDC3007I ** VSAM CATALOG RETURN-CODE IS 8
IDC1566I ** MHLRES1.MMM NOT LISTED
```

This output shows that the first and second DELETE statements each deleted four data sets. This means that the third and fourth DELETE statements did not complete successfully because the selected data sets had already been deleted.

In Figure 8-7, we show the JCL for a job very similar to the one in Figure 8-6 on page 247, but in this case, the DELETE statement that deletes all the MLRES1.MMM.** data sets has been removed, which allows the more selective statements using the % filter key to be effective.

```
//MHLRES1N JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*
//* JOB TO DEMONSTRATE USE OF MASK
//*
//ALLOC1 EXEC PGM=IEFBR14
//DD1   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD
//DD2   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.DDD.DDD
//DD3   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD.EEE
//DD4   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC
//DD5   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.AAC.BBC.EEE
//DD6   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ABC.BBC.EEE
//DD7   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BBDD.EEEE
//DD8   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BCCD.EEEE
//DELETE1   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD  *
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
  DELETE MHLRES1.BBB.**           MASK CATALOG(UCAT.VSBOX01)
  DELETE MHLRES1.MMM.A%C.BBC.EEE  MASK CATALOG(UCAT.VSBOX01)
  DELETE MHLRES1.MMM.ADC.B%%D.EEEE MASK CATALOG(UCAT.VSBOX01)
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
```

Figure 8-7 IDCAMS example job (#2) to illustrate DELETE statements using MASK and filters

In Figure 8-8, we show the output from the IDCAMS parts of the job shown in Figure 8-7 on page 249. The statistics parts of the IDCAMS LISTC statements have been suppressed.

```
IDCAMS  SYSTEM SERVICES                                          TIME: 17:23:34

  LISTC LVL(MHLRES1.BBB)
IDCAMS  SYSTEM SERVICES                                          TIME: 17:23:34
                            LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.DDD.DDD
     IN-CAT --- UCAT.VSBOX01
IDCAMS  SYSTEM SERVICES                                          TIME: 17:23:34

  LISTC LVL(MHLRES1.MMM)
IDCAMS  SYSTEM SERVICES                                          TIME: 17:23:34
                            LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.AAC.BBC.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ABC.BBC.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ADC.BBDD.EEEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ADC.BCCD.EEEE
     IN-CAT --- UCAT.VSBOX01
IDCAMS  SYSTEM SERVICES                                          TIME: 17:23:34

  DELETE MHLRES1.BBB.**           MASK CATALOG(UCAT.VSBOX01)
IDC0550I ENTRY (A) MHLRES1.BBB.CCC DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.CCC.DDD DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.CCC.DDD.EEE DELETED
IDC0550I ENTRY (A) MHLRES1.BBB.DDD.DDD DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DELETE MHLRES1.MMM.A%C.BBC.EEE   MASK CATALOG(UCAT.VSBOX01)
IDC0550I ENTRY (A) MHLRES1.MMM.AAC.BBC.EEE DELETED
IDC0550I ENTRY (A) MHLRES1.MMM.ABC.BBC.EEE DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DELETE MHLRES1.MMM.ADC.B%%D.EEEE MASK CATALOG(UCAT.VSBOX01)
IDC0550I ENTRY (A) MHLRES1.MMM.ADC.BBDD.EEEE DELETED
IDC0550I ENTRY (A) MHLRES1.MMM.ADC.BCCD.EEEE DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  LISTC LVL(MHLRES1.BBB)
IDC3012I ENTRY MHLRES1.BBB NOT FOUND
IDC3007I ** VSAM CATALOG RETURN-CODE IS 8
IDC1566I ** MHLRES1.BBB NOT LISTED
  LISTC LVL(MHLRES1.MMM)
IDC3012I ENTRY MHLRES1.MMM NOT FOUND
IDC3007I ** VSAM CATALOG RETURN-CODE IS 8
IDC1566I ** MHLRES1.MMM NOT LISTED
```

Figure 8-8 IDCAMS output from a job (#2) using MASK and filters to DELETE data sets

This output shows that the first DELETE statement deleted four data sets, and the second and third statements each deleted two data sets.

8.3.2 MASK mode conditions

You should be aware of certain conditions that are connected to the MASK mode. We discuss these conditions in this section.

Filter keys without MASK

The ** and % filter keys are not permitted when the MASK operand is omitted.

The * filter key is the same as the generic key, so it will be accepted if the MASK operand is omitted.

In Figure 8-9, we show a job that attempts to use the ** and % filter keys without specifying MASK. (The job also includes the correctly specified DELETE statements for cleanup.)

```
//MHLRES1O JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=144O,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*
//* JOB TO DEMONSTRATE USE OF FILTER KEYS WITHOUT MASK
//*
//ALLOC1 EXEC PGM=IEFBR14
//DD1   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD
//DD2   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.DDD.DDD
//DD3   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD.EEE
//DD4   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC
//DD5   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.AAC.BBC.EEE
//DD6   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ABC.BBC.EEE
//DD7   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BBDD.EEEE
//DD8   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BCCD.EEEE
//DELETE1   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD  *
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
  DELETE MHLRES1.BBB.**.
  DELETE MHLRES1.MMM.**
  DELETE MHLRES1.MMM.A%C.BBC.EEE
  DELETE MHLRES1.MMM.ADC.B%%D.EEEE
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
  DELETE MHLRES1.BBB.** ^^^^^^^^MASK CATALOG(UCAT.VSBOX01)
  DELETE MHLRES1.MMM.**            MASK CATALOG(UCAT.VSBOX01)
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
```

Figure 8-9 IDCAMS JCL demonstrating the use of filter keys without the MASK operand

In Figure 8-10, we show the IDCAMS output from the job in Figure 8-9 on page 251, together with the IDCAMS LISTC statements that show that the data sets were not deleted. The statistics part of the LISTC output has been suppressed.

```
IDCAMS  SYSTEM SERVICES                                          TIME:
17:43:28

  DELETE MHLRES1.BBB.**.
IDC3203I ITEM 'MHLRES1.BBB.**.' DOES NOT ADHERE TO RESTRICTIONS
IDC3202I ABOVE TEXT BYPASSED UNTIL NEXT COMMAND. CONDITION CODE IS 12

  DELETE MHLRES1.MMM.**
IDC2901I MASKING ENTRY NAME REQUIRES A KEYWORD MASK TO BE SPECIFIED
IDC3003I FUNCTION TERMINATED. CONDITION CODE IS 12

  DELETE MHLRES1.MMM.A%C.BBC.EEE
IDC2901I MASKING ENTRY NAME REQUIRES A KEYWORD MASK TO BE SPECIFIED
IDC3003I FUNCTION TERMINATED. CONDITION CODE IS 12

  DELETE MHLRES1.MMM.ADC.B%%D.EEEE
IDC2901I MASKING ENTRY NAME REQUIRES A KEYWORD MASK TO BE SPECIFIED
IDC3003I FUNCTION TERMINATED. CONDITION CODE IS 12

  LISTC LVL(MHLRES1.BBB)
IDCAMS  SYSTEM SERVICES                                          TIME:
17:43:28
                          LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.DDD.DDD
     IN-CAT --- UCAT.VSBOX01
  LISTC LVL(MHLRES1.MMM)
IDCAMS  SYSTEM SERVICES                                          TIME:
17:43:28
                          LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.AAC.BBC.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ABC.BBC.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ADC.BBDD.EEEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ADC.BCCD.EEEE
     IN-CAT --- UCAT.VSBOX01
```

Figure 8-10 IDCAMS output illustrating the use of filter keys without the MASK operand

This output shows that the use of the ** and % filter keys have been detected without the required MASK operand. The first DELETE statement illustrates an incorrect specification; there is a trailing period (.) after the ** filter key.

It also confirms, via the output from the LISTC statements, that the data sets were not deleted.

MASK mode without the CATALOG specification

In contrast to the catalog lookup that occurs when you are not using the MASK function, it is necessary to ensure that an appropriate catalog is being used. If no effective catalog is found, the results can be misleading.

In Figure 8-11, we show a job that attempts to delete data sets using the MASK operand but without specifying the catalog to be used. The job also includes the correctly specified DELETE statements for cleanup.

```
//MHLRES1P JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*
//* JOB TO DEMONSTRATE USE OF FILTER KEYS WITH MASK
//* BUT WITHOUT CATALOG SPECIFICATION
//*
//ALLOC1 EXEC PGM=IEFBR14
//DD1   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD
//DD2   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.DDD.DDD
//DD3   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD.EEE
//DD4   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC
//DD5   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.AAC.BBC.EEE
//DD6   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ABC.BBC.EEE
//DD7   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BBDD.EEEE
//DD8   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BCCD.EEEE
//DELETE1   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD  *
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
  DELETE MHLRES1.BBB.**            MASK
  DELETE MHLRES1.MMM.**            MASK
  DELETE MHLRES1.MMM.A%C.BBC.EEE   MASK
  DELETE MHLRES1.MMM.ADC.B%%D.EEEE MASK
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
  DELETE MHLRES1.BBB.**            MASK CATALOG(UCAT.VSBOX01)
  DELETE MHLRES1.MMM.**            MASK CATALOG(UCAT.VSBOX01)
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
```

Figure 8-11 IDCAMS JCL that shows the use of the filter keys for the MASK operand

In Figure 8-12, we show the output from the IDCAMS steps from the job in Figure 8-11 on page 253, which demonstrates the failure to locate data sets.

```
IDCAMS  SYSTEM SERVICES                                          TIME:
18:15:37

  DELETE MHLRES1.BBB.**           MASK
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.BBB.**
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

  DELETE MHLRES1.MMM.**           MASK
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.MMM.**
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

  DELETE MHLRES1.MMM.A%C.BBC.EEE   MASK
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.MMM.A%C.BBC.EEE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

  DELETE MHLRES1.MMM.ADC.B%%D.EEEE MASK
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.MMM.ADC.B%%D.EEEE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
  LISTC LVL(MHLRES1.BBB)
IDCAMS  SYSTEM SERVICES                                          TIME:
18:15:37
                          LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.CCC.DDD.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.BBB.DDD.DDD
     IN-CAT --- UCAT.VSBOX01
IDCAMS  SYSTEM SERVICES                                          TIME:
18:15:37

  LISTC LVL(MHLRES1.MMM)
IDCAMS  SYSTEM SERVICES                                          TIME:
18:15:37
                          LISTING FROM CATALOG -- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.AAC.BBC.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ABC.BBC.EEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ADC.BBDD.EEEE
     IN-CAT --- UCAT.VSBOX01
NONVSAM ------- MHLRES1.MMM.ADC.BCCD.EEEE
     IN-CAT --- UCAT.VSBOX01
```

Figure 8-12 IDCAMS JOB output using filter keys and the MASK operand without adequate catalog specification

This output shows that as far as the IDCAMS/CSI search process is concerned, no data sets were found. This is misleading. Reference to the output in Figure 8-8 on page 250, for example, show that if the CATALOG parameter is specified appropriately, the data sets are located.

8.3.3 Use of multiple MASK specifications

The release of the Multiple MASK specification does not allow more than one element of a data set name to have a mask specified. This restriction is being addressed through APAR OA29980.

If APAR OA29980 is not installed, any data set name specification with more than one element containing a mask is rejected.

In Figure 8-13, we show a copy of the job that was used in Figure 8-11 on page 253 that attempts to delete data sets using the MASK operand but without specifying the catalog to be used. One DELETE command has been added that has two elements with masks. The job also includes the correctly specified DELETE statements for cleanup.

```
//MHLRES1P JOB (999,POK),'MHLRES2',CLASS=A,MSGCLASS=T,
// NOTIFY=&SYSUID,TIME=1440,REGION=6M
/*JOBPARM L=999,SYSAFF=*
//*
//* JOB TO DEMONSTRATE USE OF FILTER KEYS WITH MASK
//* BUT WITHOUT CATALOG SPECIFICATION
//*
//ALLOC1 EXEC PGM=IEFBR14
//DD1   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD
//DD2   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.DDD.DDD
//DD3   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC.DDD.EEE
//DD4   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.BBB.CCC
//DD5   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.AAC.BBC.EEE
//DD6   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ABC.BBC.EEE
//DD7   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BBDD.EEEE
//DD8   DD DISP=(,CATLG),UNIT=SYSDA,SPACE=(TRK,1),
//      DSN=MHLRES1.MMM.ADC.BCCD.EEEE
//DELETE1   EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD  *
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
  DELETE MHLRES1.BBB.**           MASK
  DELETE MHLRES1.MMM.**           MASK
  DELETE MHLRES1.MMM.A%C.BBC.EEE  MASK
^DELETE MHLRES1.M%M.A%C.BBC.EEE   MASK
  DELETE MHLRES1.MMM.ADC.B%%D.EEEE MASK
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
  DELETE MHLRES1.BBB.**           MASK CATALOG(UCAT.VSBOX01)
  DELETE MHLRES1.MMM.**           MASK CATALOG(UCAT.VSBOX01)
  LISTC LVL(MHLRES1.BBB)
  LISTC LVL(MHLRES1.MMM)
```

Figure 8-13 IDCAMS JCL use of filter keys and the MASK operand specifying more than one mask

In Figure 8-14, we show the output from the IDCAMS steps from the job in Figure 8-13 on page 256, which demonstrates the failure to locate data sets, and the effect of specifying more than one element with a mask.

```
IDCAMS  SYSTEM SERVICES                                          TIME: 18:49:47
08/26/09     PAGE      7

 DELETE MHLRES1.BBB.**         MASK
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.BBB.**
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

 DELETE MHLRES1.MMM.**         MASK
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.MMM.**
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

 DELETE MHLRES1.MMM.A%C.BBC.EEE   MASK
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.MMM.A%C.BBC.EEE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8

 DELETE MHLRES1.M%M.ADC.B%%D.EEEE MASK
IDC3203I ITEM 'MHLRES1.M%M.ADC.B%%D' DOES NOT ADHERE TO RESTRICTIONS
IDC3202I ABOVE TEXT BYPASSED UNTIL NEXT COMMAND. CONDITION CODE IS 12

 DELETE MHLRES1.MMM.ADC.B%%D.EEEE MASK
IDC2902I NO ENTRIES FOUND FOR MASK MHLRES1.MMM.ADC.B%%D.EEEE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 8
```

Figure 8-14 IDCAMS JOB output demonstrating the effect of a DELETE with two masks

9

IBM System Storage Virtual Tape Facility for Mainframe

In this chapter, we give you an overview of the IBM System Storage Virtual Tape Facility for Mainframe (VTFM) and how you can implement and use a VTFM.

In this chapter, we discuss the following topics:

- Overview
- Terminology
- Description
- Technical information
- Implementation
- Customization
- Starting the VTFM server
- Enabling the HVTUDST utility to write the migration job
- ICF user catalog considerations
- Scratching or reclaiming Virtual Tapes

9.1 Overview

VTFM provides a virtual tape alternative for many IBM System z users. It is compatible with and integrates within existing IBM and IBM compatible System z tape environments, delivering a high performance yet low cost tape virtualization strategy.

By storing its virtual tapes on a standard System z disk, VTFM adds device independence to batch tape processing and enhances disaster recovery by providing tape data with the same resiliency features as System z disk products.

VTFM provides Parallel Access Tape (PAT) support. PAT enables multiple and simultaneous read accesses to virtual tape volumes. z/OS does not support multiple and simultaneous read accesses to a physical tape volume. Figure 9-1 provides an overview of VTFM.

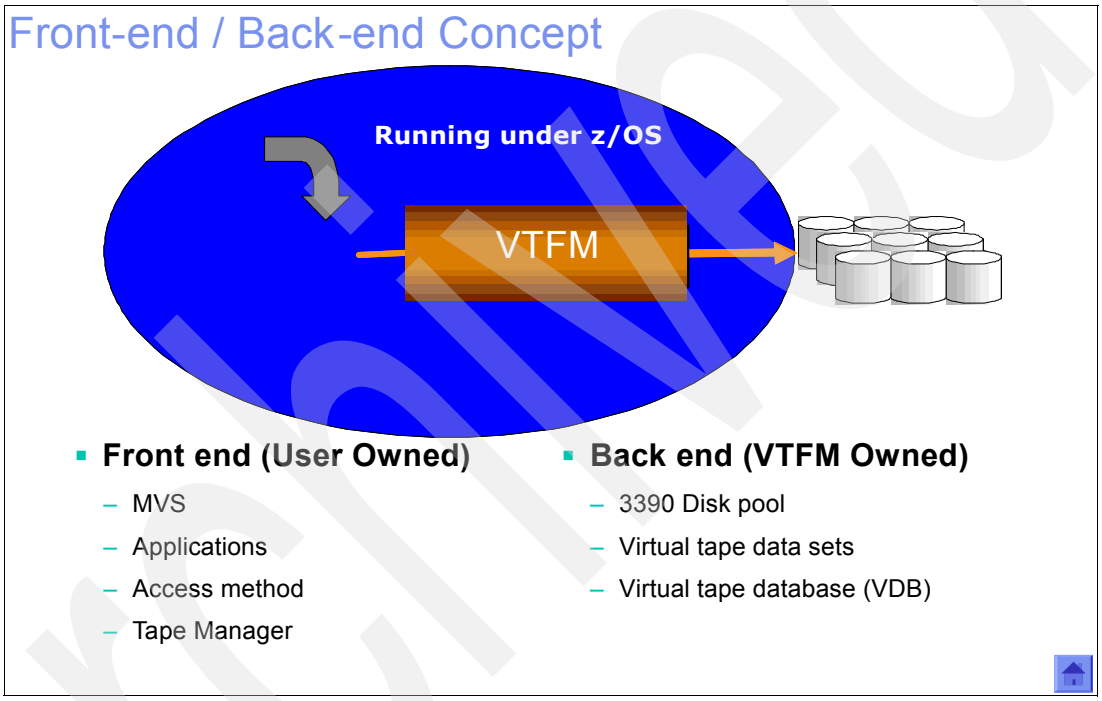


Figure 9-1 VTFM overview

9.2 Terminology

In this chapter, we use the following terminology:

PAT Parallel Access Tape (PAT) is a method for sharing virtual tapes access for reads in a single MVS system.

UIM A Unit Information Module (UIMs) performs the device-dependent part of the operating system configuration definition. There is a UIM for each supported device or device group. Each UIM recognizes and processes the values coded for its device or device group. HCD routines load all UIMs, either IBM or customer supplied, into virtual storage and make calls to the UIMs during initialization, processing of an Add device or Change device request, or when you perform an IPL.

VDB Virtual Tape Database.

Tape data set	The data set written onto a real or virtual tape. This tape data set has an associated tape data set name.
Virtual tape	The tape entity that is similar to a reel tape or a cartridge. This virtual tape has an associated volume serial number.
Virtual tape data set	The disk data set that contains the data written to the virtual tape. This virtual tape data set has an associated disk data set name.
VTFM	Virtual Tape Facility for Mainframe.

9.3 Description

VTFM is an MVS software product that simulates IBM compatible tape controllers, tape drives and tape volumes. It does not introduce new hardware. Real tape devices are replaced by logical, memory resident, and virtual tape devices, and tape volumes are replaced with virtual tape data sets. VTFM simulates IBM Tape Library 3480, 3490, 3590, and 3592 devices. During this process, it remains fully transparent to the host, the MVS applications, the tape management systems, and the users.

VTFM virtual tapes can be configured in a “VTFM complex” that enables concurrent access to tape volumes. VTFM supports standard-label and no-label tapes, as well as multiple data set volumes and multiple volume data sets.

The implementation of VTFM does not require any changes to existing JCL or applications. The user may code rules into the VTFM server parameter list to permit sharing of virtual tapes by applications within the same z/OS.

VTFM can provide up to 256 virtual tape devices per VTFM server and can handle an unlimited number of virtual tapes. Additionally, the virtual tape library can be shared among systems. Additional media capacity is achieved by adding more disk space to the system.

VTFM virtual tapes are managed by the site's tape management system. VTFM virtual tape data sets, being disk data sets, can be system managed using SMS and disk management products. They can be browsed using ISPF or another online browser.

In conjunction with the Extended Remote Copy (XRC) or Peer to Peer Remote Copy (PPRC) mirroring solutions, VTFM offers quick recovery of and access to virtual tape libraries in a disaster recovery situation. XRC or PPRC can also be used to electronically transfer all tape data while migrating or merging data centers. In conjunction with an FTP, VTFM can electronically store virtual tapes to one or more remote off-site storage locations to support business continuity and resiliency.

VTFM runs under z/OS V1.8 or later. Figure 9-2 shows you how VTFM handles virtual tape requests.

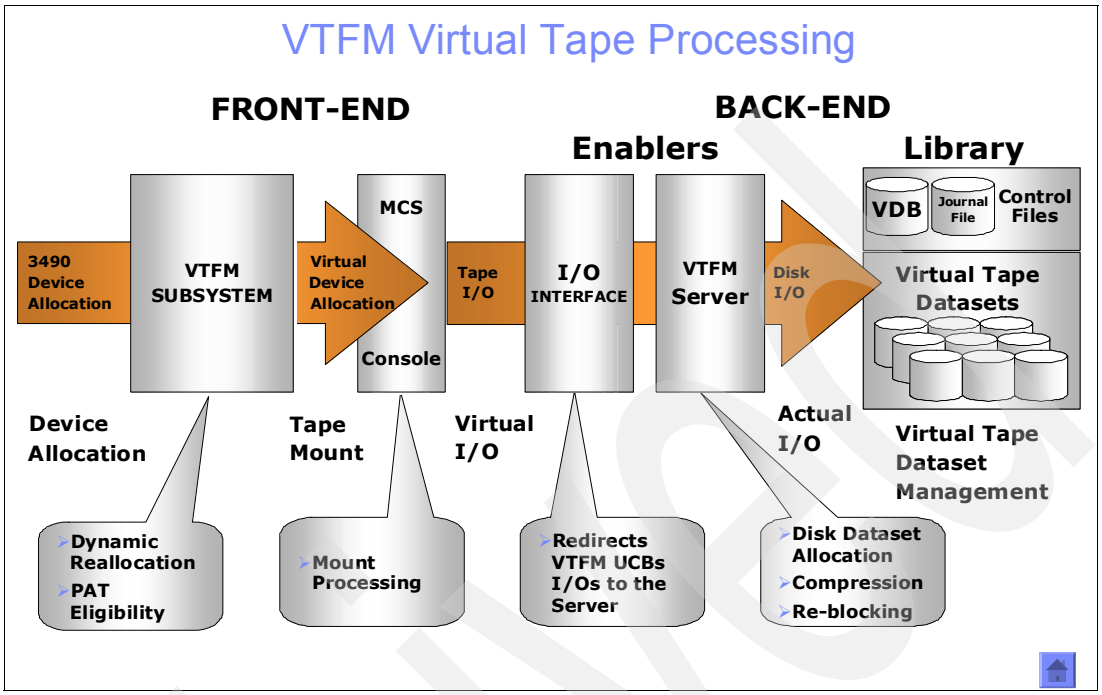


Figure 9-2 VTFM virtual tape processing

9.4 Technical information

In this section, we describe hardware and software requirements and the key points of IBM System Storage VTFM V3.

9.4.1 Hardware requirements

IBM System Storage VTFM V3 must have sufficient disk space available on which to install the product. The minimum required to store the tape is one 3390 disk volume. VTFM must have a console in order to perform system installation and maintenance. One console (monitor, keyboard, and mouse) is required.

9.4.2 Software requirements

This program runs only on System z processors with z/OS V1.8 or later.

The program's specifications and specified operating environment information may be found in documentation accompanying the program, such as a README file, or other information published by IBM, such as an announcement letter. Documentation and other program content may be supplied only in the English language.

9.4.3 Planning information

Physical planning for VTFM is your responsibility. Refer to the planning information provided in *IBM System Storage VTF Mainframe V3.0.0,* GI11-8936. This document can be found on the publication CDROM shipped with the product.

9.4.4 Security, auditability, and control

VTFM V3 uses the security and auditability features of the host hardware, host software, or application software to which it is attached. The customer is responsible for evaluation, selection, and implementation of security features, administrative procedures, and appropriate controls in application systems and communication facilities.

9.4.5 VTFM highlights

Here is a list of the key points of the VTFM product:

- The implementation of VTFM does not require any changes to the existing JCL or applications.
- There is full tape emulation, including support for VOL=REF groups, UNIT=AFF chains, and all tape commands.
- Parallel Access Tape (PAT) allows virtual tapes to be shared by applications within the same MVS system.
- VTFM supports up to 256 virtual tape devices per VTFM server.
- VTFM supports an unlimited number of virtual tapes volumes.
- The VTFM virtual tape library may be shared among systems.
- VTFM is fully integrated with any MVS environment. Its virtual tapes are managed by the site's tape management system. VTFM virtual tape data sets, being disk data sets, can be system managed by all aspects of SMS and disk management products.
- The e-Vault feature electronically stores virtual tapes to one or more remote offsite storage locations.
- *IBM System Storage VTF Mainframe Release Notes*, GC53-1189 provides useful information about upgrading and new product features.

9.5 Implementation

Before performing any of the installation steps described in this chapter, make sure that all installation steps included in the Program Directory are completed successfully. See *IBM System Storage VTF Mainframe,* GI11-8936 for detailed information. Only after you complete successfully the installation steps in the Program Directory can you continue with the customization steps in this chapter. The customization steps included in this chapter are based on the steps in *IBM System Storage VTF Mainframe Program Directory*, GI11-8936. It is assumed that the VTFM product is compliant with SMP/E APPLY and ACCEPT and both the distribution and target libraries are allocated and contain the required data.

9.5.1 Virtual tape candidates

Before implementing VTFM, a decision must be made to define what are the best tape data sets candidates to be reallocated by VTFM to virtual tapes. The decision varies from site to site according to the site's special needs:

- Complete tape replacement: VTFM can be used to completely replace all real tape processing so no real tape device is attached to the mainframe.
- Disaster recovery: VTFM can be used to reallocate disaster recovery data that used to reside on tape and now can be mirrored remotely using a remote replication product.
- Shorter batch window: VTFM can be used to back up application data, thus shortening the backup time and the application down time.
- Short-term backups: VTFM can be used to hold daily, weekly, and monthly backups that eventually expire on disk without any real tape involved.
- Workload sharing of tape data sets: Can be used to parallel process data on tape.
- Tape caching: VTFM can be used to cache the tape work on a diskpool as a station before the data set expires on the cache or is migrated to real tape.
- Electronic vaulting: VTFM can be used to electronically vault tapes offsite instead of using a manual process (the so-called *Truck Access method*).
- Data center migration: VTFM can be used to merge or migrate a data center. The virtual tapes data can be moved over a remote replication product.
- Media for Migration Level 2: VTFM can be used as a media to hold all ML2 data for ABR, CA-Disk, or DFSMShsm.

Data sets that should never be written to virtual tapes

Here is a list of data sets that should never be written to virtual tapes:

- Volume Database (VDB).
- Journal file.
- A user catalog that contains the VDB and the journal file.
- A user catalog that contains the virtual tape data set's entries.
- Master catalog.
- Virtual tape data sets.
- Data sets cataloged in the catalog that contains the VDB. The best way to avoid this situation is to define the VDB in a separate user catalog.

This situation also applies to discrete data sets and to full disk dumps that might contain the above data sets.

VTFM Assessor tool

The VTFM Assessor tool can analyze a customer's tape environment and make recommendations about VTFM diskpools sizing, simulations of tape workload over VTFM, and identification of the most suitable tape candidates for VTFM in the event of a partial tape replacement.

The VTFM Assessor tool is part of IBM Services, which also assists in implementing VTFM strategies ranging from library replacements to full Business Continuance and data recovery schemes.

9.6 Customization

In this section, we describe how to start VTFM on your system and test VTFM functions without affecting your currently running tape environment.

9.6.1 Hardware configuration

Two major hardware types must be made available to VTFM for proper functioning. These hardware types are real disks and virtual tape devices.

The real disks can be of any type, model, and vendor. They are defined to MVS using a standard HCD process, so there are no special requirements. The disks are later managed by any standard available tool.

The customization process assumes the disks are already defined and available. The virtual tape devices definition requires special attention. The following sections take you through the steps to define the new VTFM virtual devices.

Installing the VTFM virtual devices UIM

The VTFM virtual devices simulate IBM Tape Library 3480, 3490, 3590, and 3592 compatible devices. The installation of the VTFM virtual devices is done the same way as though they were real hardware devices.

The VTFM virtual devices are introduced to the hardware configuration with a UIM. Some of the hardware configuration parameters are supplied as defaults with the VTFM default UIM. If the default values conflict with your site's configuration, you may change them through VTFM EXIT002.

The default UIM number is 254 and it supplies six simulations: Two for 3480 devices, two for 3490 devices, and two for 3590 or 3592 devices, as shown in Table 9-1.

Table 9-1 UIM devices

Simulation	Generic	DEVUNIT	UCBTBYT1	PREVVAL
3480	VT3480GN	VT3480DU	39	9013
3480	VT3480G2	VT3480D2	40	9014
3490	VT3490GN	VT3490DU	41	9015
3490	VT3490G2	VT3490D2	42	9016
3590/3592	VT3590GN	VT3590DU	43	9017
3590/3592	VT3350G2	VT3590D2	44	9018

3592 Devices considerations

Although the real 3592 devices can write up to 1, 000 GB (full capacity), VTFM virtual tapes capacity (uncompressed) cannot be larger than 52 GB. Therefore, if 3592 devices are used for duplexing, the real 3592 device should be scaled to 60 GB for optimal performance.

VTFM EXIT001

The VTFM EXIT001 exit is invoked before the buffer is written to virtual tape and after a buffer is read from virtual tape. This exit might change data in the buffer. However, the data length should not be modified.

The sample EXIT001 is stored in the SAMPLIB library and the member name is HVTX001.

VTFM EXIT002

This exit defines one control unit type and a series of attached devices to be supported by the VTFM UIM. These devices are used by the HCD process to define the VTFM virtual devices.

One exit contains definitions for all VTFM installations in a single system. If a new VTFM environment is installed, the exit should be updated with the values of the new installation. This enables one VTFM UIM module to support all VTFM installations in the system.

The sample EXIT002 is stored in the SAMPLIB library and the member name is HVTX002 (Figure 9-3).

```
*
*  -- ONE CONTROL UNIT SUPPORTS UP TO 256 3480/3490/3590/3592 DEVS ----
*
         HVTX002 START,                                                *
               CUNAME=VT3480CU    <--- VIRTUAL CONTROL UNIT
*
```

Figure 9-3 HVTX002 sample to simulate 3480 devices

Where:

START The first occurrence of the macro should be in START mode. This mode should be invoked only once. The following parameters are associated with this mode:

CUNAME A new control unit name to which all the VTFM devices (3480, 3490, 3590, and 3592) will be attached.

Value range A one to eight character string.

Default None.

Note: CUNAME is not a required parameter with the current VTFM version. It is left here for compatibility reasons.

Figure 9-4 shows the sample EXIT002 code to simulate 3480 devices.

```
*
*  -------------------- DEVICE UNIT VT3480DU (3480) -----------------
*
         HVTX002 DEFINE,                                              *
               DEVUNIT=VT3480DU,  <--  DEVICE UNIT TYPE               *
               GENERIC=VT3480GN,       ..GENERIC NAME                 *
               TYPE=3480,              ..BE A 3480 DEVICE             *
               UCBTBYT1=39,            ..UCBTBYT1                     *
               PREFVAL=9013            ..UNIQUE PREFERENCE VALUE
*
*  -------------------- DEVICE UNIT VT3480D2 (3480) -----------------
*
         HVTX002 DEFINE,                                              *
               DEVUNIT=VT3480D2,  <--  DEVICE UNIT TYPE               *
               GENERIC=VT3480G2,       ..GENERIC NAME                 *
               TYPE=3480,              ..BE A 3480 DEVICE             *
               UCBTBYT1=40,            ..UCBTBYT1                     *
               PREFVAL=9014            ..UNIQUE PREFERENCE VALUE
*
```

Figure 9-4 HVTX002 sample to simulate 3480 devices

Figure 9-5 shows the sample EXIT002 code to simulate 3490 devices.

```
*
         HVTX002 DEFINE,                                              *
               DEVUNIT=VT3490DU,  <--- DEVICE UNIT TYPE               *
               GENERIC=VT3490GN,       ..GENERIC NAME                 *
               TYPE=3490,              ..BE A 3490 DEVICE             *
               UCBTBYT1=41,            ..UCBTBYT1                     *
               PREFVAL=9015            ..UNIQUE PREFERENCE VALUE
*
*  -------------------- DEVICE UNIT VT3490D2 (3490) -----------------
*
         HVTX002 DEFINE,                                              *
               DEVUNIT=VT3490D2,  <--  DEVICE UNIT TYPE               *
               GENERIC=VT3490G2,       ..GENERIC NAME                 *
               TYPE=3490,              ..BE A 3490 DEVICE             *
               UCBTBYT1=42,            ..UCBTBYT1                     *
               PREFVAL=9016            ..UNIQUE PREFERENCE VALUE
*
```

Figure 9-5 HVTX002 sample to simulate 3490 devices

Figure 9-6 shows you the sample EXIT002 code to simulate 3590 devices.

```
*
         HVTX002 DEFINE,                                                *
               DEVUNIT=VT3590DU,  <--- DEVICE UNIT TYPE                 *
               GENERIC=VT3590GN,       ..GENERIC NAME                   *
               TYPE=3590,              ..BE A 3590/3592 DEVICE          *
               UCBTBYT1=43,            ..UCBTBYT1                       *
               PREFVAL=9017            ..UNIQUE PREFERENCE VALUE
*
*  -------------------- DEVICE UNIT VT3590D2 (3590/3592) ------------- ---
*
         HVTX002 DEFINE,                                                *
               DEVUNIT=VT3590D2,  <--  DEVICE UNIT TYPE                 *
               GENERIC=VT3590G2,       ..GENERIC NAME                   *
               TYPE=3590,              ..BE A 3590/3592 DEVICE          *
               UCBTBYT1=44,            ..UCBTBYT1                       *
               PREFVAL=9018            ..UNIQUE PREFERENCE VALUE
*
```

Figure 9-6 HVTX002 sample to simulate 3590 devices

Where:

DEFINE In this mode, you can define one attached device unit type along with its unique specific values. The following parameters are associated with this mode:

DEVUNIT A new device unit type attached to the VTFM control unit.

Value range A one to eight character string.

Default None.

GENERIC A new generic name to be assigned to this VTFM device type.

Value range A one to eight character string.

Default None.

TYPE

3480 This device unit simulates a 3480 device.

3490 This device unit simulates a 3490 device.

3590 This device unit simulates a 3590 or 3592 device.

Further selection between 3590 and 3592 is done by using the SIMULATE parameter of the VDEVICE initialization statement.

Value range N/A.

Default None.

UCBTBYT1 The first byte of the four bytes that make up the UCBTYP field of this VTFM virtual device. The other three bytes will be set automatically as x'048080', x'048081', or x'048083', according to the TYPE value specified earlier.

When a data set is cataloged, the UCBTYP of the device on which the data set was mounted is kept in the catalog. This UCBTYP is

then used when the data set is accessed in order to allocate an appropriate device. UCBTBYT1 should not be changed after product implementation is started; otherwise, MVS fails to select devices eligible for virtual tape mount as requested by cataloged data sets.

Value range Any two hexadecimal digit from 00 to FF (except 78).

Default None.

PREFVAL The unique device unit preference value. *MVS Device Validation Support*, SA22-7586 gives a list of preference values used by IBM. Here you should code a value that is not used by IBM for its devices.

Value range Integer from 1 to 9999.

Default None.

END The last occurrence of the macro should be in END mode. This mode should be invoked only once.

After you have made any changes to EXIT002, implement the changes as a USERMOD, as shown in Figure 9-7.

```
//USERMOD  EXEC PGM=GIMSMP,REGION=5120K
//SMPCSI   DD  DISP=SHR,DSN=#smphlq.CSI   <== SMP/E data set HLQ
//SMPRPT   DD  SYSOUT=*
//SMPOUT   DD  SYSOUT=*
//SMPPTFIN DD  DATA,DLM=§§
++USERMOD(HVTU002)          /* SYSMOD ID OF USERMOD.    */
 /*
   +--------------------------------+
   !   USERMOD TO INSTALL HVTX002   !
   +--------------------------------+
 */ .
++VER(Z038)                 /* SREL FOR MVS.                    */
    FMID(HHVT300)           /* FUNCTION OF VERSION 3.0.0        */
    .
++SRC(HVTX002).
§§
//         DD  DISP=SHR,
//             DSN=#tgthlq.SHVTSAMP(HVTX002) <== TGT LIB HLQ
//SMPCNTL  DD *
SET BOUNDARY(GLOBAL).
 RECEIVE SELECT(HVTU002) SYSMODS .
SET BOUNDARY(#tzone) .
 APPLY SELECT(HVTU002) CHECK REDO .
 APPLY SELECT(HVTU002) REDO .
//
```

Figure 9-7 USERMOD creation for EXIT002

VTFM EXIT004

This exit is invoked after a virtual tape mount process is completed, and the virtual tape data set is allocated, opened and marked “in-use”. The mount process and this invoked exit are triggered by the first I/O sent to the VTFM virtual device when a new virtual tape is mounted. This I/O is usually sent by the application’s OPEN process.

The exit caller might invoke other services based on the virtual tape characteristics.

The sample EXIT004 is stored in the SAMPLIB library and the member name is HVTX004.

Note: At the time the application sends the first I/O to the virtual tape and the mount process is initiated, the application's OPEN process has not completed.

The exit does not get control if an application opens a virtual tape and this open does not result in a mount request (for example, the exit does not get control for virtual tapes that are passed to later steps by means of VOL=(,RETAIN) or DISP=(,PASS)).

Defining the VTFM virtual devices characteristics

Make sure that the User Unit Information Module (UIM) default definitions do not conflict with other UIMs or devices in your system.

If the supplied UIM defaults conflict with your installation, modify them using VTFM EXIT002.

For detailed information about EXIT002 and its installation, refer to *IBM VTF Mainframe Customization and Reference,* GC53-1190.

Copying the UIM modules to SYS1.NUCLEUS

Edit the HVTJCUIM member in the SAMPLIB library. It contains a job that copies the UIM modules to SYS1.NUCLEUS. Substitute the #-prefixed parameters as required, then submit the job.

The job should end with a return code of 0. In Figure 9-8, we show you the changes we have made to the HVTJCUIM member.

```
//COPYUIM  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN1      DD DISP=SHR,
//            DSN=HVT.SHVTLOAD              <== TGT LIB HLQ
//OUT1     DD DISP=SHR,
//            DSN=SYS1.NUCLEUS
//SYSIN    DD *
 COPY       INDD=((IN1,R)),OUTDD=OUT1
 S          M=((HVTEC000,CBDEC254))         <== UIM NUMBER
 S          M=((HVTUC000,CBDUC254))         <== UIM NUMBER
//
```

Figure 9-8 Modified HVTJCUIM

Defining the VTFM virtual devices to MVS using HCD

You should display the current active IODF data set used on your system. Figure 9-9 shows the MVS command we used in our system.

```
DISPLAY IOS,CONFIG
or
D IOS,CONFIG
```

Figure 9-9 Use of the DISPLAY IOS,CONFIG MVS command

Figure 9-10 shows you the result of the command.

```
RESPONSE=SC70
 IOS506I 17.43.53 I/O CONFIG DATA 561
 ACTIVE IODF DATA SET = SYS6.IODF18
 CONFIGURATION ID = TEST3287       EDT ID = 01
 TOKEN:  PROCESSOR DATE     TIME     DESCRIPTION
  SOURCE: SCZP101  09-07-30 16:51:01 SYS6     IODF18
 ACTIVE CSS:  1    SUBCHANNEL SETS CONFIGURED: 0
 CHANNEL MEASUREMENT BLOCK FACILITY IS ACTIVE
```

Figure 9-10 Display IOS result

Table 9-2 contains the values we used to illustrate the HCD process.

Table 9-2 Installation of 32 3490 virtual devices

Parameter	Value
Generic name	VT3490G2
Device unit	VT3490D2
Esoteric	VT3490ES
Number of devices	32
First device number	0570

Initializing the HCD environment

Perform the following steps:

1. Open the Hardware Configuration Definition (HCD) panels.
2. Save the current production I/O Definition File (option 6.2 on the main Hardware Configuration panel).
3. Select the work I/O Definition File in the I/O Definition File field (you may press PF4 on that field to get a list of the existing IODFs).

Entering the Hardware Configuration Definition (HCD)

Specify the name of the IODF to be used in the HCD primary option menu, as shown in Figure 9-11. Than select option 1 to define, modify, or view the configuration data.

```
                        z/OS V1.11 HCD
Command ===> ____________________________________________________________

                            Hardware Configuration

Select one of the following.

1_  0.  Edit profile options
    1.  Define, modify, or view configuration data
    2.  Activate or process configuration data
    3.  Print or compare configuration data
    4.  Create or view graphical configuration report
    5.  Migrate configuration data
    6.  Maintain I/O definition files
    7.  Query supported hardware and installed UIMs
    8.  Getting started with this dialog
    9.  What's new in this release

For options 1 to 5, specify the name of the IODF to be used.

I/O definition file . . . 'SYS6.IODF18'                          +
```

Figure 9-11 Hardware configuration primary panel

Defining the VTFM virtual devices esoteric name

Perform the following steps:

1. Select option 1. Operating system configurations in the Define, Modify, or View Configuration Data panel, as shown in Figure 9-12.

```
              Define, Modify, or View Configuration Data

     Select type of objects to define, modify, or view data.

      1_ 1. Operating system configurations
               consoles
               system-defined generics
               EDTs
                 esoterics
                 user-modified generics
           2. Switches
               ports
               switch configurations
                 port matrix
           3. Processors
               channel subsystems
                 partitions
                 channel paths
           4. Control units
           5. I/O devices

  F1=Help    F2=Split   F3=Exit    F9=Swap    F12=Cancel
```

Figure 9-12 Define, Modify, or View Configuration Data panel

2. Select the operating system you will change, as shown in Figure 9-13.

```
  Goto  Backup  Query  Help
--------------------------------------------------------------------------------
                 Operating System Configuration List              Row 1 of
8
Command ===> _____________________________________________ Scroll ===> PAGE

Select one or more operating system configurations, then press Enter. To
add, use F11.

/ Config. ID   Type     Description
_ ALLDEV       MVS      All devices
_ LO6RMVS1     MVS      Sysplex systems
_ MVSW1        MVS      Production systems
_ OPENMVS1     MVS      OpenEdition MVS
S TEST2094     MVS      Sysplex systems
_ TEST3287     MVS      Test 3287 devices
_ TRAINER      MVS      Trainer - Local Site Online
_ TRAINER2     MVS      Trainer - Remote Site Online
****************************** Bottom of data ******************************
```

Figure 9-13 Operating System Configuration List panel

3. Select the configuration ID in the EDT List panel you like to update. The EDT List panel displays, as shown in Figure 9-14.

```
                              EDT List
  Goto  Backup  Query  Help
------------------------------------------------------------------------
                                                                Row 1 of 1
Command ===> _____________________________________________ Scroll ===> PAGE

Select one or more EDTs, then press Enter. To add, use F11.

Configuration ID . : TEST2094     Sysplex systems

/ EDT Last Update By       Description
S 01  2009-06-02  HAIMO    ________________________________
*************************** Bottom of data ****************************
```

Figure 9-14 EDT List panel

4. In the Esoteric List panel, press the PF11 function key to add a new esoteric unit name. Figure 9-15 shows an example of the Esoteric List panel.

```
                            Esoteric List
  Goto  Filter  Backup  Query  Help
------------------------------------------------------------------------
                                                                Row 1 of 9
Command ===> _____________________________________ Scroll ===> PAGE

Select one or more esoterics, then press Enter. To add, use F11.

Configuration ID . : TEST2094     Sysplex systems
EDT identifier . . : 01

/ Esoteric  VIO    Token  State
_ ATL3      No     15
_ CART      No     16
_ NOVIO     No     9
_ SYSDA     No     10
_ SYSSQ     No     11
_ TAPE      No     12
_ VIO       Yes    13
  F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Reset
  F7=Backward  F8=Forward   F9=Swap      F10=Actions  F11=Add
F12=Cancel
```

Figure 9-15 Esoteric List panel

5. In the Add Esoteric panel, enter the esoteric name for the VTFM virtual devices, as shown in Figure 9-16.

```
                  Add Esoteric


 Specify the following values.

 Esoteric name  . . . VT3590__
 VIO eligible . . . . No     (Yes or No)
 Token  . . . . . . . ____


  F1=Help    F2=Split   F3=Exit    F5=Reset   F9=Swap
 F12=Cancel
```

Figure 9-16 Add Esoteric panel

In Figure 9-17, you see that “No device defined” is in the State field of this esoteric name. This esoteric name will be used later to group the VTFM virtual devices.

```
                      Esoteric List
  Goto  Filter  Backup  Query  Help
------------------------------------------------------------------------
                                                            Row 9 of 10
Command ===> _____________________________________ Scroll ===> PAGE

Select one or more esoterics, then press Enter. To add, use F11.

Configuration ID . : TEST2094     Sysplex systems
EDT identifier . . : 01

/ Esoteric  VIO    Token  State
_ VT3590    No     24
_ VT3592    No     ____   No device defined
*************************** Bottom of data ****************************

 F1=Help       F2=Split      F3=Exit       F4=Prompt     F5=Reset
 F7=Backward   F8=Forward    F9=Swap      F10=Actions   F11=Add
F12=Cancel
```

Figure 9-17 Esoteric List panel with message

6. Exit the Esoteric List panel to return to the Define, Modify, or View Configuration data panel, as shown in Figure 9-18. In the Define, Modify, or View Configuration Data panel, select option 5. I/O devices.

```
             Define, Modify, or View Configuration Data

    Select type of objects to define, modify, or view data.

     5_ 1. Operating system configurations
              consoles
              system-defined generics
              EDTs
                esoterics
                user-modified generics
        2. Switches
              ports
              switch configurations
                port matrix
        3. Processors
              channel subsystems
                partitions
                channel paths
        4. Control units
        5. I/O devices

  F1=Help    F2=Split   F3=Exit    F9=Swap   F12=Cancel
```

Figure 9-18 Define, Modify, or View Configuration Data panel: Specifying I/O devices

7. In the I/O Device List panel, shown in Figure 9-19, perform the following steps:
   a. Press PF11 to add new devices.
   b. In the Device Number field, enter the first device number for the VTFM virtual devices.
   c. In the Number Of Devices field, enter the number of devices that you want to add to the system.
   d. In the Device Type field, enter the device unit type.
   e. You may press PF4 in this panel to receive a list of the device types known to the system.

.

```
                            I/O Device List       Row 1 of 1060 More:
>
Command ===> ____________________________________________ Scroll ===> PAGE

Select one or more devices, then press Enter. To add, use F11.

  ----------Device------ --#--- --------Control Unit Numbers + --------
/ Number   Type +       CSS OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
_ 001A     9032             8  ____ ____ ____ ____ ____ ____ ____ ____
_ 001F     9032-5       7   8  001F ____ ____ ____ ____ ____ ____ ____
_ 0030     3287-2       2   1  E000 ____ ____ ____ ____ ____ ____ ____
_ 0032,2   3287-2       2   1  E000 ____ ____ ____ ____ ____ ____ ____
_ 0034     3287-1C      2   1  E000 ____ ____ ____ ____ ____ ____ ____
_ 0035     3287-2       2   1  E000 ____ ____ ____ ____ ____ ____ ____
_ 0066     2032         1   8  0066 ____ ____ ____ ____ ____ ____ ____
_ 0090     9032-5       1   8  0090 ____ ____ ____ ____ ____ ____ ____
_ 0091     9032-5       1   8  0091 ____ ____ ____ ____ ____ ____ ____
_ 0130     3287-1       2   1  E100 ____ ____ ____ ____ ____ ____ ____
_ 0132,2   3287-1       2   1  E100 ____ ____ ____ ____ ____ ____ ____
_ 0134     3287-2C      2   1  E100 ____ ____ ____ ____ ____ ____ ____
 F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Reset
F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F11=Add      F12=Cancel
```

Figure 9-19 I/O Device List

8. Specify the device number, the number of devices to be added, and select the requested VTFM device type from the list. Figure 9-20 shows the data that we used in our environment.

```
                              Add Device

Specify or revise the following values.

Device number  . . . . . . . . 0400  + (0000 - FFFF)
Number of devices  . . . . . . 32__
Device type  . . . . . . . . . VT3590D2_____  +

Serial number  . . . . . . . . __________
Description  . . . . . . . . . ________________________________

Volume serial number . . . . . ______  (for DASD)

Connected to CUs . . ____  ____  ____  ____  ____  ____  ____  ____  +


 F1=Help     F2=Split    F3=Exit     F4=Prompt   F5=Reset    F9=Swap
F12=Cancel
```

Figure 9-20 Add Device panel

9. In the Define devices to Operating System Configuration panel (Figure 9-21), select the systems you want the VTFM virtual devices to be connected to by typing S on the requested Config ID line.

```
          Define Device to Operating System Configuration
                                                              Row 1 of 8
 Command ===> ___________________________________________ Scroll ===> PAGE

 Select OSs to connect or disconnect devices, then press Enter.

 Device number  . : 0400          Number of devices  : 32
 Device type  . . : VT3590D2

 / Config. ID   Type     SS Description                      Defined
 _ ALLDEV       MVS        All devices
   LO6RMVS1     MVS        Sysplex systems
 _ MVSW1        MVS        Production systems
 _ OPENMVS1     MVS        OpenEdition MVS
 _ TEST2094     MVS        Sysplex systems
 S TEST3287     MVS        Test 3287 devices
 _ TRAINER      MVS        Trainer - Local Site Online
 _ TRAINER2     MVS        Trainer - Remote Site Online
 ************************** Bottom of data ***************************


  F1=Help       F2=Split      F3=Exit       F4=Prompt     F5=Reset
  F6=Previous   F7=Backward   F8=Forward    F9=Swap      F12=Cancel
```

Figure 9-21 Define Device to Operating System Configuration panel

10. The panel shown in Figure 9-22 is optional. Type YES in the LOCANY Parameter/Feature field if you want the virtual devices UCBs to reside above the line.

```
                    Define Device Parameters / Features
                                                                  Row 1 of 6
Command ===> ____________________________________________ Scroll ===> PAGE

Specify or revise the values below.

Configuration ID . : TEST3287     Test 3287 devices
Device number  . . : 0400         Number of devices  : 32
Device type  . . . : VT3590D2

Parameter/
Feature    Value +          R Description
OFFLINE    No                 DEVICE CONSIDERED ONLINE OR OFFLINE AT IPL
DYNAMIC    Yes                DEVICE SUPPORTS DYNAMIC CONFIGURATION
LOCANY     YES                UCB CAN RESIDE IN 31 BIT STORAGE
ALTCTRL    No                 SEPARATE PHYSICAL CONTROL UNIT PATH
DATACONV   No                 DATA CONVERSION
DUALDENS   No                 DUAL DENSITY
****************************** Bottom of data ******************************


 F1=Help       F2=Split      F3=Exit       F4=Prompt      F5=Reset
 F7=Backward   F8=Forward    F9=Swap      F12=Cancel
```

Figure 9-22 Define Device Parameters / Features panel

11. In the Assign/Unassign Device to Esoteric panel (Figure 9-23), assign the esoteric name you defined previously for the VTFM virtual devices by typing YES on the Assigned field of the requested esoteric name. Press Enter.

```
                    Assign/Unassign Device to Esoteric
                                                                  Row 10 of 16
Command ===> _____________________________________________ Scroll ===> PAGE

Specify Yes to assign or No to unassign.  To view devices already
assigned to esoteric, select and press Enter.

Configuration ID : TEST3287          Test 3287 devices
Device number  . : 0400              Number of devices  : 32
Device type  . . : VT3590D2          Generic  . . . . . : VT3590G2

/ EDT.Esoteric  Assigned  Starting Number Number of Devices
_ 01.NOVIO      No        ____           ____
_ 01.SYSDA      No        ____           ____
_ 01.SYSSQ      No        ____           ____
_ 01.TAPE       No        ____           ____
_ 01.TAPE3592   No        ____           ____
_ 01.VIO        No        ____           ____
_ 01.VT3590     YES       ____           ____
****************************** Bottom of data ******************************

 F1=Help       F2=Split      F3=Exit       F4=Prompt      F5=Reset
 F6=Previous   F7=Backward   F8=Forward    F9=Swap       F12=Cancel
```

Figure 9-23 Assign/Unassign Device to Esoteric panel

12. The optional I/O device list panel is displayed (Figure 9-24). It shows the new VTFM devices. Press Enter to return to the Define Device to Operating System Configuration panel.

```
                          I/O Device List      Row 21 of 1060 More:
>
Command ===> _____________________________________________ Scroll ===> PAGE

Select one or more devices, then press Enter. To add, use F11.

  ----------Device------ --#--- --------Control Unit Numbers + --------
/ Number   Type +        CSS OS 1--- 2--- 3--- 4--- 5--- 6--- 7--- 8---
_ 0135     3287-1        2   1  E100 ____ ____ ____ ____ ____ ____ ____
_ 01A0,32  CTC               1  ____ ____ ____ ____ ____ ____ ____ ____
_ 0200     3270-X            8  ____ ____ ____ ____ ____ ____ ____ ____
_ 02C0,31  OSA               8  02C0 ____ ____ ____ ____ ____ ____ ____
_ 02DF     OSAD              8  02C0 ____ ____ ____ ____ ____ ____ ____
_ 0400,32  VT3590D2          8  ____ ____ ____ ____ ____ ____ ____ ____
_ 0500,64  IQD               1  0500 ____ ____ ____ ____ ____ ____ ____
 F1=Help      F2=Split     F3=Exit      F4=Prompt    F5=Reset
F7=Backward
 F8=Forward   F9=Swap     F10=Actions  F11=Add      F12=Cancel
```

Figure 9-24 I/O Device List panel including the new defined devices

Creating a production IODF

Perform the following steps:

1. In the Hardware Configuration panel, select option 2. Activate or process configuration data, as shown in Figure 9-25.

```
                        z/OS V1.11 HCD
Command ===>
_______________________________________________________________________

                            Hardware Configuration

Select one of the following.

2_  0.  Edit profile options
    1.  Define, modify, or view configuration data
    2.  Activate or process configuration data
    3.  Print or compare configuration data
    4.  Create or view graphical configuration report
    5.  Migrate configuration data
    6.  Maintain I/O definition files
    7.  Query supported hardware and installed UIMs
    8.  Getting started with this dialog
    9.  What's new in this release

For options 1 to 5, specify the name of the IODF to be used.

I/O definition file . . . 'SYS1.IODF00.WORK'                        +
```

Figure 9-25 HCD configuration panel

2. In the Activate or Process Configuration Data panel, select option 1. Build Production I/O Definition File, as shown in Figure 9-26.

```
      Activate or Process Configuration Data

Select one of the following tasks.

1_  1.  Build production I/O definition file
    2.  Build IOCDS
    3.  Build IOCP input data set
    4.  Create JES3 initialization stream data
    5.  View active configuration
    6.  Activate or verify configuration
        dynamically
    7.  Activate configuration sysplex-wide
    8.  Activate switch configuration
    9.  Save switch configuration
    10. Build I/O configuration data
    11. Build and manage S/390 microprocessor
        IOCDSs and IPL attributes
    12. Build validated work I/O definition file

 F1=Help     F2=Split    F3=Exit     F9=Swap
F12=Cancel
```

Figure 9-26 HCD Activate or Process Configuration Data panel

Figure 9-27 shows you all the messages received from the build process.

```
                        Message List
  Save  Query  Help
------------------------------------------------------------------------------
                                                                Row 1 of 319
Command ===> _____________________________________________ Scroll ===> PAGE

Messages are sorted by severity. Select one or more, then press Enter.

/ Sev Msg. ID  Message Text
_ W   CBDG081I Following 3 operating system configurations of type MVS
#              have no console devices defined: ALLDEV, OPENMVS1,
#              TRAINER2
_ W   CBDG483I CFP channel path 0.01 of processor B706STP1 is not
#              connected. It should be connected to a channel path of
#              type CFP.
_ W   CBDG483I CFP channel path 0.01 of processor B706STP2 is not
#              connected. It should be connected to a channel path of
#              type CFP.
_ W   CBDG483I CBP channel path 0.C5 of processor SCZP101 is not
#              connected. It should be connected to a channel path of
#              type CBP.
 F1=Help       F2=Split      F3=Exit       F4=Prompt     F5=Reset
 F7=Backward   F8=Forward    F9=Swap      F10=Actions   F12=Cancel
```

Figure 9-27 IODEF message list

3. Specify the name of the new IODF and the volume serial on which your IODF files reside. Figure 9-28 shows an example.

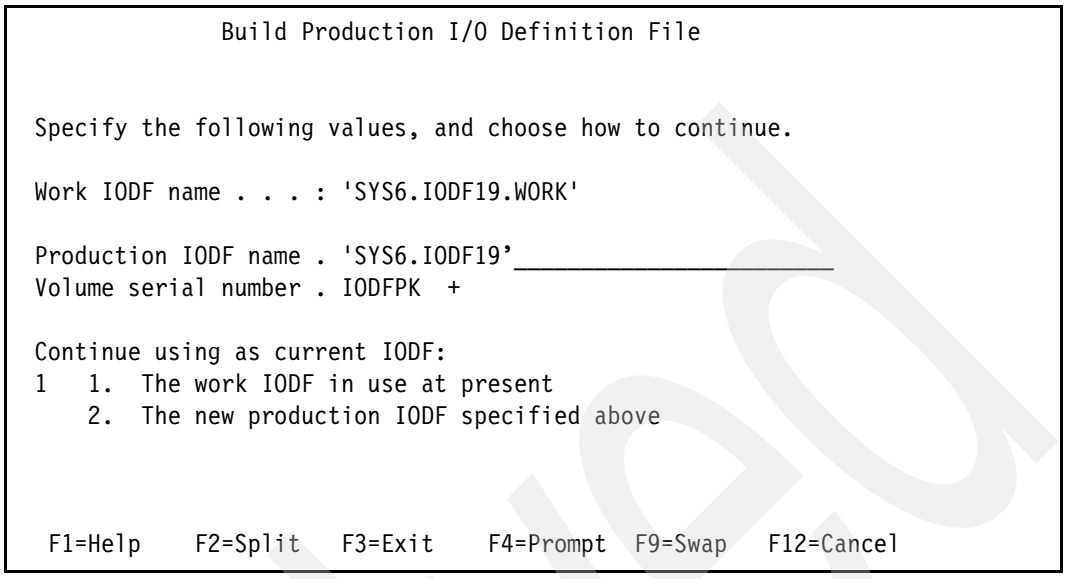


Figure 9-28 Build Production I/O Definition File panel

Figure 9-29 shows you the Define Descriptor Fields panel. You can overwrite the values of Descriptor field 1 and Descriptor field 2.

```
                    Define Descriptor Fields


Specify or revise the following values.

Production IODF name  . : 'SYS6.IODF19'

Descriptor field 1  . . . SYS6
Descriptor field 2  . . . IODF19
 F1=Help      F2=Split     F3=Exit      F5=Reset     F9=Swap
F12=Cancel
```

Figure 9-29 Define Descriptor Fields panel

Figure 9-30 shows you the message that a new production IODF has been successfully created.

```
 Production IODF SYS6.IODF19 created.
```

Figure 9-30 IODF success message

After the IODF creation (Figure 9-28 on page 283 to Figure 9-30), the new configuration should be activated. Select Option 7. Activate configuration sysplex-wide in the Activate or Process Configuration Data panel, as shown in Figure 9-31.

```
       Activate or Process Configuration Data

 Select one of the following tasks.

 7_  1.  Build production I/O definition file
     2.  Build IOCDS
     3.  Build IOCP input data set
     4.  Create JES3 initialization stream data
     5.  View active configuration
     6.  Activate or verify configuration
         dynamically
     7.  Activate configuration sysplex-wide
     8.  Activate switch configuration
     9.  Save switch configuration
     10. Build I/O configuration data
     11. Build and manage S/390 microprocessor
         IOCDSs and IPL attributes
     12. Build validated work I/O definition file

  F1=Help     F2=Split    F3=Exit     F9=Swap
 F12=Cancel
```

Figure 9-31 HCD Activate or Process Configuration Data panel

9.6.2 Security definitions in a RACF environment

This step provides detailed explanations and sample definitions to allow RACF to manage the security aspects of VTFM. We used the HVTJRACF member from the SAMPLIB library.

Perform the following steps:

1. Add five new RACF groups to manage the VTFM product profiles, as shown in Figure 9-32.

```
   PROF NOPREF
/*  -- DEFINE A RACF GROUP TO MANAGE THE PRODUCT PROFILES       --  */
   AG (VTFMGRP) OWNER(HAIMO) SUPGROUP(SYS1)
/*  -- DEFINE A RACF GROUP TO BE ASSIGNED TO THE VDB PROFILE    --  */
   AG (VTFMVDB) OWNER(VTFMGRP) SUPGROUP(VTFMGRP)
/*  -- DEFINE A RACF GROUP TO BE ASSIGNED TO THE JOURNAL PROFILE -  */
   AG (VTFMJRN) OWNER(VTFMGRP) SUPGROUP(VTFMGRP)
/*  -- DEFINE A RACF GROUP TO BE ASSIGNED TO THE HISTORY PROFILE -  */
   AG (VTFMHST) OWNER(VTFMGRP) SUPGROUP(VTFMGRP)
/*  -- DEFINE A RACF GROUP TO BE ASSIGNED TO THE VIRT TAPE DS PROF. */
/*  -- IT WILL BE USED TO ADMINISTRATE THE VIRTUAL TAPE data setS,   */
/*  -- THE VDB AND THE JOURNAL FILE.                                */
   AG (VTFMTAPE) OWNER(VTFMGRP) SUPGROUP(VTFMGRP) OMVS(AUTOGID)
```

Figure 9-32 Define RACF groups

Where:

VTFMGRP	The superior group for all other RACF groups that we define for VTFM.
VTFMVDB	This RACF group is used to protect the Volume Database (VDB).
VTFMJRN	This RACF group is used to protect the journal file.
VTFMHST	This RACF group is used to protect the history file.
VTFMTAPE	This RACF group is used to administrate the virtual tape data sets.

2. Figure 9-33 shows how you can add a RACF user, define a RACF profile for the VTFM started task, and make the started task a trusted started task.

```
   PROF NOPREF
/*  -- DEFINE A RACF USER TO BE ASSIGNED TO THE STARTED TASK      --*/
   AU (VTFM) OWNER(VTFMGRP) DFLTGRP(VTFMTAPE)                    -
      NOPASSWORD NOOIDCARD                                       -
      OMVS(AUTOUID  HOME('/') PROGRAM('/bin/sh'))
/*  -- DEFINE A RACF PROFILE TO THE VTFM STARTED TASK           --  */
   RDEF STARTED (VTFM*.**) STDATA(USER(VTFM)                     -
      GROUP(VTFMTAPE)) OWNER(VTFMGRP)
/*  -- DEFINE THE STARTED TASK AS A TRUSTED STARTED TASK.        --  */
   RALT STARTED (VTFM*.**) STDATA(TRUSTED(YES))
```

Figure 9-33 Add a user and define the started task

Where:

PRIVILEGED RACF allows you to specify that a started procedure is privileged, which means that most authorization requests done for the procedure are considered to be successful, without actually performing any checking. This includes bypassing the checks for security classification on users and data.

TRUSTED RACF allows you to specify that a started procedure is trusted, which means that most authorization requests done for the procedure are considered successful, without actually performing any checking. This includes bypassing the checks for security classification on users and data.

Note: The TRUSTED attribute is similar the PRIVILEGED attribute, but auditing is done. Refer to *z/OS Security Server RACF System Programmer's Guide,* SA22-7681 for more information.

3. Define five data set profiles to protect the virtual tape data sets, the VDB, and the journal file, as shown in Figure 9-34.

```
   PROF NOPREF
/*  -- DEFINE A RACF PROFILE TO VIRTUAL TAPE data setS          --  */
   AD ('VTFMTAPE.**') GEN OWNER(VTFMGRP) UACC(NONE)
/*  -- DEFINE A RACF PROFILE TO THE VDB                        --  */
  AD ('VTFMVDB.DATABASE') GEN OWNER(VTFMGRP) UACC(NONE)
/*  -- DEFINE A RACF PROFILE TO THE JOURNAL FILE               --  */
   AD ('VTFMJRN.JOURNAL') GEN OWNER(VTFMGRP) UACC(NONE)
/*  -- DEFINE A RACF PROFILE TO THE HISTORY FILE                --  */
   AD ('VTFMHST.HISTORY') GEN OWNER(VTFMGRP) UACC(NONE)
/*  -- DEFINE A RACF PROFILE TO THE HISTORY BACKUP FILE         --  */
   AD ('VTFMHST.HISTORY.BKP') GEN OWNER(VTFMGRP) UACC(NONE)
```

Figure 9-34 Add data set profiles to protect the VTFM files

4. Grant the VTFM started task an ALTER authorization to the virtual tape data sets, the VDB, and the journal file. Figure 9-35 shows an example.

```
   PROF NOPREF
/*  -- GRANT THE VTFM STC AN ALTER AUTH TO THE VIRTUAL TAPE DS  --  */
   PE 'VTFMTAPE.**' ID(VTFMTAPE) ACC(ALTER)
/*  -- GRANT THE VTFM STC AN ALTER AUTH TO THE VDB              --  */
   PE 'VTFMVDB.DATABASE' ID(VTFMTAPE) ACC(ALTER) GEN
/*  -- GRANT THE VTFM STC AN ALTER AUTH TO THE JOUNRAL FILE     --  */
   PE 'VTFMJRN.JOURNAL' ID(VTFMTAPE) ACC(ALTER) GEN
/*  -- GRANT THE VTFM STC AN ALTER AUTH TO THE HISTORY FILE     --  */
   PE 'VTFMHST.HISTORY' ID(VTFMTAPE) ACC(ALTER) GEN
/*  -- GRANT THE VTFM STC AN ALTER AUTH TO THE HISTORY BKP FILE --  */
   PE 'VTFMHST.HISTORY.BKP' ID(VTFMTAPE) ACC(ALTER) GEN
```

Figure 9-35 Grants VTFM authorization

5. If the stated task VTFM has either the PRIVILEGED or TRUSTED attribute, you can skip this step.

   Grant VTFM authorization to issue MODIFY operator commands, authorization to issue VARY commands to its devices, and authorization to issue MONITOR commands. Figure 9-37 shows you examples of how you can grant the VTFM started task to use operator commands.

```
/* RDEF OPERCMDS (MVS.MODIFY.STC.#optproc.#optproc) UACC(NONE)      */
/* PE MVS.MODIFY.STC.#optproc.#optproc CLASS(OPERCMDS) -            */
/*     ID(#vdsnq1) ACC(CONTROL)                                     */
/*                                                                  */
/*  -- GRANT THE VTFM STC THE AUTH TO ISSUE START COMMANDS      --  */
/*                                                                  */
/* RDEF OPERCMDS (MVS.START.STC.#optproc.V*) UACC(NONE)             */
/* PE MVS.START.STC.#optproc.V* CLASS(OPERCMDS) -                   */
/*     ID(#vdsnq1) ACC(CONTROL)                                     */
/*                                                                  */
/*  -- GRANT THE VTFM STC THE AUTH TO ISSUE VARY COMMANDS       --  */
/*                                                                  */
/* PE MVS.VARY.DEV CLASS(OPERCMDS) ID(#vdsnq1) ACCESS(CONTROL)      */
/*                                                                  */
/*  -- GRANT THE VTFM STC THE AUTH TO ISSUE MONITOR COMMANDS    --  */
/*                                                                  */
/* PE MVS.MONITOR CLASS(OPERCMDS) ID(#vdsnq1) ACCESS(CONTROL)       */
/*                                                                  */
/*  -- GRANT THE VTFM STC THE AUTH TO ISSUE SETCON COMMANDS     --  */
/*                                                                  */
/* PE MVS.SETCON.TRACKING CLASS(OPERCMDS) ID(#vdsnq1) ACCESS(CONTROL)/
/*                                                                  */
/* --  GRANT THE VTFM STC THE AUTH TO ISSUE JES3 VARY COMMANDS  --  */
/*                                                                  */
/* PE JES3.VARY.DEV CLASS(OPERCMDS) ID(#vdsnq1) ACCESS(CONTROL)     */
```

Figure 9-36 Grant VTFM authorization to issue MODIFY operator commands

6. Refresh the RACF in-storage information, as shown in Figure 9-37.

```
/*                                                                  */
/*  -- REFRESH RACF'S STARTED AND OPERCMDS CLASSES              --  */
/*                                                                  */
  SETR RACLIST(STARTED ) REFRESH
  SETR RACLIST(OPERCMDS) REFRESH
/*                                                                  */
```

Figure 9-37 Refresh in-storage information

"ERASE-ON-SCRATCH" option

Check if the "ERASE-ON-SCRATCH" option is active in your RACF configuration by issuing the command shown in Figure 9-38.

```
SETR LIST
```

Figure 9-38 List RACF settings

Figure 9-39 shows the current active RACF settings.

```
....
SECURITY RETENTION PERIOD IN EFFECT IS  9999 DAYS.
ERASE-ON-SCRATCH IS INACTIVE
SINGLE LEVEL NAME PREFIX IS PASSWORD
...
```

Figure 9-39 RACF settings

If ERASE-ON-SCRATCH IS INACTIVE is displayed, no additional actions are required. If it is active, make sure it is not applied to the profile protecting the virtual tape data sets on disk.

If the above instruction is not observed, each virtual tape data set associated with a virtual tape being scratched by the HVTUSCR utility is overwritten by data prior to its deletion. This, in turn, causes a high and undesirable impact to the HVTUSCR utility.

9.6.3 Preparing your DISK volumes

One 3390 disk volume is the minimum requirement to store the virtual tape volumes. As you can see in Figure 9-40, we have prepared two 3390-27 volumes to retain approximate 50 3592 volumes with a capacity of 1.000 MB.

```
//INIT     EXEC PGM=ICKDSF,PARM='NOREPLYU'
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
 INIT UNIT(6209) VOLID(SBOXOQ) VFY(NW6209) -
      PURGE VTOC(0007,0,480) INDEX(0001,0,90) SG
 INIT UNIT(6309) VOLID(SBOXOR) VFY(NW6309) -
      PURGE VTOC(0007,0,480) INDEX(0001,0,90) SG
```

Figure 9-40 Prepare DISK volumes

Virtual tape data sets DISK storage usage

This section describes the DASD storage requirements:

Virtual tape data sets

Each “active” virtual tape volume is represented by a multi-volume disk data set whose size is the size of the data contained in the virtual tape. Scratch tapes are not represented by disk data sets.

VDB

The VDB contains a record for each 'active' or scratch virtual tape. Its size is 256 bytes. This file size should be calculated according to the number of virtual tapes. For example, if the VDB contains 100,000 virtual tapes, its size would be 500 tracks.

Journal file

The journal file contains:

- One control record.
- Two records for each mounted virtual tape.
- One record for each virtual tape status change done any VTFM utility.

A journal record size is 384 bytes and 16 records are blocked in one block.

The file size is calculated according to the above factors and the number of days that information should be kept in the journal file.

For example, if there are 600 mounts per day, 600 virtual tapes are handled by offline utilities, and the VDB is backed up every 2 days, the journal file size would be 45 tracks.

History file (optional)

The history file contains:

- One control record.
- One data set record for each generic name.
- One record for each user pattern.

A data set record uses 6 bytes for each observation and another 60 bytes for fixed data.

Assuming virtual tapes are used on a weekly basis, each data set record uses 372 (60*52*6) bytes.

If 500,000 different data set records are to be handled, the history file size would be 3400 tracks.

History backup file (optional)

The history file is backed up by the history backup file. Its size is similar to the history file size.

9.6.4 Defining a user catalog to contain the virtual tape data sets

Perform the following steps:

1. Define an ICF user catalog for the virtual tape data set entries, as shown in Figure 9-41.

```
//STEP0001 EXEC PGM=IDCAMS
//FILE     DD DISP=SHR,UNIT=3390,VOL=SER=SBOX0P
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DEFINE -
      USERCATALOG                     -
            (NAME (UCAT.VTFM) -
            VOLUMES(SBOX0P)    -
            ICFCATALOG                -
            STRNO(8)                  -
            CYLINDERS(100 50)         -
            SHAREOPTIONS(3 4))        -
      DATA(BUFND(8)                   -
            NAME (UCAT.VTFM.DATA) -
            CYLINDERS(100 50)         -
            CONTROLINTERVALSIZE(4096) -
            FREESPACE(20 20)          -
            RECORDSIZE(100 32400))    -
      INDEX(BUFNI(7)                  -
            NAME (UCAT.VTFM.INDEX) -
            TRACKS(300 50)            -
            CONTROLINTERVALSIZE(4096) -
            IMBED                     -
            REPLICATE)
/*
```

Figure 9-41 Define an ICF user catalog

2. Define a second user catalog for all other VTFM related data sets, as shown in Figure 9-42.

```
//STEP0002 EXEC PGM=IDCAMS
//FILE     DD DISP=SHR,UNIT=3390,VOL=SER=SBOX0P
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
    DEFINE -
      USERCATALOG                     -
            (NAME (UCAT.VTFMTAPE) -
            VOLUMES(SBOX0P)    -
            ICFCATALOG                -
            STRNO(8)                  -
            CYLINDERS(100 50)         -
            SHAREOPTIONS(3 4))        -
      DATA(BUFND(8)                   -
            NAME (UCAT.VTFMTAPE.DATA) -
            CYLINDERS(100 50)         -
            CONTROLINTERVALSIZE(4096) -
            FREESPACE(20 20)          -
            RECORDSIZE(100 32400))    -
      INDEX(BUFNI(7)                  -
            NAME (UCAT.VTFMTAPE.INDEX) -
            TRACKS(300 50)            -
            CONTROLINTERVALSIZE(4096) -
            IMBED                     -
            REPLICATE)
/*
```

Figure 9-42 Define an ICF user catalog or all other VTFM related data sets

3. Optionally, connect the previously defined ICF user catalogs to other master catalogs when you are sharing the VTFM between multiple systems. Figure 9-43 shows you an example.

```
//CONNECT  EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 IMPORT CONNECT -
   OBJECTS((UCAT.VTFM  DEVT(3390) VOL(SBOX0P)))
 IMPORT CONNECT -
   OBJECTS((UCAT.VTFMTAPE  DEVT(3390) VOL(SBOX0P)))
```

Figure 9-43 Connect a user catalog

Important: VTFMTAPE specifies the high level qualifier to be assigned to the virtual tape data set name. This data set name high level qualifier should be unique throughout the system. No other system component should allocate and catalog data sets having this value as their data set's name high level qualifier. If this condition is not observed, the VTFM recovery procedure is not able to properly locate the cataloged virtual tape data sets and is not able to recover any VTFM virtual tape.

4. Define one or more ALIASes to fit you naming conventions, as shown in Figure 9-44.

```
// EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  DEFINE ALIAS   (NAME('VTFM') RELATE('UCAT.VTFM')) +
    CATALOG('MCAT.SANDBOX.Z1A.SBOX00')
  DEFINE ALIAS   (NAME('VTFM') RELATE('UCAT.VTFM')) +
    CATALOG('MCAT.SANDBOX.Z1B.SBOX00')
  DEFINE ALIAS   (NAME('VTFMHST') RELATE('UCAT.VTFM')) +
    CATALOG('MCAT.SANDBOX.Z1A.SBOX00')
  DEFINE ALIAS   (NAME('VTFMHST') RELATE('UCAT.VTFM')) +
    CATALOG('MCAT.SANDBOX.Z1B.SBOX00')
  DEFINE ALIAS   (NAME('VTFMJRN') RELATE('UCAT.VTFM')) +
    CATALOG('MCAT.SANDBOX.Z1A.SBOX00')
  DEFINE ALIAS   (NAME('VTFMJRN') RELATE('UCAT.VTFM')) +
    CATALOG('MCAT.SANDBOX.Z1B.SBOX00')
  DEFINE ALIAS   (NAME('VTFMVDB') RELATE('UCAT.VTFM')) +
    CATALOG('MCAT.SANDBOX.Z1A.SBOX00')
  DEFINE ALIAS   (NAME('VTFMVDB') RELATE('UCAT.VTFM')) +
    CATALOG('MCAT.SANDBOX.Z1B.SBOX00')
  DEFINE ALIAS   (NAME('VTFMTAPE') RELATE('UCAT.VTFMTAPE')) +
    CATALOG('MCAT.SANDBOX.Z1A.SBOX00')
  DEFINE ALIAS   (NAME('VTFMTAPE') RELATE('UCAT.VTFMTAPE')) +
    CATALOG('MCAT.SANDBOX.Z1B.SBOX00')
/*
```

Figure 9-44 Sample job to define one or more ALIAS

9.6.5 Creating a VTFM started procedure

Perform the following steps:

1. Use the sample JCL shown in Figure 9-45 to copy the VTFM procedure to the system PROCLIB library.
2. Edit the HVTJCPRC member in the SAMPLIB library. Substitute the #-prefixed parameters as required, and then submit the job. The job should end with a return code of 0.

```
//COPYPROC EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN1      DD DISP=SHR,
//            DSN=HVT.SHVTSAMP            <== TGT LIB HLQ
//OUT1     DD DISP=SHR,
//            DSN=SYS1.PROCLIB            <== SYSTEM PROCLIB
//SYSIN    DD *
 COPY       INDD=((IN1,R)),OUTDD=OUT1
 S          M=((HVTPROC,VTFMSRV,R))        <== PROCEDURE NAME
//
```

Figure 9-45 Sample JCL to copy HVTPROC member

After you have copied the HVTPROC member to your PROCLIB and renamed it, make the following changes to the VTFMSRV member that contains the VTFM procedure:

1. Specify a procedure name of VTFMSRV.
2. Specify a step name of VTVM.
3. Change the TIME parameter from 1440 to NOLIMIT.
4. Change the data set name of the STEPLIB DD to HVT.SHVTLOAD.
5. Change the data set name of the HVTSIN DD to VTFM.JCL.
6. Change the data set name of the HVTVDB DD to VTFMVDB.DATABASE.
7. Change the data set name of the HVTJRN DD to VTFMVDB.JOURNAL.

In Figure 9-46, we have highlighted the changes we made to the VTFMSRV member for our lab environment.

```
//VTFMSRV   PROC SERVICE=SRV,                    <== PROCEDURE NAME
//            SYS=00
//VTFM      EXEC PGM=HVT&SERVICE,                <== STEP NAME
//            PARM='SYS=&SYS',
//            TIME=NOLIMIT,
//            REGION=0M
//*
//STEPLIB  DD DISP=SHR,DSN=HVT.SHVTLOAD          <== TGT LIB HLQ
//*----------------------- SYSOUT data setS -------------------------
//HVTLIST  DD SYSOUT=*,OUTLIM=16777215    LIST INITIALIZATION STATS
//HVTSOUT  DD SYSOUT=*,OUTLIM=16777215    SERVER MESSAGES
//HVTTRC   DD SYSOUT=*,OUTLIM=16777215    INTERNAL TRACE (TRCON)
//HVTGTF   DD SYSOUT=*,OUTLIM=16777215    VIRTUAL GTF TRACE (GTFON)
//SYSTSPRT DD SYSOUT=*,OUTLIM=16777215    DICT BUILD FOR TYPE3 COMPRESS
//SYSABEND DD SYSOUT=*,OUTLIM=16777215
//SYSPRINT DD SYSOUT=*,OUTLIM=16777215
//*---------------------- INITIALIZATION STATEMENTS ----------------
//HVTSIN   DD DISP=SHR,DSN=VTFM.JCL              <== TGT LIB HLQ
//HVTUTIL  DD DSN=NULLFILE
//*----------------------- DATABASE data setS ------------------------
//HVTVDB   DD DISP=SHR,DSN=VTFMVDB.DATABASE
//HVTJRN   DD DISP=SHR,DSN=VTFMJRN.JOURNAL
//*----------------------- SORT data setS ----------------------------
//SORTWK01 DD UNIT=VIO,SPACE=(TRK,(150,150)),DISP=(,DELETE)
//SORTWK02 DD UNIT=VIO,SPACE=(TRK,(150,150)),DISP=(,DELETE)
//SORTIN   DD UNIT=VIO,SPACE=(TRK,(180,180)),DISP=(,DELETE)
//SORTOUT  DD UNIT=VIO,SPACE=(TRK,(180,180)),DISP=(,DELETE)
//SYSOUT   DD SYSOUT=*
//*----------------------- WORK data setS ----------------------------
//SYSUT1   DD UNIT=VIO,SPACE=(TRK,(180,180)),DISP=(,DELETE)
//SYSUT2   DD UNIT=VIO,SPACE=(TRK,(180,180)),DISP=(,DELETE)
//*----------------------- PROSMS, STOPX37, ACC ---------------------
//PROIGN   DD DUMMY
//ACCIGN   DD DUMMY
```

Figure 9-46 VTMSRV JCL

9.6.6 APF authorizing the VTFM load library

Perform the actions in this section to APF authorize the VTFM load library.

Dynamically adding the VTFM load library to the APF list

Issue the STEPROG operator command from the system console in order to APF authorize the VTFM library, as shown in Figure 9-47.

```
SETPROG APF,ADD,DSNAME=steplib,VOLUME=prodvol

SETPROG APF,ADD,DSNAME=HVT.SHVTLOAD,VOLUME=Z1BRB2
```

Figure 9-47 APF authorize the VTFM library

Where:

steplib The name of the VTFM load library

prodvol The disk volume serial number where the VTFM load library resides

Adding the VTFM load library to the APF list

Enter the appropriate PROGxx member in the SYS1.PARMLIB library, as shown in Figure 9-48, and add the VTFM load library to the APF list.

```
APF FORMAT(DYNAMIC)
/*                                                              */
/* PLEASE KEEP THIS LIST IN ALPHABETIC ORDER                    */
....
APF ADD
    DSNAME(HAA.V2R2M0.SHAALOAD)                    VOLUME(SBOX91)
APF ADD
    DSNAME(HAIM0.LOAD)                             VOLUME(SBOX81)
APF ADD
    DSNAME(HVT.SHVTLOAD)                           VOLUME(&SYSR2)
APF ADD
    DSNAME(HES.SHESLMOD)                           VOLUME(******)
APF ADD
    DSNAME(IDI.SIDIAUTH)                           VOLUME(******)
APF ADD
    DSNAME(IDI.SIDIAUTH)                           VOLUME(&SYSR2)
....
```

Figure 9-48 Add the VTFM load library to the APF list

Displaying the APF list

You can check that the VTFM load library is in the APF list by using the MVS display PROG command.

To display all libraries in the APF list, you can use either DISPLAY PROG,APF or DISPLAY PROG,APF,ALL, as shown in Figure 9-49.

```
   D PROG,APF,ALL
```

Figure 9-49 MVS display PROG command to list all APF authorized libraries

In Figure 9-50, you can see a list of all APF authorized load libraries.

```
RESPONSE=SC70
 CSV450I 18.05.21 PROG,APF DISPLAY 410
 FORMAT=DYNAMIC
 ENTRY VOLUME DSNAME
    1  Z1BRB1 SYS1.LINKLIB
    2  Z1BRB1 SYS1.SVCLIB
    3  SBOX11 $DSN810.SDSNLINK
...
  318  SBOX81 HAIM0.LOAD
  319  Z1BRB2 HVT.SHVTLOAD
  320  Z1BRB1 HES.SHESLMOD
...
  535  CCC001 ZSK060.BDG.TEST0.LOAD
  536  TOTTS5 VAINI.DGA.LOAD
```

Figure 9-50 List of all APF authorized libraries

To display all entries for a specific library name, use the DISPLAY PROG,APF, DSNAME command, as shown in Figure 9-51.

```
   D PROG,APF,DSNAME=HVT.SHVTLOAD
```

Figure 9-51 MVS display PROG command to list only one APF authorized library

When you have specified DSNAME in the DISPLAY PROG command, you get a list of only the libraries that match the data set name you have specified, as shown in Figure 9-52.

```
RESPONSE=SC70
 CSV450I 18.05.21 PROG,APF DISPLAY 410
 FORMAT=DYNAMIC
 ENTRY VOLUME DSNAME
    1  Z1BRB2 HVT.SHVTLOAD
```

Figure 9-52 List of a specific APF authorized library

9.6.7 Copying the UIM modules to SYS1.NUCLEUS

Perform the following steps to copy the UIM modules to SYS1.NUCLEUS:

1. The HVTJCUIM member in the SAMPLIB library contains a job that copies the UIM modules to SYS1.NUCLEUS. Edit the job as required based on your environment. Figure 9-53 shows the changes we made to the HVTJCUIM member for our environment.

```
//COPYUIM  EXEC PGM=IEBCOPY
//SYSPRINT DD SYSOUT=*
//IN1      DD DISP=SHR,
//            DSN=HVT.SHVTLOAD              <== TGT LIB HLQ
//OUT1     DD DISP=SHR,
//            DSN=SYS1.NUCLEUS
//SYSIN    DD *
 COPY      INDD=((IN1,R)),OUTDD=OUT1
 S         M=((HVTEC000,CBDEC254))         <== UIM NUMBER
 S         M=((HVTUC000,CBDUC254))         <== UIM NUMBER
//
```

Figure 9-53 Modified HVTJCUIM

2. Substitute the #-prefixed parameters as required, and then submit the job. The job should end with a return code of 0.

9.6.8 Defining the VTFM subsystem

These steps are optional. If the subsystem is not predefined, VTFM dynamically defines it during its initialization.

Define the subsystem name in the appropriate IEFSSNxx member in SYS1.PARMLIB. The subsystem name should be identical to the SSNAME parameter specified in the HVTOPTxx member.

Adding the VTFM subsystem to IEFSSNnn

Set the VTFM subsystem enabled in IEFSSNxx, as shown in Figure 9-54.

```
SUBSYS SUBNAME(SMS)            /* SYSTEM MANAGED STORAGE             */
  INITRTN(IGDSSIIN)
  INITPARM('ID=00,PROMPT=DISPLAY')
SUBSYS SUBNAME(JES2)           /* JES2 IS THE PRIMARY SUBSYSTEM NAME */
  PRIMARY(YES)  START(YES)
/* SUBSYS SUBNAME(JES3) */   /* COMMENTED: JES3 AS PRIMARY SUBSYSTEM */
/*  PRIMARY(YES)  START(YES) */
SUBSYS SUBNAME(OAM1)           /* OAM FOR DFSMS                      */
  INITRTN(CBRINIT)
SUBSYS SUBNAME(DFRM)           /* DFSMSRMM                           */
  INITRTN(EDGSSSI)
SUBSYS SUBNAME(VT01)           /* IBM System Storage VTFM            */
....
```

Figure 9-54 Add the VTFM subsystem to IEFSSNnn

Dynamically adding the VTFM subsystem

You can use the MVS system SETSSI command, as shown in Figure 9-55, to define the VT01 subsystem dynamically. You can issue the SETSSI command from one of the following locations:

- A console that has master authority
- A console to which an operator with sufficient RACF authority has logged on

```
SETSSI ADD,SUBNAME=VT01
```

Figure 9-55 MVS command SETSSI

Where:

ADD Specifies that a subsystem is to be dynamically added.

SUBNAME Specifies the subsystem name to be dynamically added.

You can check that the subsystem was successfully added by using the MVS system command shown in Figure 9-56.

```
DISPLAY SSI
```

Figure 9-56 MVS command display SSI

Figure 9-57 shows the output of the command.

```
DISPLAY SSI
IEFJ100I  18.31.01  SSI DISPLAY 493
SUBSYS=JES2 (PRIMARY)
   DYNAMIC=YES     STATUS=ACTIVE       COMMANDS=REJECT
...
SUBSYS=VT01
   DYNAMIC=NO      STATUS=ACTIVE       COMMANDS=N/A
```

Figure 9-57 Display SSI result

9.6.9 Global Resource Serialization considerations (optional)

The disks on which the VDB and the journal file reside are subject to frequent RESERVEs. This might block other systems that share the same disks from accessing them.

The resource names are as follows:

QNAME V_TAPE

RNAME The name of the VDB or journal file, respectively, padded with blanks. Its total length is 44 characters.

In addition, you might have contentions (or even deadlocks) on the SYSIGGV2 QNAME.

If a Global Resource Serialization product is employed, it is necessary to convert these RESERVEs, For example, GRS customers should:

- Not place the resources in the SYSTEMS Exclusion RNL
- Place the resources in the RESERVE Conversion RNL

Figure 9-58 shows you the correct use of the GRSRNLxx definitions for VTFM.

```
RNLDEF RNL(CON) TYPE(GENERIC)
       QNAME('V_TAPE')
RNLDEF RNL(CON) TYPE(GENERIC)
       QNAME(SYSIGGV2)
```

Figure 9-58 GRS GRSRNLxx sample

9.6.10 Allowing VTFM user SMF record writing (optional)

This step is optional and should be carried out only if the VTFM user SMF record will be written (that is, the OPTIONS SMF=ENABLE parameter is specified in the HVTOPTxx member and the VTFM user SMF record number is not 0).

IEFU83 and IEFU84 SMF exits

If IEFU83 or IEFU84 SMF exits are active, make sure they do not suppress the write of the VTFM user SMF record.

SMF record number

The VTFM user SMF record number should be included in the appropriate SYS or SUBSYS statement in the SMPPRMxx member.

When finished, the SMFPRMxx member should be refreshed by using the operator command, as shown in Figure 9-59.

```
SET SMF=xx
```

Figure 9-59 Refresh SMS settings

Where:

xx The SMFPRMxx member suffix

9.6.11 Performance considerations (optional)

Users that run in WLM compatibility mode should assign the VTFM started task a PGN number of a performance group assigned to batch jobs that use tapes.

Users that run in WLM goal mode should assign the VTFM started task a Service Class assigned to batch jobs that use tapes.

When these two actions are performed, other applications will compete fairly with the VTFM started task over the available system resources.

9.6.12 SMS ACS routines considerations

This section provides details about what SMS ACS routines modifications should be made in order to let VTFM function properly in SMS-managed and non-SMS-managed environments.

Refer to *IBM VTF Mainframe Customization and Reference,* GC53-1190 for more information.

Important: Ensure that a data class is not assigned to virtual tape data sets.

Assuming that the virtual tape data sets high level qualifier is VTFMTAPE, code the statements shown in Figure 9-60 at the very beginning of your DATACLAS ACS routine.

```
PROC DATACLAS
...
SELECT
  WHEN (&HLQ= 'VTFMTAPE')
    DO
      SET &DATACLAS = ''
      EXIT
    END
```

Figure 9-60 SMS data class sample

Important: Ensure that the management class set by VTFM is not modified.

Assuming the virtual tape data sets high level qualifier is VTFMTAPE, code the statements shown in Figure 9-61 at the very beginning of your MGMTCLAS ACS routine.

```
PROC MGMTCLAS
...
    IF (&HLQ = 'VTFMTAPE') THEN DO
       EXIT
    END
```

Figure 9-61 SMS management class sample

Important: Ensure that the storage class set by VTFM is not modified.

Assuming the virtual tape data sets high level qualifier is VTFMTAPE, code the statements shown in Figure 9-62 at the very beginning of your STORCLAS ACS routine.

```
PROC STORCLAS
...
    IF (&HLQ = 'VTFMTAPE') THEN DO
       EXIT
    END
```

Figure 9-62 SMS storage class sample

If at least one diskpool is SMS-managed, you should relate the Storage Group that contains the SMS-managed disks dedicated to VTFM to the storage class assigned to the virtual tape data set, as shown in Figure 9-63.

```
PROC STORGRP
...
 SELECT
     WHEN (&STORCLAS = 'VTFM')
     SET &STORGRP = 'VTFM'
```

Figure 9-63 SMS storage group sample

9.6.13 Copying VTFM SAMPLIB to your own library

Copy the delivered SAMPLIB members to your own VTFM SAMPLIB library. Figure 9-64 shows you the sample JCL to copy all of the members.

```
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   DUMMY
//SYSIN    DD   *
 DELETE VTFM.JCL    NONVSAM PURGE
/*
//CREATE   EXEC PGM=IEBCOPY
//SYSPRINT DD   SYSOUT=*
//IN       DD   DSN=HVT.SHVTSAMP,DISP=SHR
//OUT      DD   DSN=VTFM.JCL,DISP=(,CATLG),UNIT=SYSDA,
//         SPACE=(TRK,(45,30,50)),DCB=*.IN
//SYSUT3   DD   SPACE=(CYL,(3,2)),UNIT=SYSDA
//SYSUT4   DD   SPACE=(TRK,(3,2)),UNIT=SYSDA
//SYSIN    DD   *
  COPY OUTDD=OUT,INDD=((IN,R))
/*
```

Figure 9-64 Sample JCL to create your own SAMPLIB library

Figure 9-65 shows the HVTOPT00 member containing the VTFM initialization options used in our environment.

```
*
 IDENT   SERVERID=WTSCPLX2          SERVER ID
*
 JRN     THRESH=85                  JOURNAL FILE THRESHOLD
*
 OPTIONS DYR=ENABLE,                DYNAMIC REALLOCATION
         JRN=ENABLE,                JOURNAL
         PAT=(DISABLE,DISABLE),     PARALLEL ACCESS TAPE
         SMF=DISABLE,               SMF USER RECORD
         SSN=ENABLE,                SUBSYSTEM
         TCP=DISABLE,               TCP/IP
         VLT=DISABLE                E-VAULT
*
 PAT     INCLUDE=PRULE,             INCLUDE ACCORDING TO PRULES
         ALIASPFX=0#                <== PAT TAPES PREFIX
*
 PERFORM MAXTASK=0,                 MAX. TASK CONCURRENCY, MIN. CPU
         VDBACC=90                  ACCESS VDB EVERY 90 SECONDS
*
 SMF     RECORD=250                 <== SMF RECORD NUMBER
*
 SUBSYS  NAME=VT01                  <== SUBSYSTEM NAME
*
 TCP     TYPE=SERVER,               RUNS AS A TCP/IP SERVER
         SERVERIP=9.12.4.202,
         PORT=6160
*
 VDB     RMODE=RW                   VDB RMODE=READ/WRITE
*
 VDEVICE ESOTERIC=VT3590,           <== VIRTUAL DEVICES ESOTERIC NAME
         VCHPID=(40,41,42,43),      <== VIRTUAL CHPID
         VARYON=16,                 VARY ONLINE 16 DEVICES ON STARTUP
         SIMULATE=3592
*
 VDSN    QUAL1=VTFMTAPE,            <== 1-ST QUAL OF VIRTUAL TAPE DSN
         QUAL2=%VOL,                2-ND QUAL OF VIRTUAL TAPE data set
         QUAL3=%TDSN                3-RD QUAL OF VIRTUAL TAPE data set
```

Figure 9-65 HVTOPT00 member

Where:

IDENT The IDENT initialization statement defines the symbolic name (identifier) of VTFM.

SERVERID= This name specifies the VTFM identifier. This identifier is used to relate the VTFM server and offline utilities to the database files. It is also used in various displays and messages and is used by the VTFM server to ensure no other similar VTFM server(s) are started simultaneously in the same system.

Value range: One to eight alphanumeric characters, $, #, or @. The first character cannot be a number.

Default: None.

JRN The JRN initialization statement defines the VDB journaling options.

THRESH= Specifies the overall percentage of used records in the journal file over which a warning message will be issued on each new written record. If the VDB journaling is disabled, this parameter is checked for syntax and is further ignored.

Value range: An integer from 50 to 99.

Default: None.

OPTIONS The OPTIONS initialization statement defines those options that are to be handled (enabled) or ignored (disabled) by VTFM. When used with the VTFM server, some of the options may be enabled while the VTFM server is running.

DRY= Specifies whether the dynamic reallocation is to be enabled or disabled on VTFM server's startup.

ENABLE Specifies that the dynamic reallocation feature is enabled on the VTFM server's startup.

DISABLE Specifies that the dynamic reallocation feature is disabled on the VTFM server's startup.

Value range: N/A.

Default: None.

JRN= Specifies whether the VDB journaling is enabled or disabled.

ENABLE Specifies that the VDB journaling is enabled.

DISABLE Specifies that the VDB journaling is disabled.

Value range: N/A.

Default: None.

PAT= Specifies whether Parallel Access Tape (PAT) is enabled or disabled on VTFM server's startup.

The parameter has two sub-parameters. The first sub-parameter is related to batch allocations, while the second sub-parameter is related to dynamic allocations. Any combination of ENABLE|DISABLE values in the two sub-parameters is acceptable.

ENABLE Specifies that PAT is enabled on the VTFM server's startup. For the first sub-parameter, the PAT feature is enabled for batch jobs only. For the second sub-parameter, the PAT feature is enabled for dynamic allocation only.

DISABLE Specifies that PAT is disabled on the VTFM server's startup. For the first sub-parameter, the PAT feature is disabled for batch jobs only. For the second sub-parameter, the PAT feature is disabled for dynamic allocation only.

Value range: N/A.

Default: None.

SMF= Specifies whether the VTFM user SMF record generation is enabled or disabled.

ENABLE Specifies that the VTFM user SMF record generation is enabled.

DISABLE Specifies that the VTFM user SMF record generation is disabled.

Value range: N/A.

Default: None.

SSN= Specifies whether the VTFM subsystem is to be enabled or disabled on the VTFM server's startup.

The VTFM subsystem is required to perform dynamic reallocation, to enable PAT, and to perform type3 and type4 e-Vault.

ENABLE Specifies that the VTFM subsystem is enabled on the VTFM server's startup.

DISABLE Specifies that the VTFM subsystem is disabled on the VTFM server's startup.

Value range: N/A.

Default: None.

TCB= Specifies whether the TCP/IP connectivity is enabled or disabled. The TCP/IP connectivity is required by the VTFM server if the e-Vault option is enabled and one or more remote off-site storage locations are type4 e-Vault locations.

ENABLE Specifies that the TCP/IP connectivity is enabled.

DISABLE Specifies that the TCP/IP connectivity is disabled.

Value range: N/A.

Default: None.

VLT= Specifies whether the e-Vault option is enabled or disabled.

ENABLE Specifies that the e-Vault option is enabled.

DISABLE Specifies that the e-Vault option is disabled.

Value range: N/A.

Default: None.

The TCP/IP connectivity is required as well if the e-Vault option is enabled and one or more remote off-site storage locations are type4 e-Vault locations.

PAT The PAT initialization statement defines the method upon which DD statements are marked as PAT-Eligible and the PAT alias volumes that are used by PAT.

ALIASPFX= Specifies the range of PAT alias volumes that PAT uses when satisfying a mount request of a PAT-Eligible DD statement. The range starts with the two characters defined in this parameter followed by an integer from 0000 to 9999.

For example, if the alias prefix is 9$, the range of PAT alias volumes is 9$0000 to 9$9999.

If the VTFM server is part of a VTFM complex and the SYSZVOLS resource is a global (SYSTEMS) resource, the aliaspfx value should be different in each server.

Value range: Two alphanumeric or national ($,#, or @) character string.

Default: None.

INCLUDE= This parameter specifies the method upon which DD statements are marked as PATEligible.

PRULE Specifies that DD statements are considered to be PATEligible according to the PAT Rules.

SHR Specifies that DD statements defining a disposition of DISP=SHR are considered to be PAT-Eligible.

CPRI Specifies that DD statements defining a DCB of DCB=CPRI=S are considered to be PAT-Eligible

Value range: N/A.

Default: None.

PERFORM The PERFORM initialization statement specifies parameters that let the VTFM server control some of its resources use.

MAXTASK= The VTFM server uses a series of subtasks (I/O subtasks) to serve I/O requests that are issued by jobs using virtual tapes. On one hand, each such subtask uses server resources (CPU and storage), and on the other hand, it provides useful work for the served jobs. This parameter provides the ability to set the trade-off between the main storage and the CPU used by the server for that purpose.

This parameter also specifies the number of I/O subtasks that are allowed to be active concurrently, use server CPU and storage, and provide useful work for the served jobs.

As the parameter value is set lower, less storage is used, but more CPU is spent for the concurrency synchronization.

As the parameter value is set higher, more storage is used and less CPU is spent for the concurrency synchronization.

If the parameter value is set sufficiently high (that is, equal to the number of defined devices), the concurrency algorithm is skipped so no CPU is spent for it at all. In this case, the maximum CPU usage efficiency is gained (an estimated decrease of 5% to 7% in the overall server CPU usage).

Thus, CPU usage reduction can be gained at the cost of more used storage. Refer to Appendix B, “Storage Usage”, in *IBM System Storage VTF Mainframe*, GC53-1190 to calculate the storage usage as a function on the other server defined resources.

Assuming VLT=DISABLE, if the site generated 128 devices, the number of concurrent tasks is 64, and 32 devices are active, the server will use approximately 76 MB.

If the number of concurrent tasks is set to its maximum (in this case, 128), the server will use approximately 114 MB. In this case, the CPU usage will drop by 5% to 7%.

This parameter is not associated with the concurrent number of virtual devices currently served by the VTFM server. If the max_task value is lower than the number of devices and I/Os are issued at once by more devices, the excess I/Os will be queued.

Value range: An integer from 16 to 256 or 0.

If it is greater than the number of generated devices or 0, it is adjusted to the number of generated devices.

The recommended value is 0, in which case the maximum CPU usage reduction is gained.

Default: None.

VDBACC= VTFM periodically accesses the VDB to check whether a reload of its contents is needed (as a result of adding or deleting new virtual tapes, scratching virtual tapes, and so on).

The frequency at which the VDB is accessed for VDB-In-Storage update can be tuned by this parameter.

A low VDBACC value enables VTFM to faster recognize any change to the VDB contents. However, it creates a great impact on the VDB, which might lead to high contention and even to delays in getting virtual tapes mounted.

A high VDBACC value minimizes the risk of experiencing contention on the VDB, but changes to the VDB contents will be recorded slower. The expected impact might be that new added virtual tapes will be recognized by the VTFM subsystem only after the VBBACC interval has expired.

The VDBACC value default of 90 seconds means that, on average, 45 seconds will pass from the time new virtual tapes are added to the VDB until the VTFM subsystem recognizes them or from the time virtual tapes are scratched until they will be mounted as a result of a scratch mount request.

interval Specifies the interval (in seconds) in which the VTFM server accesses the VDB for testing whether a VDB-In-Storage update is needed.

Value range: An integer from 10 to 600.

Default: 90.

CPU%= The VTFM server CPU% utilization can be limited to a specific percentage. The VTFM server CPU% utilization will be kept below that limit, regardless of the workload or the overall CPU% utilization in the system.

Use the standard SRM or WLM for VTFM tuning and ignore this parameter.

percent Specifies the CPU% utilization, which the server should not exceed. The percent value should be normalized. If your system has more that one CPU (and the total CPU% utilization goes beyond 100%), the percent value reflects the normalized CPU% utilization (for example, total CPU% divided by number of online CPU's). That way, the CPU% utilization seen by the server will never go beyond 100%.

Value range: An integer from 10 to 99. If CPU% is not specified or a value of 0 is specified, no CPU% utilization restrictions apply, and the server will use all resources made available to it by the system.

Default: 0.

SMF The SMF initialization statement defines the VTFM user SMF record definitions.

RECORD= Specifies the VTFM user SMF record number.

Value range: An integer from 128 to 255 or 0. If the specified value is 0, no VTFM user SMF record is written (regardless of the SMF parameter value on the OPTIONS initialization statement).

Default: 0.

SUBSYS The SUBSYS initialization statement defines the VTFM subsystem name.

NAME= Specifies the name of the VTFM subsystem. This name should be unique (not used by any other MVS subsystem). It might or might not be defined in `SYS1.PARMLIB(IEFSSNxx)`. If it is not defined, the VTFM server dynamically defines the VTFM subsystem and uses it. If it is defined and is not yet in use, the VTFM server uses it. Otherwise, the VTFM server fails in its initialization process.

Value range: A one to four alphanumeric character string. The first character must range from G to Z.

Default: None.

TCP The TCP initialization statement defines the TCP/IP connection properties of the VTFM server. This connection is required by the type4 e-Vault feature.

TYPE= Specifies whether VTFM runs as a TCP/IP server or TCP/IP client. Conceptually, the TCP/IP server waits passively for requests from one or more clients, processes the request, and returns the result to the client that originated the request. On the other side, the TCP/IP clients initiate service requests to the TCP/IP server.

The type4 e-Vault implementation requires that the VTFM server that resides at the local location be defined as a TCP/IP server. VTFM servers that reside at the remote off-site storage locations should be defined as TCP/IP clients.

SERVER The VTFM server runs in the local location and is defined as a TCP/IP server.

CLIENT The VTFM server runs in a remote off-site storage location and is defined as a TCP/IP client.

Value range: N/A.

Default: None.

SERVERIP= Specifies the dotted decimal IP address of the MVS that runs the VTFM TCP/IP server. This parameter should be specified in each one of the VTFM servers that run as TCP/IP clients. The VTFM server that runs the TCP/IP server checks the initialization statement's syntax, and then ignores it.

Value range: Each IP is an integer from 0 to 255.

Default: None.

PORT= Specifies the port number that binds to the sockets used by the VTFM TCP/IP server and clients. The port number should be identical in the server and all its clients. This port number should not be used by any other TCP/IP application in the MVS system that runs the VTFM TCP/IP server.

Value range: An integer value from 5001 to 65535.

Default: None.

TCPNAME= Specifies the name of the TCP/IP address space to which the server connects.

If there is only one TCP/IP address space and the parameter is not specified, the system derives a value from the TCP/IP configuration file.

Value range: One to eight alphanumeric characters, $, #, or @. The first character cannot be a number.

Default: The TCP/IP address space name derived from the TCP/IP configuration file.

VDB The VDSN initialization statement defines the name structure of the virtual tape data sets.

QUAL1= Specifies the high level qualifier to be assigned to the virtual tape data sets name. This data set name high level qualifier should be unique throughout the system. No other system component should allocate and catalog data sets having this value as their data set's name high level qualifier. If this condition is not observed, the VTFM recovery procedure is not able to properly locate the cataloged virtual tape data sets and is not be able to recover any VTFM virtual tape.

Value range: One to eight alphanumeric characters, $, #, or @. The first character cannot be a number.

Default: None.

QUAL2= Specifies the second qualifier to be assigned to the virtual tape data set name.

%VOL The virtual tape volume serial number preceded by "'V".

%TDSN The data set name of the first data set allocated on the virtual tape. Unless the application modifies this name before the data set is opened, this name is the name of the first tape data set written to the virtual tape. If needed, this name is truncated so that the total virtual tape data set name does not exceed 44 characters.

Value range: N/A.

Default: None.

Note: If the first tape data set name cannot be determined, the TEMPORAR value is assigned.

QUAL3= Specifies the second qualifier to be assigned to the virtual tape data set name.

%VOL The virtual tape volume serial number preceded by "V".

%TDSN The data set name of the first data set allocated on the virtual tape. Unless the application modifies this name before the data set is opened, this name is the name of the first tape data set written to the virtual tape. If needed, this name is truncated so that the total virtual tape data set name does not exceed 44 characters.

If this tape data set is a GDG data set, the GDG suffix (that is, GnnnnVnn) is truncated so as not to make the virtual tape data set a GDG-like data set.

Note: If the first tape data set name cannot be determined, a value of TEMPORAR is assigned.

qual3 Specifies a literal to be used as the third virtual tapes data set name qualifier.

Value range: N/A

Default: None

Attention:

- At least QUAL2 or QUAL3 should be set to %VOL.
- If virtual tapes are moved between type4 e-Vault locations, all locations must specify the %VOL qualifier as the same qualifier (either the second or third qualifier).

VDEVICE The VDEVICE initialization statement defines the VTFM virtual devices to be used by the VTFM server.

ESOTERIC= The VTFM virtual devices esoteric name. This name should be identical to the name you defined for VTFM in the HCD during the installation process. Either this esoteric name or the generic name defined for VTFM virtual devices in the HCD should be coded on the UNIT parameter of DD statements that use VTFM virtual devices.

Value range: One to eight alphanumeric characters, $, #, or @. The first character cannot be a number.

Default: None.

VDS Specifies virtual (existing or non-existing) channel path IDs used by VTFM virtual devices.

The following rules apply to the virtual CHPIDs support:

- As many as four virtual CHPIDs are supported. By attaching the virtual control unit to more than one virtual CHPID, the VTFM server continues to function properly even when a CHPID attached to its control unit is varied offline.
- Virtual CHPIDs connected to the VTFM virtual control unit can consist of real (existing) channels, virtual (nonexisting) channels, or both.

For a non-existing channel, it is mandatory that the output of the Display Channel Path status indicate `CHP=xx DOES NOT EXIST`. Any other description can cause unpredictable results. For example, in the following example text, non-existing CHPID 60 can be used:

```
D M=CHP(60)
IEE174I 18.19.33 DISPLAY M 963
CHPID 60: TYPE=00, DESC=UNKNOWN, OFFLINE
DEVICE STATUS FOR CHANNEL PATH 60
CHP=60 DOES NOT EXIST
****************** SYMBOL EXPLANATIONS*****************
+ ONLINE @ PATH NOT VALIDATED - OFFLINE . DOES NOT EXIST
* PHYSICALLY ONLINE $ PATH NOT OPERATIONAL
```

- Although not recommended, the VTFM control unit and its attached CHPIDs can be defined in HCD. You should define them in the VCHPID parameter (and omit them from the HCD).
- If the control unit and CHPIDs are defined in HCD, they override any VCHPID parameter. In that case, the VCHPID parameter is ignored and the HVT038W message is issued.
- If the control unit and CHPIDs are not defined in HCD, the VTFM server uses the values supplied for the VCHPID parameter.
- If two or more servers run in a single MVS system, each VTFM server should define different VCHPID values (even when the virtual devices are defined in HCD and the VCHPID values are ignored).
- Non-existing CHPIDs defined through the VCHPID parameter should not be taken offline (by means of the CF CHP(xx),OFFLINE operator command). If taken offline, the server has to be bounced in order to vary them online. Existing CHPIDs can be taken offline and online by operator commands.
- When more than one existing CHPID is defined through the VCHPID parameter (and not through HCD), they should not be taken offline all at once (because, in this case, MVS does not check for online virtual devices attached to these CHPIDs).
- When the VTFM server starts, it varies online only nonexisting CHPIDs. Existing CHPIDs should be online or brought online prior to the server startup by using the CF CHP(xx),ONLINE operator command.
- The D M=DEV(xxxx) operator command does not display real information for non-existing CHPIDs and should be ignored.
- On server start, the HVT037I message lists the final used CHPIDs.

Value range: Any VCHPID value is a two hexadecimal digit from 00 to FF.

If only one VCHPID value is specified, the parentheses can be omitted.

Default: N/A.

SIMULATE= Specifies the simulated device type. This parameter is applicable only for virtual devices generated as 3590. If specified for other generated device types, it is checked for valid syntax and is then ignored.

For further information about virtual devices types and media, refer to Chapter 3, “The Simulated Virtual Device Types”, in *IBM System Storage VTF Mainframe User’s Guide*, GC53-1191.

Note that the virtual tape capacity is always set by the TAPECAP parameter value on the TPOOL initialization statement (regardless of the simulated device type).

3590 Specifies that the generated VTFM 3590 virtual devices do not simulate other devices. Virtual tapes from all tapepools are of MEDIA3.

3592 Specifies that the generated VTFM 3590 virtual devices simulate 3592 virtual devices. Virtual tapes from all tapepools are of MEDIA5.

Value range: N/A.

Default: 3590.

VARYON= Specifies the number of devices to be varied online after VTFM server startup. Devices already online are not varied again. However, they are counted as online, so that at the end of the process the number of online devices do not exceed the count parameter.

Value range: An integer from 0 to 256 (but not exceeding the number of virtual devices defined for VTFM). If 0 is coded, no device will be varied online on VTFM server startup.

Default: 0.

9.6.14 Customizing the VTFM tapepool and diskpool

The VTFM tapepools and diskpools initialization statements are located in the HVTPOLxx member in the SAMPLIB library. We have copied the member to our VTFM.JCL library and made the changes shown in bold in Figure 9-66. The HVTPOLxx member defines the VTFM tapepools and diskpools infrastructure and must contain:

- At least one and as many as 50 DPOOL initialization statements.
- At least one and as many as 50 TPOOL initialization statements.

```
*
 DPOOL   NAME=DPOOL1,                    INTERNAL NAME FOR DISK POOL
         UNIT=(,VTFM),                   STORAGE FOR VIRTUAL DS
         THRESH=(85,1800,50)             %FULL,SEC'S TO WARN,%LOWTHRESH
*        UNIT=(#unit,#storclas,#mgmtclas)
*
 TPOOL   NAME=$$DFLT,                    DEFAULT TPOOL
         DPOOL=DPOOL1,
         RANGE1=(0,0),                   INDICATES IT IS DEFAULT POOL
         TAPECAP=(1000,16),
         THRESH=(90,1800),               %FULL,SEC'S TO WARN
         COMPRESS=0,
         DEFERBY=24,
         CKPINTVL=0
```

Figure 9-66 HVTPOL00 parmlib member

Note: We have only specified the storage class in the UNIT operand for virtual tape data sets allocation.

Where:

DPOOL The DPOOL initialization statement defines one diskpool (that is, one group of disks eligible for virtual tape data sets allocation). Usually this diskpool contains disks grouped under one esoteric name or one Storage Group.

NAME= Specifies the diskpool name. This name is later referenced by one or more tapepool(s) definition(s).

Value range: A 1 to 16 alphanumeric character string.

Default: None.

UNIT= This parameter specifies the disks (either SMS-managed or non-SMS-managed) that are eligible for virtual tape data sets allocations in this diskpool.

The rules for UNIT sub-parameters co-existence and allocation decision are:

- Either the unit or storclas sub-parameters should be specified. The MGMTCLAS subparameter is optional.
- Virtual tape data sets are allocated on non-SMS-managed disks if the unit subparameter is specified and the STORCLAS sub-parameter is not specified. The MGMTCLAS sub-parameter (if specified) is ignored.
- Virtual tape data sets are allocated on SMS-managed disks if the STORCLAS (and optionally, the mgmtclas) sub-parameter is specified. The unit sub-parameter (if specified) is ignored. The Data Class ACS routine should not assign any Data Class value to the virtual tape data sets.
- The Storage Class ACS routine should not remove the storclas value assigned by this parameter.

unit Specifies the esoteric name of a group of non-SMS-managed disks. This esoteric name should be defined in the EDT table (via HCD). The disks in this group should be of the same type (for example, only 3380 or 3390) and should have a mount attribute of STORAGE.

Value range: A one to eight alphanumeric character string.

Default: None. If not specified, the storclas (and optionally, the mgmtclas) sub-parameter should be specified.

If only the unit value is specified, the parentheses can be omitted.

storclas Specifies the name of a valid Storage Class. Adjust the appropriate Storage Group ACS routine to relate this Storage Class with the Storage Group assigned to contain the virtual tape data sets.

The Storage Class must have Guaranteed Space=N specified.

Value range: A one to eight alphanumeric or national ($,#,@) characters string. The first character must be alphabetic or national ($,#,@).

Default: None. If not specified, the unit subparameter should be specified.

mgmtclas Specifies the name of a valid Management Class.

The following restrictions apply to this Management Class:

- Expire After Days Non-Usage=NOLIMIT
- Expire After Date/Days=NOLIMIT
- Partial Release=N

Value range: A one to eight alphanumeric or national ($,#,@) characters string. The first character must be alphabetic or national ($,#,@).

Default: None.

THRESHOLD= This parameter specifies the conditions under which warning messages will be issued when the used space on the disks defined under the UNIT parameter exceeds a predefined threshold.

Additionally, it sets the low threshold target for the HVTUDST utility. The HVTUDST utility creates a list of virtual tape data sets to be de-staged, such that the threshold defined by the low% parameter for this diskpool is reached.

If an optional positional parameter (which is not the last) is omitted, a comma should be coded to indicate its position.

percent Specifies the overall percentage of used space in the esoteric name or Storage Group associated with the current diskpool over which the HVT347W message will be issued.

Value range: An integer from 50 to 99.

Default: None.

seconds Specifies the interval (in seconds) on which the used space recalculation is done and HVT347W is re-issued. The recommended value is 180.

Value range: An integer from 1 to 1800.

Default: None.

low% Specifies the percentage of used space in the esoteric name or Storage Group associated with the current diskpool, under which the HVTUDST utility stops selecting virtual tape data sets to be de-staged.

Value range: An integer from 50 to 100.

Default: 100. (No virtual tape data sets are selected.)

volmask Specifies the group of disks to be included in the used space calculation. If the parameter is omitted, all disks in the diskpool participate in the calculation. Otherwise, only the disks whose volser matches the volmask value participate in the calculation.

The volmask parameter is effective for non-SMS-managed diskpools only. If specified on an SMS-managed diskpool, it is ignored.

The volmask parameter accepts mask characters as character placeholders.

Value range: A one to six alphanumeric or national ($,#,@) character string.

Default: *.

9.6.15 Allocating and formating the VTFM database files

In this section, we describe how you can allocate and format the VTFM database files.

Allocating and formating the journal file

Edit the HVTUJFR member in the SAMPLIB library, which contains a job that:

- Deletes the existing VTFM journal file (if any)
- Defines a new journal file
- Initializes the new journal file

Substitute the #-prefixed parameters as required and then save the member. Calculate the journal file size and customize the utility initialization statements according to the HVTUJFR utility explanation. Optionally, uncomment the SMS parameters and submit the job.

Planning the journal file size

The journal file size is automatically calculated according to user parameters that consider all the expected manipulations done to the VDB in a given time range.

The formula shown in Figure 9-67 is used to calculate the size of the journal.

```
blk#=((days*mntsday)*4+utils)/16
```

Figure 9-67 Calculating the journal file size

Where:

blk#	The number of blocks required for the file.
days	The number of days between two consecutive VDB backups.
mntsday	The number of virtual tape mounts per day.
utils	The number of virtual tapes manipulated by VTFM offline utilities during the days period.

Note: The journal file record size is 384 bytes. The allocation is done in blocks. Each block contains 16 records (plus a 16 byte block header), and its size is 6160 bytes.

Figure 9-68 shows you the sample JCL that can be used to allocate and format the VTFM journal file.

```
//UJFR     EXEC VTFMSRV,SERVICE=UJFR     <== PROCEDURE NAME
//HVTVDB   DD DSN=NULLFILE
//HVTJRN   DD DSN=NULLFILE
//HVTUTIL  DD *
 FORMAT  JRNNAME=VTFMJRN.JOURNAL,        <== JOURNAL FILE data set NAME
         DAYS=8,
         MNTSDAY=600,
         UTIL=600,
*        STORCLAS=#jrnsc,                <== STORCLAS FOR JRN FILE
*        MGMTCLAS=#jrnmc,                <== MGMTCLAS FOR JRN FILE
*        DATACLAS=#jrndc,                <== DATACLAS FOR JRN FILE
         UNIT=3390,                      <== UNIT FOR JRN FILE
         VOL=SBOXOP                      <== DISK VOLUME FOR JRN FILE
//
```

Figure 9-68 Sample JCL to allocate and format the journal

Where:

JRNNAME The journal file data set name.

DAYS The number of days between two consecutive VDB backups.

mntsday The number of virtual tape mounts per day.

utils The number of virtual tapes manipulated by VTFM offline utilities during the days period.

UNIT A specific or esoteric unit name.

VOL The volume serial number to which the journal file is allocated.

The job should end with a return code of 0.

Allocating and formating the VDB

Edit the HVTUVSI member in the SAMPLIB library. Substitute the #-prefixed parameters as required and then save the member. Note that the job's first step is a dummy step that never executes. Calculate the VDB size according to the HVTUVSI utility by using the following settings:

RECORDS The number of virtual tapes expected to be contained in the VDB.

Value range: N/A.

Default: 200,000.

CISZ The cluster's data component control interval size. The optimal value for a 3380 disk is 22528, and the optimal value for a 3390 disk is 18432.

Value range: N/A.

Default: 18432.

RECORDSIZE The size of one record representing one virtual tape entry in the VDB. The current record size is 256 and should not be changed.

Value range: N/A.

Default: 256.

Figure 9-69 shows you a sample JCL that can be used to allocate and format the VTFM journal file. Optionally, uncomment the SMS parameters and submit the job.

```
//VDB1     EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
 DEFINE CLUSTER -
        (NAME(VTFMVDB.DATABASE)        /*<== VDB FILE data set NAME*/ -
         SHAREOPTIONS (2 3)                                          -
         RECORDS(200000 1000)          /* 200,000 VIRTUAL TAPES   */ -
         SPANNED                                                     -
         SPEED                                                       -
      /*  STORCLAS(#vdbsc)          /*<== STORCLAS FOR VDB FILE    */ -
      /*  MGMTCLAS(#vdbmc)          /*<== MGMTCLAS FOR VDB FILE    */ -
      /*  DATACLAS(#vdbdc)          /*<== DATACLAS FOR VDB FILE    */ -
          VOLUME(SBOXOP))               /*<== DISK VOL FOL VDB FILE*/ -
        DATA                                                         -
         (NAME(VTFMVDB.DATABASE.DATA)   /*<== VDB FILE data set NAME*/ -
          CISZ(18432)                   /*FOR 3390.(22528 FOR 3380)*/ -
          KEYS(7 0)                                                  -
          RECORDSIZE(256 256))          /* 1 VOL = 256 BYTES       */ -
        INDEX                                                        -
         (NAME(VTFMVDB.DATABASE.INDEX)) /*<== VDB FILE data set NAME*/
//*
//VDB2     EXEC VTFMSRV,SERVICE=UVSI     <== PROCEDURE NAME
//HVTJRN   DD DSN=NULLFILE
```

Figure 9-69 Sample JCL to allocate the VDB

Where:

RECORDS The number of virtual tapes expected to be contained in the VDB.

Value range: N/A.

Default: 200,000.

VOL The volume serial number to which the VDB is allocated.

CISZ The cluster's data component control interval size. The optimal value for a 3380 disk is 22528, and the optimal value for a 3390 disk is 18432.

Value range: N/A.

Default: 18432

RECORDSIZE The size of one record representing one virtual tape entry in the VDB. The current record size is 256 and should not be changed.

Value range: N/A.

Default: 256

SERVICE=UVSI The HVTUVSI utility can be used to initialize a newly defined VDB file. The utility will fail if the VDB is set to Read_Only mode.

Note: An attempt to initialize an existing non-empty VDB file will fail.

The job should end with a return code of 0.

Allocating the history and the history backup files

Edit the HVTUHST member in the SAMPLIB library. It contains a job that allocates the history file and the history backup file. The history backup file name is similar to the history file suffixed by .BKP. Substitute the #-prefixed parameters as required and then submit the job.

Figure 9-70 shows you sample job to create and initialize the history file. Optionally, uncomment the SMS parameters and submit the job.

```
//ALLOCHST EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DUMMY,
//            DCB=(BLKSIZE=27998,LRECL=512,RECFM=VB,DSORG=PS)
//SYSUT2   DD DSN=VTFMHST.HISTORY,       <== HST FILE data set NAME
//            DISP=(,CATLG,DELETE),
//            UNIT=3390,                 <== HST FILE UNIT
//            VOL=SER=SBOXOP,            <== HST FILE DISK VOLUME
//*           STORCLAS=#hstsc,           <== HST FILE STORCLAS
//*           MGMTCLAS=#hstmc,           <== HST FILE MGMTCLAS
//*           DATACLAS=#hstdc,           <== HST FILE DATACLAS
//            SPACE=(TRK,(3400)),        APPROX. 500,000 data set RCRDS
//            DCB=(BLKSIZE=27998,LRECL=512,RECFM=VB,DSORG=PS)
//SYSIN    DD DUMMY
//*
```

Figure 9-70 Sample JCL to allocate the HISTORY file

Where:

DSN Data set name of the VTFM history file.

VOL=SER Volume serial number to which the VTFM history file is allocated.

SPACE The allocation quantity in tracks.

The job should end with a return code of 0.

Figure 9-71 shows you a sample job that you can use to create and initialize the history backup file. Optionally, uncomment the SMS parameters and submit the job.

```
//ALLOCBKP EXEC PGM=IEBGENER
//SYSPRINT DD SYSOUT=*
//SYSUT1   DD DUMMY,
//            DCB=(BLKSIZE=27998,LRECL=512,RECFM=VB,DSORG=PS)
//SYSUT2   DD DSN=VTFMHST.HISTORY.BKP,   <== HST BKP FILE data set NAME
//            DISP=(,CATLG,DELETE),
//            UNIT=3390,                 <== HST BKP FILE UNIT
//            VOL=SER=SBOXOP,            <== HST BKP FILE DISK VOLUME
//*           STORCLAS=#hstbsc,          <== HST BKP FILE STORCLAS
//*           MGMTCLAS=#hstbmc,          <== HST BKP FILE MGMTCLAS
//*           DATACLAS=#hstbdc,          <== HST BKP FILE DATACLAS
//            SPACE=(TRK,(3400)),        APPROX. 500,000 data set RCRDS
//            DCB=(BLKSIZE=27998,LRECL=512,RECFM=VB,DSORG=PS)
//SYSIN    DD DUMMY
```

Figure 9-71 Sample JCL to allocate the HISTORY backup file

Where:

DSN	Data set name of the VTFM history backup file.
VOL=SER	Volume serial number to which the VTFM history file is allocated.
SPACE	The allocation quantity in tracks.

The job should end with a return code of 0.

9.7 Starting the VTFM server

Start the VTFM server by issuing the operator command shown in Figure 9-72.

```
S VTFMSRV
```

Figure 9-72 Start VTFM server

Where VTFMSRV is the VTFM procedure name.

Verify that the startup was successful by checking that the HVT009I message was issued. You should see that the virtual tape drives are varied online, as shown in Figure 9-73.

```
HVT304I RUL    REFRESH WAS OK, RC=00
HVT019I DYR IS NOW ENABLED
HVT039I HW: TYPE=2094,MODEL=710 ,SERIAL=02991E 690
HVT039I (CONT.) SW: NAME=z/OS    ,VERSION=01.11.00
HVT037I WTSCPLX2 SERVER: CUU=0400-041F, ES=VT3590  , GN=VT3590G2, 691
HVT037I (CONT.) DT=X'44048083', S=3592, CHP=(40 41 42 43)
HVT009I INITIALIZATION OF WTSCPLX2 SERVER SUCCESSFULLY COMPLETED
V 0400-040F,ONLINE
HVT015I WTSCPLX2 SERVER DISKPOOL(S) SCAN COMPLETED AFTER POOLS REFRESH
F VTFMSRV.VTFMSRV,D=DSP
HVT432I SG=VTFM     VOL#=    2 TRKS=    880_K USED= 78% DPOOL=DPOOL1
IEE457I 16.00.56 UNIT STATUS 697
UNIT TYPE STATUS        VOLSER     VOLSTATE
0400 VT35 O-NRD                    /REMOV
0401 VT35 O-NRD                    /REMOV
...
HVT015I WTSCPLX2 SERVER TAPEPOOL(S) SCAN COMPLETED AFTER POOLS REFRESH
F VTFMSRV.VTFMSRV,D=TSP
HVT433I TAPES=    201 ACT=     51( 25%) SCR=    150 TPOOL=$$DFLT
HVT434I TOTAL: TAPES=    201 ACT=     51( 25%) SCR=    150
```

Figure 9-73 VTFMSRV startup messages

9.8 Stopping the VTFM server

To stop the VTFM server, issue the operator command shown in Figure 9-74.

```
   P VTFMSRV
```

Figure 9-74 Stop VTFM server

Where:

VTFMSRC is the VTFM procedure name.

9.9 VTFM utilities

In this section, we provide an overview of the following VTFM utilities:

HVTUDST This utility can be used to control the utilization level of the VTFM diskpools.

HVTUREP This utility can be used to list the history file contents.

9.9.1 Enabling the HVTUDST utility to write the migration job

Edit the HVTUDST member in the SAMPLIB library. It contains a sample job that executes the HVTUDST utility.

The HVTMIG DD statement in that sample job defines a DUMMY data set.

In order for the migration job to be created on that DD statement, define it as a non-DUMMY data set (internal reader, production library, or other).

Note: The defined DCB parameters should not be altered or removed.

The HVTUDST utility can be used to control the utilization level of the VTFM diskpools. It is responsible for prioritizing the virtual tape data sets de-staging and providing the input for any storage management product that performs the migration and recall operations. However, it does not perform the migration of the virtual tape data sets by itself.

Figure 9-75 shows you a sample job that can be used to create and initialize the history backup file. Optionally, uncomment the SMS parameters and submit the job.

```
//UDST     EXEC VTFMSRV,SERVICE=UDST     <== PROCEDURE NAME
//HVTHST   DD DSN=VTFMHST.HISTORY,       <== HST FILE data set NAME
//            DISP=SHR
//HVTHSTBK DD DSN=VTFMHST.HISTORY.BKP,   <== HST BKP FILE data set NAME
//            DISP=SHR
//HVTMIG   DD DSN=VTFM.JCL.HVTMIG,                    MIGRATION JOB
//            SPACE=(CYL,(65,5),RLSE),UNIT=SYSDA,
//            DCB=(RECFM=FB,LRECL=80,BLKSIZE=27920)   DCB IS MANDATORY
//HVTUTIL  DD *
 DRUN      PROD=HSM                              PRODUCT=HSM/ABR/DMS
*
 DRULE     INC,DSN=M20.ECOO*
 DRULE     EXC,DSN=DFHSMMG.*
 DRULE     EXC,DSN=RMM.*
*
 DPTRN     NAME=G????V??,QUAL#=99
```

Figure 9-75 Sample job to update the HISTORY file

Where:

VTFMSRV The procedure name of your VTFM server.

HVTHST The VTFM history file.

HVTHSTBK The VTFM history backup file.

HVTMIG The file to which the JCL and de-stage statements of the selected data sets are written.

PROD The PROD parameter determines the storage management product for which the migration job is created.

HSM Use this option when the local storage management product is DFSMShsm.

When using this option, HVTUDST uses member HVTSKHSM in the VTFM SAMPLIB library to create a migration job. The above member is a skeleton in which:

- One selected virtual tape data set names replaces the %%DSN keyword.
- The text lines between the)REPEATSTART and)REPEATEND statements are repeated for each virtual tape data set.

ABR Use this option when the local storage management product is FDR/ ABR.

When using this option, HVTUDST uses the HVTSKABR member in the VTFM SAMPLIB library to create a migration job. The above member is a skeleton in which:

- One selected virtual tape data set names replaces the %%DSNkeyword.
- The text lines between the)REPEATSTART and)REPEATEND statements are repeated for each virtual tape data set.

DMS Use this option when the local storage management product is CADisk.

When using this option, HVTUDST uses the HVTSKDMS member in the VTFM SAMPLIB library to create a migration job. The above member is a skeleton in which:

- One selected virtual tape data set names replaces the %%DSN keyword.
- The text lines between the)REPEATSTART and)REPEATEND statements are repeated for each virtual tape data set.

Value range: N/A.

Default: None.

DRULE

The DRULE initialization statement identifies tape data set names for which the associated virtual tape data set will be INCluded or EXCluded from the de-stage process regardless of the calculated grade. The following rules apply to the DRULE initialization statements:

– Each dsname value is matched against all DRULE initialization statements starting with the first. When a DRULE initialization statement matches, the associated tape data set is INCluded or EXCcluded respectively, and no further statements are scanned.

– If the diskpool low threshold (the third positional parameter of the respective DPOOL initialization statement) is reached and more tape data sets should be INCluded, they are not INCluded.

INC/EXC

Specifies that the current DRULE initialization statement is an INC or EXC type DRULE initialization statement.

Value range: N/A.

Default: None.

DSN

Specifies the tape data set name whose associated virtual tape data set is to be INCluded or EXCluded from the de-stage process. This tape data set name should be the first on the virtual tape; otherwise, it is ignored.

The DSN parameter value accepts mask characters as character placeholders. The following mask rules apply when using these character placeholders:

- An asterisk (*) matches none, one, or more successive characters.
- A question mark (?) matches exactly one character.

Value range: A 1 to 44 alphanumeric character string, including periods. Each qualifier between the periods can be one through eight alphanumeric or national ($,#,@) characters. The first character must be alphabetic or national ($,#,@).

Default: None.

DPTRN

The DPTRN initialization statement defines patterns according to which the HVTUDST utility derives generic names from tape data set names.

The order of the DPTRN initialization statements is important, as the HVTUDST utility considers the first pattern that matches the virtual tape data set name. Once a pattern matches a tape data set name and a generic name is derived, no more DPTRN initialization statements are scanned.

NAME

Specifies the name portion of the pattern definition.

According to the QUAL# parameter value, the NAME value can represent one data set name qualifier or a full data set name (including character placeholders).

The NAME parameter value accepts mask characters as characters placeholders. The following mask rules apply when using these characters placeholders:

- An asterisk (*) matches none, one, or more successive characters.

- A question mark (?) matches exactly one character.

Value range: A 1 to 44 alphanumeric or national ($, #, @) character string. The first character must be alphabetic or national ($, #, @).

Default: N/A.

QUAL Specifies the qualifier level portion of the pattern definition. When the specified pattern in the current DPTRN initialization statement matches the specified qualifier level in a given data set name, a generic name can be derived.

Two special values can be specified:

0 The NAME parameter value is considered to represent a full tape data set name (including character placeholders).

99 The last qualifier within the data set name.

Figure 9-76 shows you the sample report you get when you run the HVTUDST utility.

```
HVT304I POL    REFRESH WAS OK, RC=00
HVT200I VTF Mainframe VER 3.0.0 UTILITY HVTUDST  STARTED

                ======== PHASE 01 - CONSOLIDATE HST FILE ========


                ======== PHASE 02 - LOAD DPOOLS =================


                ======== PHASE 03 - FILL DPOOL NAME + GENERIC ===


                ======== PHASE 04 - ASSIGN GRADE ================


HVT450I ****DPOOL=DPOOL1          SG=VTFM     TRKS=    880_K USED= 78%  LTHRESH= 50% DTRKS=    246_K
HVT451I VT0035 T=  18720 C= 10 S=  0 M=   0 G=2147483646 OB * D=  47  HSM.BACKTAPE.data set
....
HVT451I VT0001 T=   1170 C=  0 S=  0 M=   0 G=2147483647 EX   D=   1  RMM.X1234567.B1234567.C1234567
HVT451I VT0019 T=   1170 C=  2 S=  0 M=   0 G=2147483647 EX   D=   1  RMM.B1234567.B1234567.C1234567

                ======== PHASE 05 - PREPARE MIGRATION JOB =======

HVT453I DE-STAGE JOB PREPARATION WAS OK,   14 VIRTUAL TAPE data setS INCLUDED

                ======== PHASE 06 - UPDATE HST FILE =============

HVT448W LAST UTILITY RUN WAS LESS THAN 23 HOURS AGO, NEW HISTORY OBSERVATIONS NOT ADDED
HVT201I UTILITY HVTUDST  ENDED, RC=04
```

Figure 9-76 HVTUDST sample report

Where:

HVT448W The last utility run was less than 23 hours ago, and new history observations have not been added.

The HVTUDST utility was found to run more than once in a less than a one day interval. Because the events observations are recorded on a daily basis, multiple runs of the utility in a less than a one day interval would cause multiple observations to be arbitrarily recorded. Therefore, this run will not add any new observations to the history file data set records.

The 23 hours is considered to mark one day because, due to various reasons, production sites jobs might run in advance or be delayed.

Consideration: The HVTUDST utility can be run only one time a day.

Enabling HVTUREP utility to list the history file contents

Edit the HVTUREP member in the SAMPLIB library. It contains a sample job that executes the HVTUREP utility. Uncomment the JCL statements that define the HVTHST DD statement and save the member.

Figure 9-77 shows you a sample job that creates VTFM REP001 reports.

```
//UREP     EXEC VTFMSRV,SERVICE=UREP     <== PROCEDURE NAME
//*HVTHST   DD DSN=#hstname,              <== HST FILE data set NAME
//*            DISP=SHR
//HVTUTIL  DD *
 REPORT  REP=REP001,LINECT=65,
                    ONLYNSCR=YES
//
                    VOLSER=VOL*,
                    ONLYNSCR=YES,
                    ONLYSCR=YES,
                    ONLYINU=YES,
                    ONLYSL=YES,
                    ONLYERR=YES
```

Figure 9-77 Sample job to create REP001 reports

Figure 9-78 shows you a sample job that creates a VTFM REP002 report. REP002 does not support any of the ONLY subparameters or the VOLSER subparameter.

```
//UREP     EXEC VTFMSRV,SERVICE=UREP     <== PROCEDURE NAME
//*HVTHST   DD DSN=#hstname,              <== HST FILE data set NAME
//*            DISP=SHR
//HVTUTIL  DD *
 REPORT  REP=REP002,LINECT=65
//
```

Figure 9-78 Sample job to create REP002 report

9.10 ICF user catalog considerations

In this section, we show you the correct use of the VIRTUAL DEVICES GENERIC NAMES to catalog or re-catalog correctly your data sets that reside on VTFM managed volumes.

Figure 9-79 shows a sample JCL that can be used to create a data set on a VTFM managed virtual tape and catalog it.

```
//STE00001 EXEC PGM=IEBGENER
//SYSPRINT DD   DUMMY
//SYSUT1   DD   DISP=SHR,DSN=MHLRES7.RMM.CNTL(T)
//SYSUT2   DD   DISP=(,CATLG),UNIT=VT3590,
//         DSN=RMM.TEST.VT3590.BATCHX
//SYSIN    DD   DUMMY
```

Figure 9-79 Creating a data set on a VTFM managed volume

All data sets that are created inside a Virtual Tape for Mainframe has the x'44048083' catalog device type, as shown in Figure 9-80.

```
 LISTC ENT('RMM.TEST.VT3590.BATCHX') ALL
NONVSAM ------- VTFM.IVP001A
     IN-CAT --- UCAT.VTFM
     HISTORY
       data set-OWNER-----(NULL)      CREATION--------2009.224
       RELEASE----------------2      EXPIRATION------0000.000
     VOLUMES
       VOLSER------------VT0017      DEVTYPE------X'44048083'      FSEQN-----------------1
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
```

Figure 9-80 Catalog device type

9.10.1 Virtual devices generic names

You must use one of the device names you have coded in HVTJ3X11 exit to re-catalog a data set, as shown in Figure 9-81.

```
....
*-------------------------------------------------------------*
*  VIRTUAL DEVICES GENERIC NAMES (AS DEFINED IN EXIT002)      *
*-------------------------------------------------------------*
UTAB     DS    0A
         DC    CL8'VT3480GN'
$UTABL   EQU   *-UTAB
         DC    CL8'VT3480G2'
         DC    CL8'VT3490GN'
         DC    CL8'VT3490G2'
         DC    CL8'VT3590GN'
         DC    CL8'VT3590G2'
$UTAB#   EQU   (*-UTAB)/$UTABL
*
APARNUM  DC    CL7'HVTU003'        APAR NUMBER
```

Figure 9-81 Sample HVTJ3X11 virtual devices generic name table

9.10.2 Recataloging VTFM managed data sets

Use one of the virtual devices you have defined in the HVT002 user exit in the DEVT operand of the DEFINE command, as shown in Figure 9-82, in order to obtain the correct device type.

```
 DEF NONVSAM (NAME('MHLRES7.TEST.VT3590G2') -
     VOL(VT0002) FSEQN(1) DEVT(VT3590G2))
```

Figure 9-82 Define NONVSAM using DEVT(VT3590G2)

As you can see in Figure 9-83, the device type is set correctly.

```
 LISTC ENT('MHLRES7.TEST.VT3590G2') ALL
NONVSAM ------- MHLRES7.VT3590G2
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       data set-OWNER-----(NULL)      CREATION--------2009.227
       RELEASE----------------2      EXPIRATION------0000.000
     VOLUMES
       VOLSER------------VT0002      DEVTYPE------X'44048083'      FSEQN------------------0
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
```

Figure 9-83 List catalog result

In our second example, shown in Figure 9-84, we used the esoteric device type we defined and not one of the virtual devices matching the HVT002 user exit.

```
  DEF NONVSAM (NAME('MHLRES7.TEST.VT3590') -
      VOL(VT0002) FSEQN(1) DEVT(VT3590))
```

Figure 9-84 Define NONVSAM using DEVT(VT3590)

We can see in the output of the LISTC command, shown in Figure 9-85, that a wrong device type is set.

```
 LISTC ENT('MHLRES7.TEST.VT3590') ALL
NONVSAM ------- MHLRES7.TEST.VT3590
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       data set-OWNER-----(NULL)      CREATION--------2009.227
       RELEASE----------------2      EXPIRATION------0000.000
     VOLUMES
       VOLSER------------VT0002      DEVTYPE------X'00188000'      FSEQN-----------------1
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
```

Figure 9-85 List catalog result with wrong device type information

Important: Use the correct catalog device type information to re-catalog a data set residing on a VTFM virtual tape volume.

9.11 Scratching or reclaiming Virtual Tapes

After a virtual tape is used by VTFM, it is marked as “active” in the VDB and it is not eligible for mount as scratch. These virtual tapes are scratched by the HVTUSCR utility (as directed by the tape management system). As the utility scratches the virtual tapes, it releases the disk space used by them while they were active. The actual scratch of the virtual tapes may be deferred according to the DEFERBY parameter’s value of the VTFM TPOOL initialization statement.

You can specify how long a volume should be retained by the VTFM before the status is really changed back to scratch, because all data sets on it are erased at this time. In our example, we have specified that a volume will be “pending scratch” for 23 hours, shown in Figure 9-86.

```
DPOOL NAME=DPOOL1, INTERNAL NAME FOR DISK POOL
UNIT=(,VTFM), STORAGE FOR VIRTUAL DS
THRESH=(85,1800,50) %FULL,SEC'S TO WARN,%LOWTHRESH
* UNIT=(#unit,#storclas,#mgmtclas)
*
TPOOL NAME=$$DFLT, DEFAULT TPOOL
DPOOL=DPOOL1,
RANGE1=(0,0), INDICATES IT IS DEFAULT POOL
TAPECAP=(1000,16),
THRESH=(90,1800), %FULL,SEC'S TO WARN
COMPRESS=0,
DEFERBY=23,
CKPINTVL=0
```

Figure 9-86 VTFM scratch pool definition

The DEFERBY parameter specifies if and by how much time virtual tapes scratching should be delayed by HVTUSCR utility, where:

hours Specifies the number of hours to pass from the time HVTUSCR first encountered a scratch-eligible virtual tape in this tapepool until the virtual tape can be actually scratched.

The virtual tape scratch is not done automatically when the deferred time expires, but on the next run of the HVTUSCR utility.

The following special values exist:

0 No defer-scratch occurs and the HVTUSCR utility immediately scratches any scratch-eligible virtual tape from this tapepool.

9999 The HVTUSCR will never scratch virtual tapes from this tapepool. Use this value when:

- The virtual tapes from this tapepool are managed by an EDM and you did not define them to the local tape management system.
- This site is used as a remote off-site storage location for type4 e-Vault virtual tapes and the virtual tapes are not defined to the local tape management system.

Value range: An integer from 0 to 1800 or 9999.

Default: 0.

DEFERBY This parameter specifies if and by how much time virtual tapes scratching should be delayed by HVTUSCR utility.

hours Specifies the number of hours to pass from the time HVTUSCR first encountered a scratch-eligible virtual tape in this tapepool until the virtual tape can be actually scratched. The virtual tape scratch is not done automatically when the deferred time expires, but on the next run of the HVTUSCR utility.

The following special values exist:

0 No defer-scratch occurs and the HVTUSCR utility immediately scratches any scratch-eligible virtual tape from this tapepool.

9999 The HVTUSCR will never scratch virtual tapes from this tapepool. Use this value when:

- The virtual tapes from this tapepool are managed by an EDM and you did not define them to the local tape management system.
- This site is used as a remote off-site storage location for type4 e-Vault virtual tapes and the virtual tapes are not defined to the local tape management system.

Value range: Integer from 0 to 1800 or 9999.

Default: 0.

Figure 9-87 shows you a VTFM inventory report and that some of the volumes are retained by the DEFERBY setting. See the (0023H) information in column SCR.

```
118.40.47 HVT304I POL    REFRESH WAS OK, RC=00                                                                    *HVTRFR  *
 18.40.47 HVT200I VTF Mainframe VER 3.0.0 UTILITY HVTUREP  STARTED                                              *HVTUREP *

 1VOLUME SCR       LOCID___(VD) LB U R SID  ALC_M* JOBNAME   JOBID    DATE     TIME   VIRTUAL TAPE data set NAME
  ------ --------- ------------ -- - - ---- ------ -------- -------- -------- -------- -----------------------------------
 0IVP001 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10503 2009.222 13:42:20 VTFMTAPE.VIVP001.HAIMO.TEST01
  VT0000 YES       *MAIN*       SL
  VT0001 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10505 2009.222 13:46:14 VTFMTAPE.VVT0001.HAIMO.TEST02
  VT0002 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10506 2009.222 13:54:37 VTFMTAPE.VVT0002.HAIMO.TEST02
  VT0003 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10507 2009.222 13:54:58 VTFMTAPE.VVT0003.HAIMO.TEST03
  VT0004 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10509 2009.222 14:01:07 VTFMTAPE.VVT0004.VTFM.IVP001A
  VT0005 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10510 2009.222 14:06:56 VTFMTAPE.VVT0005.HAIMO.TEST04
  VT0006 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10511 2009.222 14:08:05 VTFMTAPE.VVT0006.VTFM.IVP001A
  VT0007 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10513 2009.222 14:26:06 VTFMTAPE.VVT0007.HAIMO.TEST05
  VT0008 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10519 2009.222 14:51:29 VTFMTAPE.VVT0008.VTFM.IVP001A
  VT0009 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10519 2009.222 14:51:29 VTFMTAPE.VVT0009.VTFM.IVP002
  VT0010 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10520 2009.222 14:59:58 VTFMTAPE.VVT0010.VTFM.IVP001A
  VT0011 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10520 2009.222 14:59:58 VTFMTAPE.VVT0011.VTFM.IVP002
  VT0012 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10521 2009.222 15:01:48 VTFMTAPE.VVT0012.VTFM.IVP001A
  VT0013 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10521 2009.222 15:01:48 VTFMTAPE.VVT0013.VTFM.IVP002
  VT0014 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10522 2009.222 15:03:04 VTFMTAPE.VVT0014.HAIMO.IVP001A
  VT0015 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10522 2009.222 15:03:04 VTFMTAPE.VVT0015.HAIMO.IVP002
  VT0016 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10577 2009.222 20:40:28 VTFMTAPE.VVT0016.VTFM.IVP001A
  VT0017 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10732 2009.224 00:53:17 VTFMTAPE.VVT0017.VTFM.IVP001A
  VT0018 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10732 2009.224 00:53:17 VTFMTAPE.VVT0018.VTFM.IVP002
  VT0019 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10734 2009.224 01:27:30 VTFMTAPE.VVT0019.RMM.VTFM.IVP001A
  VT0020 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10734 2009.224 01:27:31 VTFMTAPE.VVT0020.RMM.VTFM.IVP002
  VT0021 YES       *MAIN*       SL
  VT0022 YES       *MAIN*       SL
....
```

Figure 9-87 VTFM volume inventory report

Where:

NO(0023H) The delay time before the tape is really scratched. This time is set by the DEFERBY parameter in your TPOOL settings.

Note: You can also bring back a single volume in status scratch by using the HVTUTIL DD statement in the VTFM procedure.

9.11.1 Using the VTFMSRV procedure to change a volume's status

Instead of updating the whole VTFM database, you can update a single volume to change the volume status from private to scratch or reclaim a volume from its deferred-scratch status back to private. Use the VTFMSRV VTFM procedure and add a SYNC control statement for each volume you would like to process. Figure 9-88 shows you an example.

```
//USCR     EXEC VTFMSRV,SERVICE=USCR     <== PROCEDURE NAME
//HVTUTIL  DD *
 SYNC      TYPE=VOLSER,VOLSER=VT0000,FORCE=YES
//
```

Figure 9-88 Use of SYNC for a single volume

The syntax of the SYNC operand is shown in Figure 9-89.

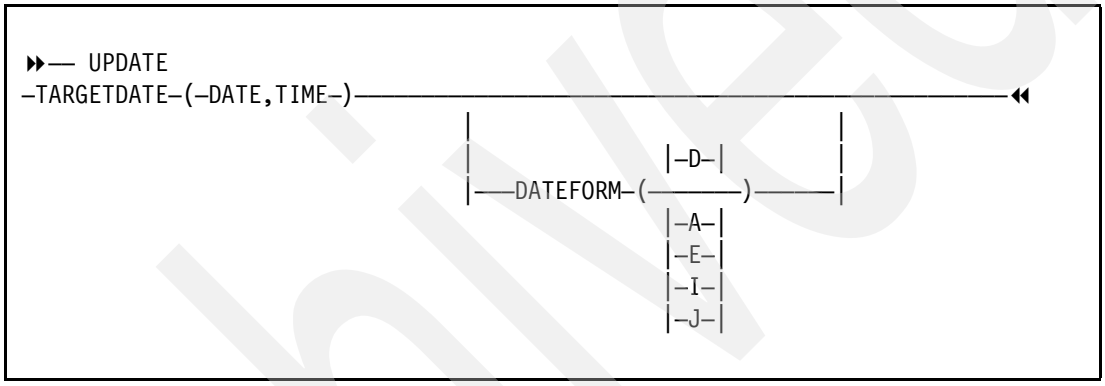


Figure 9-89 SYNC command syntax

Where:

SYNC The SYNC initialization statement requests that the scratch and e-Vault status of all the entries in the VDB be synchronized with the tape management system database or MVS catalog, or that a specific virtual tape entry be synchronized as per the user request.

TYPE The TYPE parameter defines the method upon which the VDB synchronization will be performed.

VOLSER When using this option, HVTUSCR handles separately each virtual tape specified in the VOLSER parameter. The virtual tape is handled regardless of its status in the tape management system database or MVS catalog. Incorrect use of this option might cause discrepancies between the virtual tape status in the VDB and its status in the tape management system database (if used).

If LOCID=locid is not specified, refer to HVTUSCR process when TYPE=volser is coded and LOCID=locid is not specified. If LOCID=locid is specified, the virtual tape is not scratched. Refer to the LOCID=locid parameter in Chapter 2, "VTFM Initialization Statements", in the *IBM VTF Mainframe Customization & Reference*, GC53-1190 for a description of the HVTUSCR process for that option.

If more than one specific virtual tape will be handled, the SYNC initialization statement should be repeated. A sample job to run HVTUSCR with TYPE=VOLSER can be found in the HVTUSCRV member in the VTFM SAMPLIB library.

CATALOG

This option is used by sites that do not use any tape management system and the data set residing on virtual tapes are cataloged (either in the master or user catalog(s)). A virtual tape where all the data sets residing on it are not cataloged is considered to be scratch.

During the MVS catalogs scan, the utility lists all relevant entries from the master and all connected user catalogs. User catalogs not connected to the master catalog are not scanned and the tape data sets cataloged in it are not listed. That means that virtual tapes that contain data sets cataloged in user catalogs accessed via JOBCAT or STEPCAT DD statements will be scratched. Accordingly, HVTUSCR with TYPE=CATALOG cannot be used when jobs catalog their data sets in user catalogs not connected to the master catalog.

If one or more user catalogs are not accessible, the utility terminates and no virtual tape is scratched. You might have connected user catalogs that are not accessible and do not really contain relevant entries. To allow the HVTUSCR process to complete, you can exclude these irrelevant user catalogs from the utility catalogs' scan by specifying their names in the XUCAT001-XUCAT999 DD statements. If the disks where these user catalogs reside are not accessible, you need to specify any existing disk volume serial number and unit along with the user catalog names on the XUCAT001-XUCAT999 DD statements. A sample job to run HVTUSCR with TYPE=CATALOG can be found in the HVTUSCRC member in the VTFM SAMPLIB library.

RMM

When using this option, the HVTUSCR utility scans a DFSMSrmm EDGHSKP report produced in an earlier step of the same job. The report contains the current TCDB tapes from which the virtual tapes that are not scratch (active), along with their vault location, are filtered for processing.

All virtual tapes not listed in this report are considered to be scratch.

Refer to *IBM System Storage VTF Mainframe Customization and Reference*, GC53-1190 for details about the HVTUSCR process when TYPE=RMM.

A sample job to run HVTUSCR with TYPE=RMM can be found in the HVTUSCRR member in the VTFM SAMPLIB library.

It is mandatory for the EDGHSKP report to be produced in the same job that runs the HVTUSCR utility, or the utility might scratch active virtual tapes created after the job run starts. The report generator used by previous VTFM releases can be used with the current VTFM release.

Value range: N/A.

Default: None.

VOLSER

volser This parameter is used only when TYPE=VOLSER is used and specifies the virtual tape volume serial number to be handled by the utility.

Value range: A one to six alphanumeric character string.

Default: None.

RECLAIM This parameter is used only when TYPE=VOLSER and indicates whether the specific virtual tape should be reclaimed from its deferred-scratch status. For other TYPE values, the reclamation is handled automatically and this parameter is ignored.

YES The virtual tape should be reclaimed from its deferred-scratch status. No action is taken if the virtual tape has already been scratched.

NO The virtual tape should not reclaimed from its deferred-scratch status.

Value range: N/A.

Default: NO.

FORCE This parameter indicates whether virtual tapes eligible for scratch will be scratched regardless of their deferred-scratch conditions.

YES A virtual tape eligible for scratch will be scratched regardless its deferred-scratch status.

NO A virtual tape eligible for scratch will be scratched considering its deferred-scratch status.

Value range: N/A.

Default: NO.

LOCID This parameter is used only when TYPE=VOLSER. When TYPE is not VOLSER, the e-Vault location name is extracted from the tape management system database and this parameter is ignored.

When the LOCID parameter is specified, the virtual tape is not scratched. Rather, it is marked “to be vaulted” or “to be returned” (according to the locid value).

locid Specifies either the local vault location name or a remote off-site storage location name to which this virtual tape is to be moved.

– If locid is similar to the OWNLOCID parameter specified on the VLTOWN initialization statement and the virtual tape currently resides in a remote off-site storage location, the virtual tape is marked “to be returned”.

– If locid is similar to one of the LOCID parameters in the VLTRMT initialization statements and the virtual tape currently resides in the local location, the virtual tape is marked “to be vaulted” to that location.

Value range: A one to eight alphanumeric character string or blanks surrounded by quotes.

Default: None.

9.11.2 Using VTFM with DFSMSrmm

In this section, we describe how you can use DFSMSrmm to update the VTFM VDB.

Using the SEARCHVOLUME subcommand

You can use the RMM TSO SEARCHVOLUME subcommand to create SYNC control statements that can be used to synchronize the VTFM Volume Database (VDB). Figure 9-90 shows you a sample JCL that uses the SEARCHVOLUME subcommand to create control statements.

```
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE RMM.VTFM.SYNC.D2009227 NONVSAM PURGE
 SET MAXCC=0
/*
//SEARCHV  EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//RMMCLIST DD   DISP=(,CATLG),DSN=RMM.VTFM.SYNC.D2009227,
//         UNIT=3390,SPACE=(CYL,(1,1),RLSE),
//         DCB=(LRECL=80,BLKSIZE=0,RECFM=FB,DSORG=PS)
//SYSTSIN  DD   *
 RMM SV VOLUME(VT*) OWNER(*) LIMIT(*) STATUS(SCRATCH) -
     CLIST('SYNC      TYPE=VOLSER,VOLSER=',',FORCE=YES')
/*
```

Figure 9-90 Use SCEARCHVOLUME to create VTFM commands

Now you can use the previously created control cards to update your VTFM Volume Database (VDB) using the JCL shown in Figure 9-91.

```
//USCR     EXEC VTFMSRV,SERVICE=USCR     <== PROCEDURE NAME
//HVTUTIL  DD  DISP=SHR,DSN=RMM.VTFM.SYNC.D2009227
```

Figure 9-91 Sample JCL to update VTFM VDB

EDGHSKP considerations

Use the normal daily EDGHSKP report extract file to update the VTFM volume database. Depending on the VTFM requirement, you have to copy the RPTEXT or XRPTEXT file to a temporary one. If you are using the XRPTEXT, you must be sure that the required V (Volume) records are include.

Figure 9-92 shows you a sample JCL.

Note: The input file to update the VTFM Volume Database (VDB) must be a temporary file.

```
//STEP0001 EXEC PGM=EDGHSKP,
//         PARM=('BACKUP,VRSEL,EXPROC,RPTEXT,DATEFORM(J)')
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   DISP=SHR,DSN=RMM.HSKP.MESSAGE
//BACKUP   DD   DISP=SHR,DSN=RMM.HSKP.CDSBKUP
//JRNLBKUP DD   DISP=SHR,DSN=RMM.HSKP.JNLBKUP
//XREPTEXT DD   DISP=SHR,DSN=RMM.HSKP.EXTRACT
//REPORT   DD   DISP=SHR,DSN=RMM.HSKP.REPORT
//ACTIVITY DD   DISP=SHR,DSN=RMM.HSKP.ACTIVITY
//SYSIN    DD *
 RPTEXT RECORDS(V,X)
//* RPTEXT RECORDS(B,D,X,O,P,R,S)
//STEP0002 EXEC PGM=IEBGENER
//SYSPRINT DD   DUMMY
//SYSUT1   DD   DSN=RMM.HSKP.MESSAGE,DISP=SHR
//SYSUT2   DD   SYSOUT=*
//SYSIN    DD   DUMMY
//*
//STEP0003 EXEC PGM=IEBGENER
//SYSPRINT DD   DUMMY
//SYSUT1   DD   DISP=SHR,DSN=RMM.HSKP.EXTRACT
//SYSUT2   DD   DISP=(,PASS),DSN=&REPORT,UNIT=SYSDA,
//         SPACE=(CYL,(5,5),RLSE),DCB=*.SYSUT1
//SYSIN    DD   DUMMY
//*
//STEP0004 EXEC VTFMSRV,SERVICE=UREP     <== PROCEDURE NAME
//HVTUTIL  DD *
 REPORT  REP=REP001,LINECT=60
/*
//STEP0005 EXEC VTFMSRV,SERVICE=USCR,    <== PROCEDURE NAME
//            COND=(0,NE,STEP0001)
//REPIN    DD DISP=SHR,DSN=&REPORT
//HVTUTIL  DD *
 SYNC      TYPE=RMM
/*
//STEP0006 EXEC VTFMSRV,SERVICE=UREP     <== PROCEDURE NAME
//HVTUTIL  DD *
 REPORT  REP=REP001,LINECT=60
/*
```

Figure 9-92 DFSMSrmm daily housekeeping example

Where:

STEP0001 The normal daily EDGHSKP processing, which uses the following functions:

- Vital record specification processing.
- Expiration processing.

– Report extract processing. If the XRPTEXT file is selected, the volume records must be created too.

– Backing up the DFSMSrmm control data set and journal and clearing the journal.

STEP0002 Copy the message file into the joblog.

STEP0003 Copy the report extract file (XRPTEXT) file to a temporary one. Depending on the reason, VTFM only allows the use of a temporary one to update the VTFM volume database.

STEP0004 Create a VTFM detailed volume list before updating the VTFM volume database.

STEP0005 Update the VTFM volume database based on the DFSMSrmm report extract file.

STEP0006 Create a VTFM detailed volume list after updating the VTFM volume database.

Note: STEP0004 and STEP0006 can be removed if the update of the VTFM Volume database is successfully tested.

Additional information

If you have set up all the parameters and options correctly in your DFSMSrmm's primary tape management system, you should have some other messages (WTORs) that VTFM writes to your automated operator system.

For example, VTFM will stop the processing for a JOB if DFSMSrmm rejects a volume, as shown in Figure 9-93.

```
EDG4027I VOLUME IVP001 REJECTED. IT IS NOT A SCRATCH VOLUME AND MOUNT
         REQUEST WAS NON-SPECIFIC
EDG4005E VOLUME IVP001 ON 040B REJECTED FOR USE BY RES7GENR, IKJEFT01,  SYSUT2
```

Figure 9-93 DFSMSrmm messages EDG4028I and EDG4005E

In this case, VTFM stopped processing for this JOB and writes a WTOR, as shown in Figure 9-94.

```
*619 HVT319W VOLUME=IVP001 REJECTED BY TMS. REPLY R (RETRY), N (FAIL)
```

Figure 9-94 VTFM warning message HVT319W

9.12 DFSMShsm considerations

To use VTFM with DFSMShsm, updates to the ARCCMDnn member in SYS1.PARMLIB are required, as shown in Figure 9-95.

```
/*************************************************************
/* VTFM CHANGES                                2009/07/17 NORBERT *
/*                                                               *
/* PATCH MUST BE ISSUED:                                         *
/*        PATCH PATCH .MCVT.+4B6 X'0A'                           *
/*                                                               *
/* CHANGE ALL 3590-1 UNIT NAMES TO VT3590G2                      *
/* CHANGE THE TAPEUTILIZATION TO 300 PERCENT                     *
/* CHANGE THE TAPEUTILIZATION TO NOLIMIT                         *
/*                                                               *
/*                                                               *
/*                                                               *
/*************************************************************

ARCCMDnn Member
ONLYIF HSMHOST(3)
 SETSYS USERUNITTABLE(VT3590G2)            /* VTFM       NORBERT */

ONLYIF HSMHOST(3)
 SETSYS UNITNAME(VT3590G2)                 /* VTFM       NORBERT */ -
        BACKUP(TAPE(VT3590G2))             /* VTFM       NORBERT */ -
        RECYCLEOUTPUT(BACKUP(VT3590G2))    /* VTFM       NORBERT */

 SETSYS                    /* UTILIZE 97% OF TAPE CARTRIDGE     */ -
   TAPEUTILIZATION(                                                -
    UNITTYPE(VT3590G2) PERCENTFUL(NOLIMIT)) /* VTFM       NORBERT */

....
SETSYS                    /* DIREST DFSMSHSM TO INTIALLY       */ -
  UNITNAME(VT3590G2)      /* SPECIFY A 3590-1 DEVICE FOR       */
                          /* BACKUP OR DUMP SCRATCH TAPES.     */
                                    /* VTFM       NORBERT */
....
PATCH                     /* ALLOW TO RUN AUTO MIGRATION       */ -
  .MCVT.+4B6 X'0A'        /* TO ASSUME 1:1 HARDWARE COMPRESSION*/
                                    /* VTFM       NORBERT */
......
SETSYS                    /* ACTIVATE THE BACKUP AND DUMP      */ -
  BACKUP(TAPE(VT3590G2))  /* FUNCTION OF DFSMSHSM              */
                                    /* VTFM       NORBERT */
```

Figure 9-95 DFSMShsm parmlib changes

10

DFSMSrmm enhancements

The DFSMSrmm enhancements in z/OS V1.11 DFSMS provide improvements in the areas of partitioning your tape volumes and system managed tape libraries, EDGRMMnn parmlib commands and options, fast replication to copy the DFSMSrmm CDS, and enterprise system support.

In this chapter, we discuss the following enhancements:

- EDGINERS Tape Label Scan
- Dynamic Exits Services
- Returning volumes to the system-managed library
- VRSEL GDG option
- DFSMSrmm report generator
- DFSMSrmm usability items
  - Using MATCHVRS in the ISPF data set panel
  - New ADDVOLUME subcommand fields
  - Extended SEARCHVOLUME subcommand
  - VRS location definition handling
  - Rexx variables
  - DFSMSrmm journaling
  - Using the EDGUPDT utility
  - API multi-entry return
- VRSEL(OLD) parmlib option
- Migration considerations
  - Report generator
  - VRSEL GDG options
  - CBRUXVNL user exit
  - Rexx variables
  - Dynamic installation exits
  - VRSEL(OLD) parmlib option
- DFSMSrmm VTFM setup

10.1 EDGINERS Tape Label Scan

The DFSMSrmm tape utility EDGINERS performs initialization and erasure of tapes. For tape label read and display, another utility, such as DITTO / File Manager, must be used. EDGINERS has been updated to support the reading and cross-verification of tape label information with the records defined in the DFSMSrmm control data set. The new SCAN function helps identify and manage tapes from other systems or that are in a problem state. The EDGINERS SCAN function reads the VOL1 and header labels for the first file on the specify volume.

You can SCAN volumes that are:

- SMS managed or not SMS managed
- Known or unknown to RMM
- Of all status values except volumes damaged
- Of all types, including WORM and encrypted tapes

You must have READ authorization to use SCAN.

To use this function, you need READ access to the STGADMIN.EDG.OPERATOR resource defined in the FACILITY RACF class. If this resource is not defined, UPDATE access is given as the default. Use the example shown in Figure 10-1 to define this resource and get the correct access.

```
/* ************************************************************** */
/*   DEFINE RESOURCE STGADMIN.EDG.OPERATOR                        */
/* ************************************************************** */
  RDEFINE FACILITY STGADMIN.EDG.OPERATOR -
          UACC(NONE) OWNER(SYS1   )
     RALT FACILITY STGADMIN.EDG.OPERATOR  -
          GLOBALAUDIT(ALL(UPDATE))
     PERMIT STGADMIN.EDG.OPERATOR  -
            ID(SCHLUM  ) ACC(READ) CLASS(FACILITY)
```

Figure 10-1 Define the STGADMIN.EDG.OPERATOR resource

Note: In order to mount SCRATCH volumes, the status in TCDB is changed from SCRATCH to PRIVATE temporarily and back to SCRATCH after SCAN runs.

Figure 10-2 shows you a sample JCL that is used to scan a tape label. In our case, we have specified to scan two tapes, one without any problems and one with different information between the file sequence number found on tape an the file sequence number stored in DFSMSrmm.

```
//SCAN     EXEC PGM=EDGINERS
//SYSPRINT DD SYSOUT=*
//TAPE     DD UNIT=(unit,,DEFER)
//SYSIN    DD *
  SCAN VOLUME(THS028)
  SCAN VOLUME(THS026)
/*
```

Figure 10-2 Sample JCL for scanning a tape

Where:

SYSPRINT DD Required. It contains all the messages issued by EDGINERS processing.

TAPE DD Not required on system managed, library resident volumes.

SYSIN DD Includes the command request to scan the volume.

At the end of each SCAN request, review the results displayed in the EDG6679I message.

Note: SCAN is not supported in automated processing, so Exec parameters are ignored by the SCAN command.

Use the SYSIN file to select the volumes to be scanned. The correct syntax of the SCAN command is shown in Figure 10-3.

Figure 10-3 SCAN command syntax

Where:

SCAN Use this command in the SYSIN file to select the scan function.

VOLUME This is the only valid operand for SCAN and is required.

In Figure 10-4, you can see the report produced by a scan of a volume and the comparison of the information with the information stored within DFSMSrmm without any differences.

```
1
 EDG6666I MANUAL PROCESSING REQUESTED USING SYSIN FILE COMMANDS
 EDG6679I SCAN RESULTS:
  * * * * Device 0B22, TAPE, VOLSER=THS028
       VOL1 label =          VOL1THS028

------------------------------------------------------------------------------------------------------------------------
 Data set 0001              1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
       HDR1 label =         HDR1RMM.F00001       THS02800010001      00921500000000000000IBM OS/VS 370
       HDR2 label =         HDR2F27920000800OTEST9999/STE00001P   B  206317
                * Tape mark

------------------------------------------------------------------------------------------------------------------------
          LBL volser Dsname                                          Vseq Dseq Crdate   Jobname  Step     RECFM LRECL
BLKSZ
 On volume SL  THS028                                  RMM.F00001 0001 0001 2009/215 TEST9999 STE00001 FB       80
27920
 RMM data  SL  THS028                                  RMM.F00001 0001 0001 2009/215 TEST9999 STE00001 FB       80
27920
 EDG6678I VOLUME THS028 SCAN SUCCESSFUL
 EDG6631I UTILITY EDGINERS COMPLETED WITH RETURN CODE 0
```

Figure 10-4 EDGUTIL output from a scan without any differences

Figure 10-5 shows you a different data set sequence number found on the HDR1 information volume and the data set sequence number stored in the DFSMSrmm control data set for this data set.

```
1
 EDG6666I MANUAL PROCESSING REQUESTED USING SYSIN FILE COMMANDS
 EDG6679I SCAN RESULTS:
  * * * * Device 0B22, TAPE, VOLSER=THS026
       VOL1 label =          VOL1THS026

------------------------------------------------------------------------------------------------------------------------
 Data set 0001              1...5...10...15...20...25...30...35...40...45...50...55...60...65...70...75...80
       HDR1 label =          HDR1VOLUMES.AND.CATLGTHS0280002?Š8      0092170000000000000IBM OS/VS 370
       HDR2 label =          HDR2F000800000001TEST9999/STEP03  P      206317
                * Tape mark

------------------------------------------------------------------------------------------------------------------------
          LBL volser Dsname                                        Vseq Dseq Crdate   Jobname  Step     RECFM LRECL
BLKSZ
 On volume SL  THS026                           VOLUMES.AND.CATLG 0002    8 2009/217 TEST9999 STEP03   F       80
              Mismatch(*)                                                  *
 RMM data  SL  THS026          RMM.F11000.ON.TWO.VOLUMES.AND.CATLG 0002 1000 2009/217 TEST9999 STEP03   F       80
 EDG6683I MISMATCH ON Dseq
 EDG6678I VOLUME THS026 SCAN SUCCESSFUL EDG6631I UTILITY EDGINERS COMPLETED WITH RETURN CODE 0
```

Figure 10-5 An EDGUTIL output from a scan found differences

10.1.1 EDGLABEL started procedure

You have the ability to review the results of the scan on the console and in the system log. To implement this functionality, you have to copy the EDGLABEL procedure from SYS1.SAMPLIB to the SYS1.PROCLIB library. Figure 10-6 shows that sample JCL that you can use to create the procedure.

```
//LABEL   PROC OPT=NOVERIFY,U=3480,SOUT=DUMMY                          §02C
//*
//INIT     EXEC PGM=EDGINERS,PARM='&OPT'
//SYSPRINT DD  &SOUT
//TAPE     DD  UNIT=(&U,,DEFER)                                        §02C
```

Figure 10-6 Recommended EDGLABEL job control language

Start the procedure on any MVS console and answer the WTOR that appears, as shown in Example 10-1.

Example 10-1 Using of the EDGLABEL procedure

```
S EDGLABEL

 EDG6667I MANUAL PROCESSING REQUESTED USING OPERATOR PROMPT FOR COMMANDS
*nnn EDG6626A SPECIFY VOLUME "INIT" "SCAN" OR "ERASE“ COMMAND OR "END"

R nnn,SCAN VOLUME(THS028)
```

The correct syntax for the EDG6626A message is shown in Figure 10-7.

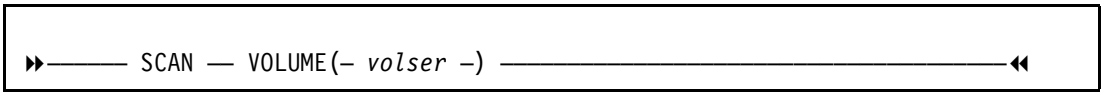


Figure 10-7 EDG6626A command syntax

On your console and in your system log, you should get the information shown in Figure 10-8.

```
*391 EDG6626A SPECIFY VOLUME "INIT" "SCAN" OR "ERASE" COMMAND OR "END"
R 391,SCAN VOLUME(THSO28)
IEE600I REPLY TO 391 IS;SCAN VOLUME(THSO28)
EDG6627A M OB23 V(THSO28) R(THSO28) TO BE SCANNED ,SL
EDG6682I SCAN RESULTS (TRUNCATED): FOR FULL DETAILS - REFER TO
SYSPRINT 862
 * * * * Device OB23, TAPE, VOLSER=THSO28
VOL1THSO28
------------------------------------------------------------------------
-
1...5...10...15...20...25...30...35...40...45...50...55...60...65...70
.
HDR1.BACKTAPE.DATASETTHSO2800010001      009219 99365000000IBM OS/VS
3
HDR2F163841638400DFHSM70 /DFHSM70 P   B  206317
               * Tape mark
------------------------------------------------------------------------
-
LBL volser Dsname                                      Vseq Dseq
Crdat
SL  THSO28                              .BACKTAPE.DATASET 0001 0001
2009/
SL  THSO28                            HSM.BACKTAPE.DATASET 0001 0001
2009/

EDG6675E K OB23 V(THSO28) R()
EDG6678I VOLUME THSO28 SCAN SUCCESSFUL
392 EDG6626A SPECIFY VOLUME "INIT" "SCAN" OR "ERASE" COMMAND OR "END"
```

Figure 10-8 EDGLABEL consol and system log messages

Note: The WTOR Scan command produces a truncated output because of the console message length limits; the complete output is available via EDG6679I in the SYSPRINT file.

EDG6683I MISMATCH ON values

DFSMSrmm issues this message to the operator console and the SYSPRINT file containing the results of the comparison between tape label contents and the information defined to DFSMSrmm during a scan request.

In the message text, *values* is a list of the values that are different between the tape labels and the information defined to DFSMSrmm. The values listed are the column headings in the EDG6679I SCAN RESULTS message:

- LBL
- volser
- Dsname
- Vseq
- Dseq
- Crdate
- Jobname
- Step
- RECFM
- LRECL
- BLKSZ
- Dev

See Table 62, “Label Data Comparison with DFSMSrmm data”, in Chapter 18, “Initializing, Erasing, and Scanning Tape Volumes”, in *z/OS DFSMSrmm Implementation and Customization Guide*, SC26-7405 for details about the compared fields.

System action Processing continues.

Operator response Review the message contents for mismatches between the volume and DFSMSrmm information. To see the complete output, review the contents of the SYSPRINT file. Use a tool such as SDSF to view the output directly on the JES spool.

Storage Administrator Response Review the message contents for mismatches between the volume and DFSMSrmm information.

System programmer response None.

Source DFSMSrmm.

Detecting Module EDGINERS.

Routing Code 3, 5, and 11.

Note: If a scan for a volume not known to RMM is requested. you get the following message:

```
EDG6681I SCAN FOR VOLUME A22261 CONTINUES - VOLUME NOT DEFINED TO DFSMSrmm
```

10.1.2 Label data comparison with DFSMSrmm data

Table 10-1 shows you a comparison of the fields you get from the volume label and header and that you get from DFSMSrmm.

Table 10-1 EDGUTIL scan fields

Label ID	position	Name	EDG6679I heading	SL	AL	RMM extract field name
N/A	N/A	Label type	LBL	Y	Y	RVLABEL
VOL1	5	Volume Serial number	volser	Y	Y	RVVOLSER or RVVOL1 for duplicate volumes

Label ID	position	Name	EDG6679I heading	SL	AL	RMM extract field name
HDR1	5	Data set identifier	Dsname 2	Y	Y	RVDSN1
HDR1	28	Volume sequence number	Vseq	Y	Y	RVVOLSEQ
HDR1	32	Data set sequence number	Dseq	Y	Y	RVLABNO1
HDR1	42	Creation Date Format: cyyddd, entry 0=20	Crdate	Y	Y	RDCRDATE
HDR2	SL(5, 37, 39) AL(5)	Record format	RECFM	Y	Y	RDRECFM
HDR2	6	Block length	BLKSZ	Y	Y	RDBLKSZ
HDR2	11	Record length	LRECL	Y	Y	RDLRECL
HDR2	18	Job identification	Jobname	Y	Y	RDCRTJBN
HDR2	27	Job step identification	Step	Y	Y	RDSTEPNM

For more information about tape label fields, see *z/OS DFSMS Using Magnetic Tapes*, SC26-7412.

10.1.3 An example of using EDGUTIL

In manual processing, you can use multiple SCAN, INIT, and ERASE commands in SYSIN DD with PARM = VERIFY to control the initialization and erasure of previously used tapes. Figure 10-9 shows a sample job.

```
//SCAN     EXEC PGM=EDGINERS,PARM=VERIFY
//SYSPRINT DD SYSOUT=*
//SYSIN    DD *
  SCAN VOLUME(THS001)
  ERASE VOLUME(THS001) LABEL(AL)
  SCAN VOLUME(THS001)
  INIT VOLUME(THS002) LABEL(AL)
  INIT VOLUME(THS003)
/*
```

Figure 10-9 Multiple SCAN, INIT, and ERASE commands

You should see the messages shown in Figure 10-10 if there was no error.

```
EDG6666I MANUAL PROCESSING REQUESTED USING SYSIN FILE COMMANDS
EDG6678I VOLUME THS001 SCAN SUCCESSFUL
EDG6643I VOLUME THS001 ERASED AND LABELLED SUCCESSFULLY WITH VOL1 THS001
EDG6623I VOLUME THS001 ERASE, INITIALIZATION AND VERIFICATION SUCCESSFUL - RETURN TO RACKNUMBER
EDG6678I VOLUME THS001 SCAN SUCCESSFUL
EDG6642I VOLUME THS002 LABELLED SUCCESSFULLY WITH VOL1 THS002
EDG6642I VOLUME THS003 LABELLED SUCCESSFULLY WITH VOL1 THS003
EDG6620I VOLUME THS003 INITIALIZATION AND VERIFICATION SUCCESSFUL - RETURN TO RACK NUMBER
EDG6620I VOLUME THS002 INITIALIZATION AND VERIFICATION SUCCESSFUL - RETURN TO RACK NUMBER
```

Figure 10-10 Multiple SCAN, INIT, and ERASE messages

10.1.4 New messages for SCAN

There are several new messages available for the new EDGUTIL SCAN function:

EDG6676E `USER NOT AUTHORIZED TO REQUEST SCAN FUNCTION`

The user is not authorized to perform the requested function.

EDG6677E `VOLUME A22201 SCAN FAILED`

The scan of the specified volume VOLSER failed.

EDG6678I `VOLUME A22201 SCAN SUCCESSFUL`

DFSMSrmm issues this message for informational purposes only. DFSMSrmm successfully scanned the volume.

EDG6679I `SCAN RESULTS`

DFSMSrmm issues this message, which contains the results of the scan request. System action: Processing continues.

EDG6680E `MISMATCH OF VOLUME STATUS - VOLUME A22201 IS ALREADY PRIVATE IN THE TCDB`

You are scanning the labels of a volume that is either in the scratch status in DFSMSrmm or is not defined to DFSMSrmm but is in the scratch status in the TCDB. EDGINERS attempts to change the volume status in the TCDB to PRIVATE so that it can be mounted for processing. The change use attribute request failed because the volume is already PRIVATE in the TCDB.

EDG6681I `SCAN FOR VOLUME A22201 CONTINUES - VOLUME NOT DEFINED TO DFSMSrmm`

You are scanning the labels of a volume that is not defined to DFSMSrmm.

EDG6682I `SCAN RESULTS (TRUNCATED): FOR FULL DETAILS REFER TO SYSPRINT FILE`

DFSMSrmm issues this multiline message to the operator console containing the results of the scan request.

EDG6683I `MISMATCH ON` *`values`*

DFSMSrmm issues this message to the operator console and `SYSPRINT` file containing the results of the comparison between tape label contents and the information defined to DFSMSrmm during a scan request.

Here are the updated messages to support the SCAN function:

EDG6626A `SPECIFY VOLUME “INIT” “SCAN” OR “ERASE” COMMAND OR “END“`

The DFSMSrmm EDGINERS utility has been started with a request to obtain its information from the console.

EDG6603E `NO VALID “ERASE” “SCAN” OR “INIT” REQUESTS ENTERED`

The SYSIN file provided to program EDGINERS was either empty or contained no valid initialize or erase requests.

10.2 Dynamic Exits Services

DFSMSrmm now uses the Dynamic Exit Services system to manage calls to the installation exits, determine if exit modules exist, and to provide error handling and recovery. The installation exits DFSMSrmm provides are shown in Table 10-2.

Table 10-2 Sample installation exits

Exit name	Default exit routine	SAMPLIB samples provided
EDG_EXIT100	EDGUX100	EDGUX100 and EDGCVRSX
EDG_EXIT200	EDGUX200	EDGUX200
EDG_EXIT300	EDGUX300	EDGUX300

DFSMSrmm installation exit modules can be loaded from any APF authorized library; the default is LINKLST.

Dynamic Exit Services is used to load and activate the default (EDGUXn00) exit module at initialization time. The default exit is activated in first position. All other exit related processing is handled by Dynamic Exit Services. If you do not use the default exit module (EDGUXn00) you must use PROGxx in the z/OS parmlib, the SETPROG operator command, or the CSVDYNEX macro.

Important: When defining the dynamic exits or activating the default exits, and you encounter a problem, the DFSMSrmm startup process provides the operator with a choice of actions:

- CANCEL
- RETRY
- CONTINUE

10.2.1 DFSMSrmm installation exit calls

We have several exit points in allocation and open processing where the EDGUX100 exit module is called to support functions, such as Special Date or Sticky Label. During Inventory Management, EDGUX200 is called before a volume returns to scratch. At one exit point during Open, EDGUX300 provides information about the tape media. See Figure 10-11 to understand how DFSMSrmm calls the exit.

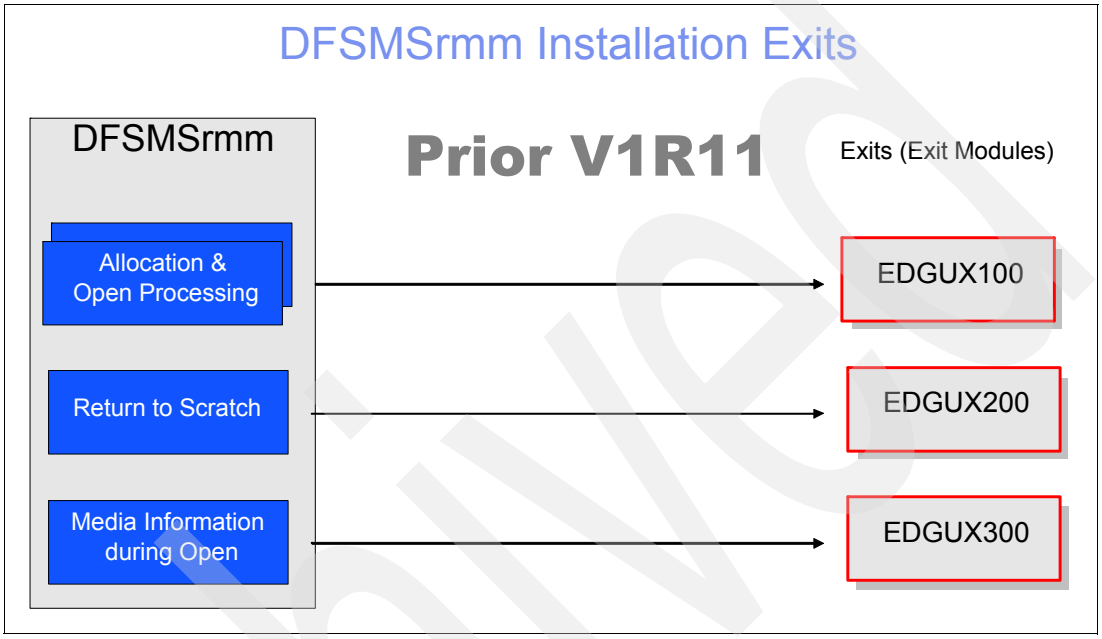


Figure 10-11 Pre-z/OS V1.11 DFSMSrmm installation exit calls

As shown in the left part of the figure, the exits points in DFSMSrmm are still the same. The sample exit modules on the right, which are shipped with DFSMSrmm, (EDGUX100/200/300) have not been changed. The Parameter Lists mapped by EDGPL100/200/300 have not been changed.

Figure 10-12 shows three new exits in the middle (EDG_EXIT100/200/300), which are controlled by Dynamic Exit Services. DFSMSrmm no longer calls the exit modules directly. There is also the new ability to add more exit modules to a given exit.

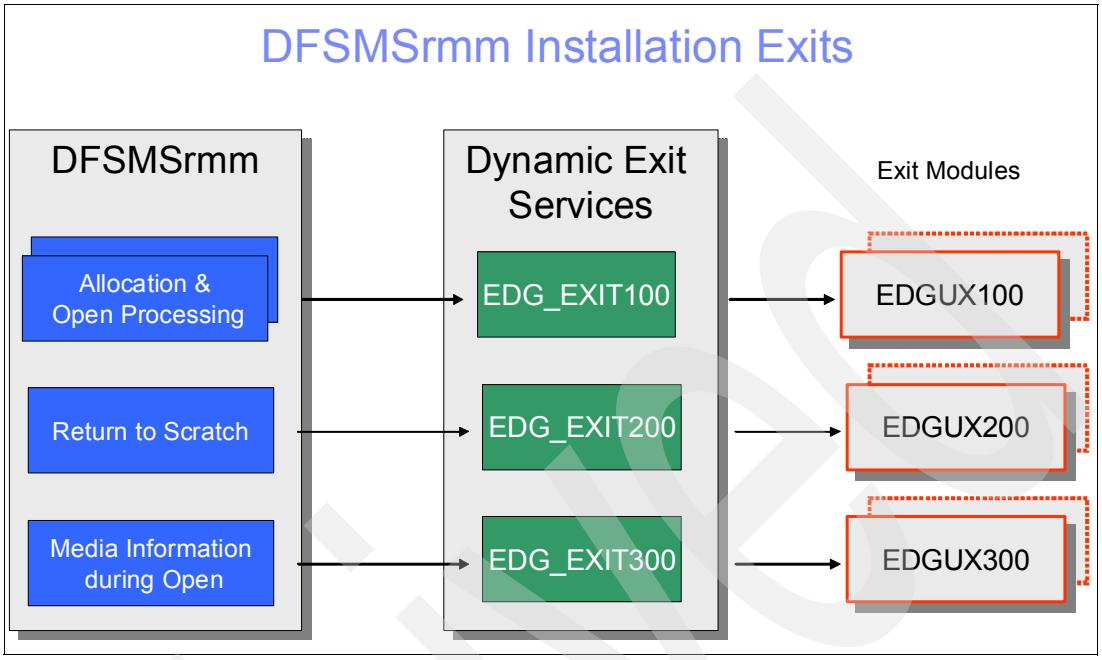


Figure 10-12 z/OS V1.11 DFSMSrmm installation exit calls

Note: The exit points have not changed, but error handling has.

10.2.2 Managing exit modules

Prior to z/OS V1.11, the exit modules could not be managed independently. Also, it was not possible to temporarily inactivate an exit module. Instead, the exit module had to be deleted from LINKLIB before the REFRESH EXITS command was issued. Figure 10-13 shows you the use of the DFSMSrmm REFRESH EXITS command. As you can see, you cannot specify an exit in the command to refresh only one exit.

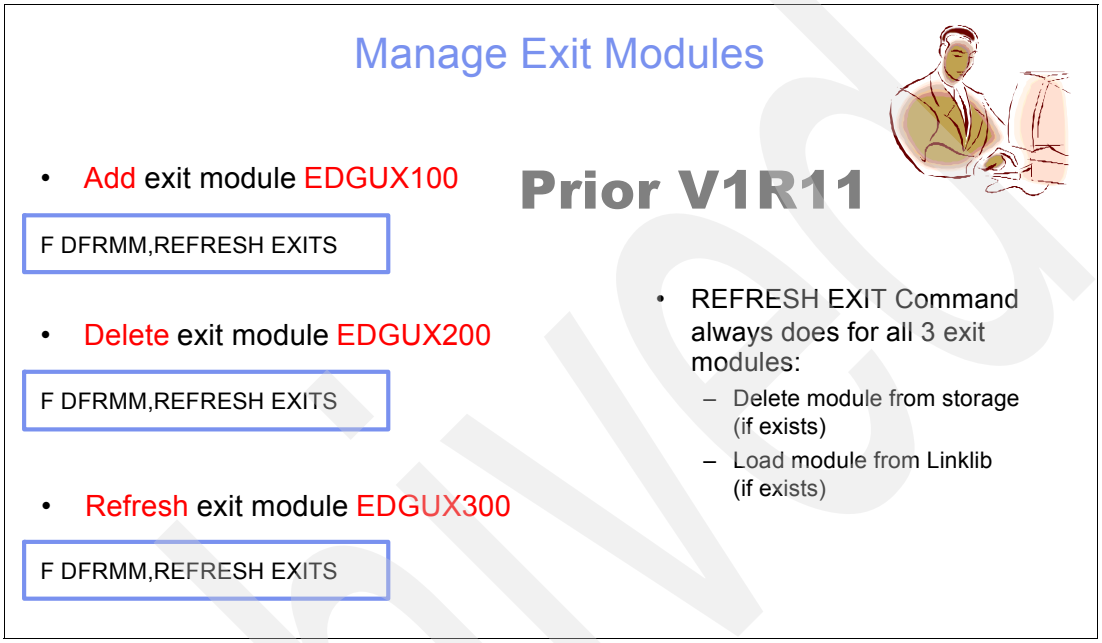


Figure 10-13 Managing exit modules prior to z/OS V1.11

With z/OS V1.11, the MODIFY DFRMM,REFRESH EXIT command is no longer supported. You can use the SETPROG EXIT command instead, which allows you to manage the exit modules independently. Remember that the EXIT statement of the PROGxx parmlib member can be used to add an exit module. In Figure 10-14, you can see how you can refresh a single DFSMSrmm installation exit using the SETPROG operator command.

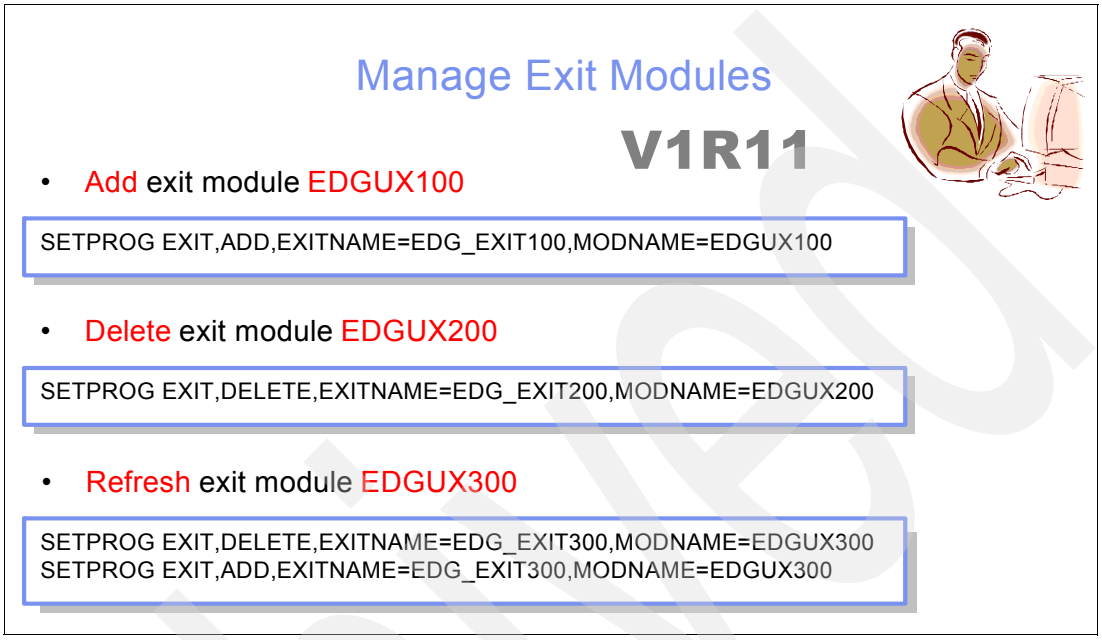


Figure 10-14 Managing exit modules with z/OS V1.11

Besides deleting and adding an exit module, you can also temporarily deactivate it, but keep in mind that setting the state to ACTIVE again does not imply a reload of the module from the library, as you can see in Figure 10-15.

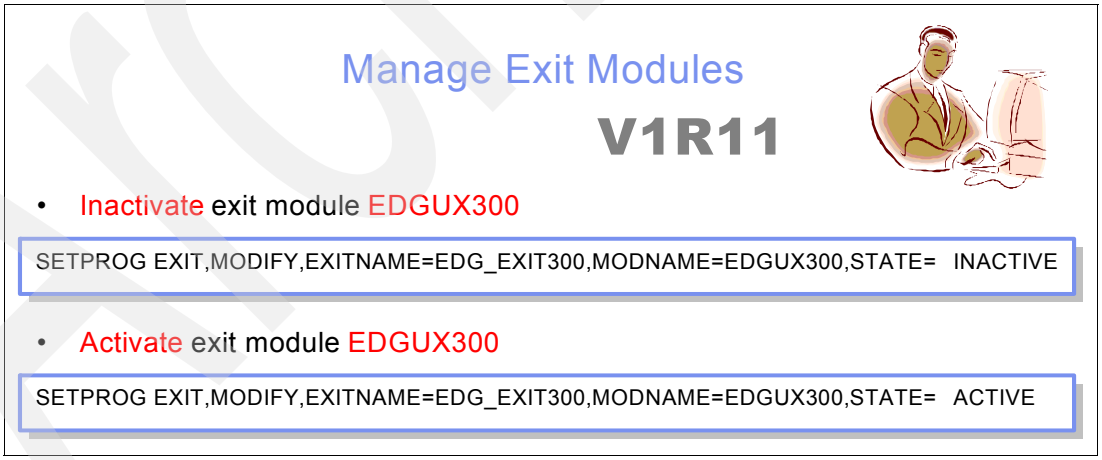


Figure 10-15 Deleting and adding exit modules with z/OS V1.11

Note: Deactivate or activate does not reload the exit module from the library.

Up to V1R11, the Exit status was shown for what are now the exit routines EDGUX100, -200, and -300. Now, the status is shown for the exits EDG_EXIT100, -200, and -300. Nevertheless, the statuses ENABLED, DISABLED, and NONE have not changed. The exact descriptions of the statuses have been adjusted to:

NONE The exit is not defined or no exit modules exist.

ENABLED At least one active exit module exists.

DISABLED One or more exit modules exist but is none are active.

In Figure 10-16, you can see the status of each installation exit if you are using the TSO RMM LISTCONTROL subcommand.

```
Control record:
Type = MASTER    Create date = 2002/064    Create time = 15:46:09
                 Update date = 2009/229    Update time = 20:05:25
Journal: Utilization =   8% (75% threshold)     STATUS: = ENABLED
CDS:     Utilization =  29%
Exit status:                                Options:
  EDG_EXIT100 = ENABLED                       Stacked Volumes        = NONE
  EDG_EXIT200 = NONE                          Extended Bin           = DISABLED
  EDG_EXIT300 = NONE                          Common Time            = ENABLED
                                              CDSID ENQ name         = ENABLED
```

Figure 10-16 LISTCONTROL output

Figure 10-17 shows you the same information when you are using the DFSMSrmm ISPF panel.

```
                      DFSMSrmm Control Record Display
Command ===>

                                                                    More:     +
Type  . . : MASTER      Create date . : 2002/064    Create time . : 15:46:09
                        Update date . : 2009/229    Update time . : 20:05:25

Journal utilization . :   8 %    ( 75 % threshold)  Status : ENABLED
CDS utilization . . . :  29 %

Exit status:                                Options:
EDG_EXIT100 . . : ENABLED                     Stacked volumes . . . : NONE
EDG_EXIT200 . . : NONE                        Extended bin  . . . . : DISABLED
EDG_EXIT300 . . : NONE                        Common time . . . . . : ENABLED
```

Figure 10-17 DFSMSrmm control panel

10.2.3 Setting up the EDG_EXIT100 routine environment

To activate or reactivate the exit routine, you can perform one of the following actions:

- Use dynamic exit services via an operator command.
- Use the CSVDYNEX macro.
- Start DFRMM for the first time since the program was initially loaded.

Note: Stopping and restarting or refreshing the DFSMSrmm procedure does not reload or reactivate your exit routine. See *DFSMSrmm Managing and Using Removable Media*, SC26-7404, for more information.

To install the EDGUX100 default exit routine, perform the following actions to update or replace the default exit module:

1. Build and install an SMP/E USERMOD to apply the updated source code for the EDGUX100 exit module. Include the necessary JCLIN statements to get the EDGUX100 load module added to the LINKLIB target library. You can install the load module by applying the updated source code using an SMP/E USERMOD, as shown in Figure 10-18.

```
//STEP1    EXEC SMPEMVS,REGION=6120K
//SYSIN    DD   *
 SET BDY(GLOBAL) -
 RECEIVE
/*
//SMPPTFIN DD DATA,DLM=##
++USERMOD (VMRMM01) REWORK(1992082) .
++VER (Z038) FMID(HDZ11D0) .
++JCLIN .
//EDGUX100 EXEC PGM=IEWL,
//         PARM='LET,NCAL,RENT,REUS,LIST,XREF,REFR'
//SYSLMOD  DD DSN=SYS1.LINKLIB,DISP=SHR
//SRCLIB   DD DSN=MY.RMM.SRCLIB,DISP=SHR
//AEDGMOD1 DD DSN=SYS1.AEDGMOD1,DISP=SHR
//SYSPRINT DD SYSOUT=*
  INCLUDE AEDGMOD1(EDGUX100)
  ENTRY   EDGUX100
  NAME    EDGUX100(R)
++SRC(EDGUX100) TXLIB(SCRLIB) DISTLIB(ASAMPLIB) .
++SAMP(EDGUX100) TXLIB(SCRLIB) DISTLIB(ASAMPLIB) .
##
/*
```

Figure 10-18 Create a SMP/E USERMOD to apply the updated EDGUX100 installation exit

Modify the FMID and PRE to reflect the release you are running:

a. Allocate a user SAMPLIB data set. In Figure 10-18, the user SAMPLIB data set is defined as MY.RMM.SRCLIB and allocated to DD card SRCLIB.

b. Copy the EDGUX100 source from SAMPLIB to the SAMPLIB user and modify it, as needed, for your installation.

c. SMP/E RECEIVE the USERMOD.

d. SMP/E APPLY the USERMOD. Ensure that a DD card exists for the user SAMPLIB in the APPLY job, or as a DDDEF to SMP/E in the target zone.

After performing these steps, the modified version of the EDGUX100 exit module source code resides in both the SAMPLIB user and SYS1.SAMPLIB. The original copy is only in the distribution libraries at this point. If you accept the USERMOD, only the modified version of the sample exit routine source code exists. The SMP/E target zone reflects RMID indicators of VMRMM01 for all of these records:

- SAMP EDGUX100 RMID=VMRMM01 SYSLIB=SAMPLIB
- SRC EDGUX100 RMID=VMRMM01 SYSLIB=SAMPLIB
- MOD EDGUX100 RMID=VMRMM01 LMOD=EDGUX100
- LMOD EDGUX100 SYSLIB=LINKLIB

The RMID of VMRMM01 for the SAMP record prevents the IBM service from being installed. This results in an ID search and notification to you that the IBM service is the servicing exit source code.

2. Copy the new exit load module into the LNKLST library.
3. Refresh LLA.
4. Refresh the exit by using MVS operator commands.

If DFSMSrmm is running in a shared environment, you must repeat this step on each system.

When you use the default exit module, EDGUX100, for the installation exit EDG_EXIT100, it is loaded and activated as an exit routine, if not already active, by DFSMSrmm when DFSMSrmm is started and stays loaded and active for the life of the session. It can be refreshed/deleted at any time by using one of the MVS system operator commands.

You can use any load module name for your exit module because you use PROGxx in z/OS parmlib, SETPROG, or the CSVDYNEX macro to associate the exit module with the exit. Therefore, you do not have to use EDGUX100 as the load module name. However, using the default exit module name simplifies your implementation because DFSMSrmm itself ensures that the exit module is loaded and activated. Figure 10-19 shows you the correct use of the SETGROG MVS command to add the EDG_EXIT100.

```
SETPROG EXIT,ADD,EXITNAME=EDG_EXIT300,MODNAME=EDGUX300
```

Figure 10-19 SETPROG EXIT ADD syntax

Using the RMM TSO LISTCONTROL subcommand, you can see the current status of the three possible DFSMSrmm installation exits, as shown in Figure 10-20.

```
RMM LICTCONTROL

 Control record:
 Type = MASTER    Create date = 2002/064    Create time = 15:46:09
                  Update date = 2009/235    Update time = 02:00:19
 Journal: Utilization =  46% (75% threshold)     STATUS: = ENABLED
 CDS:     Utilization =  29%
 Exit status:                              Options:
   EDG_EXIT100 = ENABLED                     Stacked Volumes         = NONE
   EDG_EXIT200 = NONE                        Extended Bin            = DISABLED
   EDG_EXIT300 = ENABLED                     Common Time             = ENABLED
                                             CDSID ENQ name          = ENABLED
```

Figure 10-20 RMM TSO LISTCONTROL

10.2.4 Controlling the exit routine through the z/OS dynamic exits facility

IBM has defined the EDG_EXITn00 exit to the dynamic exits facility. You can refer to the exit by the name EDG_EXITn00. You can use the EXIT statement of the PROGxx parmlib member, the SET PROG=xx operator command, the SETPROG EXIT operator command, or the CSVDYNEX macro to control this exit and its exit routines. Figure 10-118 on page 428 shows you a sample of the SETPROG to add the EDGUX300 user exit after a program has been initially loaded.

```
SETPROG EXIT,ADD,EXITNAME=EDG_EXIT300,MODNAME=EDGUX300
```

Figure 10-21 SETPROG EXIT ADD syntax

You can use the ADDABENDNUM and ABENDCONSEC parameters with the CSVDYNEX REQUEST=ADD macro or the ABENDNUM parameter with the SETPROG EXIT operator command to limit the number of times the exit routine abnormally ends before it becomes inactive. An abend is counted when both of the following conditions exist:

- The exit routine does not provide recovery, or the exit routine does provide recovery, but percolates the error.
- The system allows a retry, that is, the recovery routine is entered with the SDWACLUP bit off.

By default, the system disables the exit routines for EDG_EXIT100 and EDG_EXIT300 after one abend. By default, the system does not disable the exit routines for EDG_EXIT200.

You can use the MVS SETPROG command the change the state of an exit routine. In Figure 10-22, we change the status from ACTIVE to INACTIVE.

```
SETPROG EXIT,MODIFY,EXITNAME=EDG_EXIT300,MODNAME=EDGUX300,STATE=INACTIVE
```

Figure 10-22 SETPROG EXIT MODIFY syntax

After you have deactivated the EDG_EXIT300, you can control the result by using the RMM TSO LISTCONTROL command, as shown in Figure 10-23.

```
   RMM LICTCONTROL

Control record:
Type = MASTER    Create date = 2002/064    Create time = 15:46:09
                 Update date = 2009/235    Update time = 02:00:19
Journal: Utilization =  46% (75% threshold)     STATUS: = ENABLED
CDS:     Utilization =  29%
Exit status:                              Options:
  EDG_EXIT100 = ENABLED                     Stacked Volumes         = NONE
  EDG_EXIT200 = NONE                        Extended Bin            = DISABLED
  EDG_EXIT300 = DISABLED                    Common Time             = ENABLED
                                            CDSID ENQ name          = ENABLED
```

Figure 10-23 RMM TSO LISTCONTROL

10.2.5 Deleting the EDG_EXIT100 exit routines

To delete or selectively delete the exit routines for the EDG_EXIT100 exit from the system, you can use MVS operator commands. If DFSMSrmm is running in a shared environment, you must repeat this step on each system. Figure 10-24 shows you the correct use of the MVS SETPROG command to delete the EDG_EXIT300.

```
SETPROG EXIT,DELETE,EXITNAME=EDG_EXIT300,MODNAME=EDGUX300
```

Figure 10-24 SETPROG EXIT DELETE syntax

Now the status for EDG_EXIT300 is NONE and not DISABLED, as you can see in Figure 10-25.

```
   RMM LICTCONTROL

Control record:
Type = MASTER    Create date = 2002/064    Create time = 15:46:09
                 Update date = 2009/235    Update time = 02:00:19
Journal: Utilization =  46% (75% threshold)     STATUS: = ENABLED
CDS:     Utilization =  29%
Exit status:                              Options:
  EDG_EXIT100 = ENABLED                     Stacked Volumes         = NONE
  EDG_EXIT200 = NONE                        Extended Bin            = DISABLED
  EDG_EXIT300 = NONE                        Common Time             = ENABLED
                                            CDSID ENQ name          = ENABLED
```

Figure 10-25 RMM TSO LISTCONTROL

10.2.6 Writing an exit routine for the EDG_EXIT100 exit

EDG_EXIT100 exit routines run in SYSTEM KEY 0 or 5 AMODE(31) RMODE(ANY) in the user's address space. KEY 0 is used when a WTO address or an ACERO address is provided, and KEY 5 is used when a JFCB address is provided. The default exit routine, EXGUX100, is loaded and activated via the CSVDYNEX macro, which is the system's dynamic exits services. Exit modules can be contained in any APF authorized LNKLST library or a APF authorized library named in the CSVDYNEX macro invocation.

10.3 Returning volumes to the system-managed library

The sample volume (CBRUXVNL) that is not in the library installation exit is enhanced so that fewer installations require customization. The idea is that volumes are entered into the tape library whenever possible. Figure 10-26 shows an overview of this task.

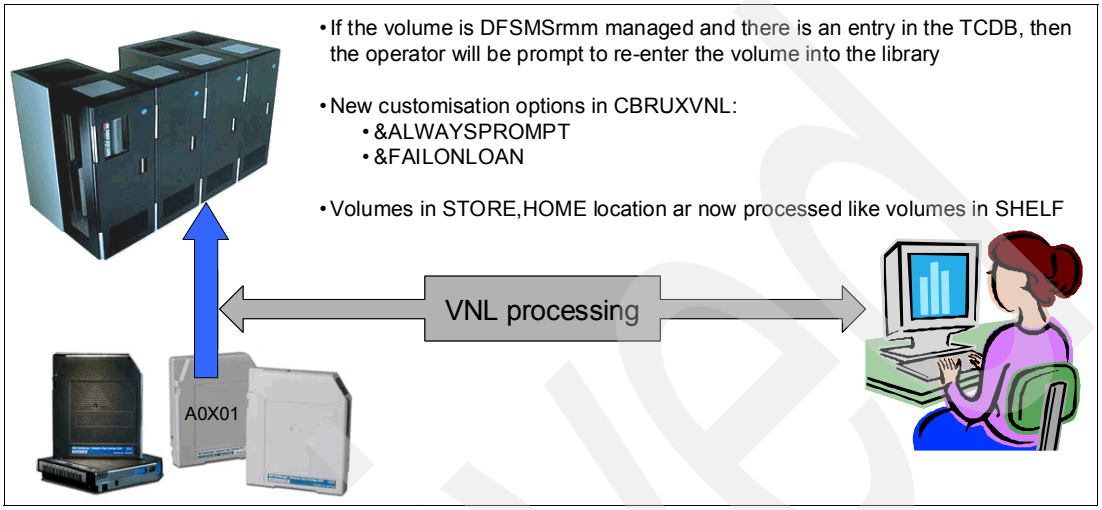


Figure 10-26 CBRUXVNL processing

Additional checks and customization are added to the sample volume that is not in the library installation exit so that there are more cases where a decision can be made. EDG8121D is issued if a Tape Configuration Database (TCDB) volume entry exists, or if the volume is known to be in a different library, but only if the volume is defined to DFSMSrmm. This situation is true as long as the volume is not on loan, and would not be rejected during OPEN processing (that is, the status is pending release, init action is pending, the volume is scratch, or the volume is not intended for use on MVS).

When you regularly store system managed volumes outside of a system-managed library, and also have the home location set to SHELF (or some other LOCDEF-defined storage location with a type of HOME), you can ensure that the volumes will be called by CBRUXVNL by using the SMSTAPE(PURGE(NO)) option. This results in the TCDB volume entry being retained while the volume is ejected. The DFSMSrmm supplied sample exit checks if the TCDB entry exists and, for DFSMSrmm managed volumes, prompts the operator to re-enter the volume into a library.

An alternative to the SMSTAPE(PURGE(NO)) option is to use SMSTAPE(PURGE(ASIS)) and ensure that the library eject default is set to KEEP. This enables you to be selective about the libraries to which the purge/keep option applies. SMSTAPE(PURGE(ASIS)) is the default.

The sample CBRUXVNL exit shown in Figure 10-27 includes simple customization options that you can easily change, with little or no assembler language knowledge, in order to keep the exit matched to your requirements.

```
CBRUXVNL TITLE 'DFSMSRMM CBRUXVNL SAMPLE USER EXIT'
*
&API     SETC  'NO'      Replace 'NO' with 'YES' if API call req'd §08A
*
&TAPEDEC SETC  'NO'      Choose 'YES' or 'NO' (YES is recommended) §09C
...
&FAILONLOAN SETC 'YES'   Choose 'YES' or 'NO'                      §NCA
...
&ALWAYSPROMPT SETC 'NO'  Choose 'YES' or 'NO'                      §NCA
...
```

Figure 10-27 CBRUXVNL provided sample exit

Where:

&API Controls whether the DFSMSrmm API can be used within the system exit to obtain DFSMSrmm information about volumes and data sets.

&TAPEDEC Controls when CBRUXVNL calls DFSMSrmm, based on the tape unit name.

&FAILONLOAN Controls the handling of requests to use volumes that are on loan.

&ALWAYSPROMPT Controls when CBRUXVNL calls DFSMSrmm a second time to request the operator to enter the volume into a system-managed tape library.

For more information, see Chapter 7, “Running DFSMSrmm with System-Managed Tape Libraries”, in the *z/OS V1.11 DFSMSrmm Implementation and Customization Guide*, SC26-7405.

Customize the hardcoded list of tape unit names shown in Example 10-2 if you use other tape unit names in your JCL than the default coded unit names. In this example, we added the tape unit name L3490J to the hard coded list.

Example 10-2 CBRUXVNL tape unit list

```
...
*****************************************************************§SHA
*  Tape units list - may be customized                          §SHA
*****************************************************************§SHA
TAPFIRST DC    CL8'TAPE'                                        §SHA
* insert your location tape unit names here                     §SHA
* or edit any entry                                             §SHA
         DC    CL8'3400-6'                                      §09A
         DC    CL8'3420-8'                                      §09A
         DC    CL8'CART'                                        §09A
         DC    CL8'3480'                                        §SHA
         DC    CL8'3480X'                                       §09A
         DC    CL8'3490'                                        §SHA
         DC    CL8'3490E'                                       §09A
         DC    CL8'3590'                                        §SHA
         DC    CL8'3590-1'                                      §09A
         DC    CL8'3590-E'                                      §09A
```

```
            DC    CL8'3590-B'                                          §09A
            DC    CL8'3590-H'                                          §09A
            DC    CL8'3590L'                                           §09A
            DC    CL8'3592'                                            §09A
            DC    CL8'3592-J'                                          §09A
            DC    CL8'SYS3480R'                                        §09A
            DC    CL8'SYS348XR'                                        §09A
            DC    CL8'VTS'                                             §SHA
            DC    CL8'L3490J'                                          §NSC
            DC    CL8'MTL'                                             §SHA
            DC    CL8' '      any blank entry is ignored               §SHA
TAPLAST     DC    CL8'ATL'                                             §SHA
...
```

The process flow of the sample volume (CBRUXVNL) that is not in the library installation exit is shown in Figure 10-28.

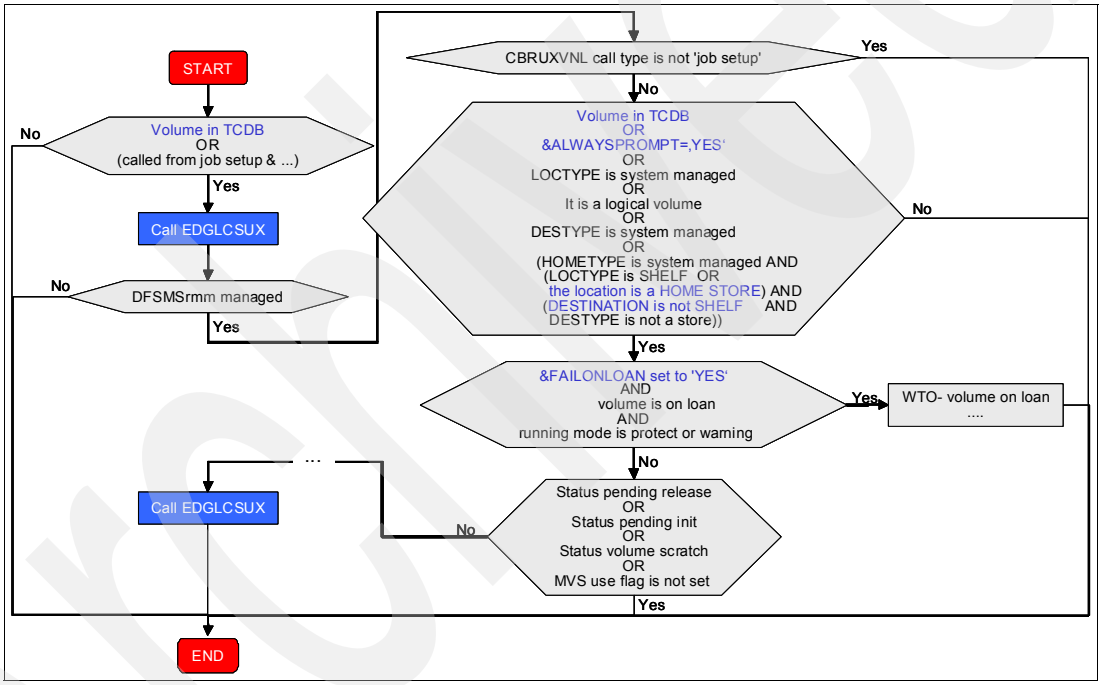


Figure 10-28 CBRUXVNL process flow

10.4 VRSEL GDG option

There are new DFSMSrmm parmlib options that provide flexibility for how tape generation data sets are managed for cyclic retention.

Attention: Non-generation data sets are not managed by this new function.

Use the GDG option to specify how generation data groups are handled for cycle retention by VRSEL processing. Cycle retention includes both the CYCLES and the BYDAYSCYCLE retention types. The correct retention sequence can be set by either the generation number or the creation order.

You can also specify how duplicate generations are handled and have the flexibility to include or exclude duplicates from the cycles count, as required by your application processing.

The correct syntax of the new GDG OPTION command is shown in Figure 10-29.

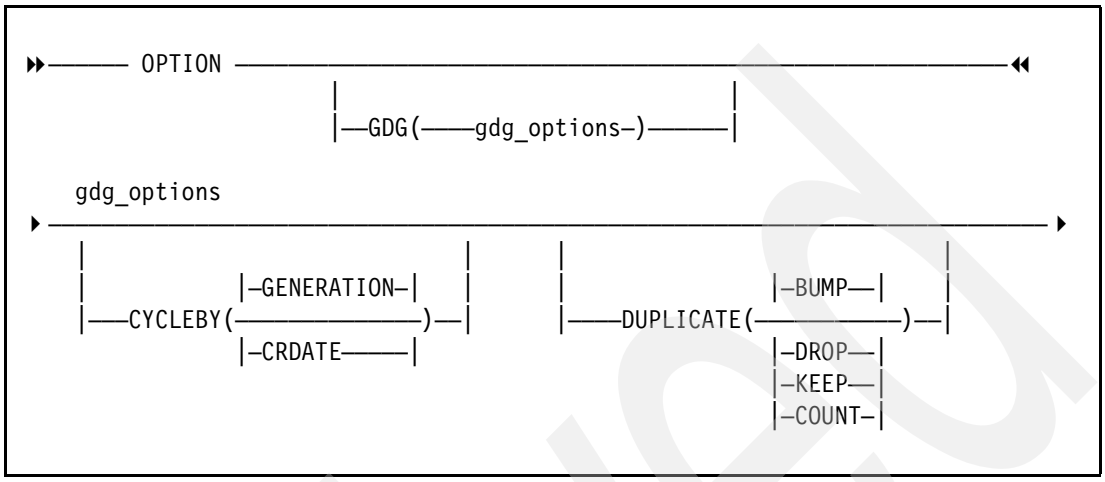


Figure 10-29 GDG option command syntax

Where:

GDG Use the GDG option to specify how generation data groups are handled for cycle retention by VRSEL processing. Cycle retention includes both the CYCLES and the BYDAYSCYCLE retention types. The correct retention sequence can be set by either the generation number or the creation order. You can also specify how duplicate generations (generation data sets) are handled and have the flexibility to include or exclude duplicates from the cycles count, as required by your application processing.

The GDG option has two operands: CYCLEBY and DUPLICATE.

Use the CYCLEBY operand to specify whether retention is based on a generation number or a creation date:

CYCLEBY(GENERATION) Specifies that retention is based on the generation number. DFSMSrmm determines the generation number by applying an algorithm similar to the one used by catalog processing. Both the creation order and the generation number from the data set name are considered, which allows wraps in the generation number to be correctly handled.

CYCLEBY(GENERATION) is the default.

CYCLEBY(CRDATE) Specifies that only the creation sequence be used to determine the retention.

Use the DUPLICATE operand to specify how VRSEL processing handles duplicate generations. You can specify one of the following items:

- Count duplicate generations.
- Keep duplicate generations, but do not count them.
- Bump duplicate generations from the current subchain.
- Drop duplicate generations from VRS retention.

Duplicate generations are determined within a single VRS subchain and only if the generation and the generation it duplicates are not already dropped for another reason. For GENERATION based cycles, the duplicate generations are determined by generation (data set name) and then by creation order. For creation date based cycles, the duplicate generations are determined only by creation order, so they can be detected only when they are created consecutively.

Duplicates are processed within the context of the matching VRS chain or subchain, depending on the DUPLICATE option. DROP is within the context of the VRS chain and all other options are within the context of the subchain. A duplicate generation is considered a duplicate for cycles retention only if the generation it duplicates is retained by the current VRS subchain.

DUPLICATE(BUMP) Specifies that a duplicate generation be bumped by the current VRS subchain and considered for retention by a subsequent VRS subchain.

DUPLICATE(BUMP) is the default.

DUPLICATE(COUNT) Specifies that a duplicate generation be treated as a non-duplicate generation in regards to CYCLE and BYDAYSCYLE processing.

DUPLICATE(DROP) Specifies that a duplicate generation be dropped from VRS retention without further consideration.

DUPLICATE(KEEP) Specifies that a duplicate generation is considered as either the same CYCLE or the same BYDAYSCYCLE depending on the VRS retention type and regardless of the duplicate generation creation date.

Before we look at the different handling options that can be specified, let us discuss the circumstances where VRSEL applies the GDG(DUPLICATE()) operand:

- The VRSEL has the same generation number as the previous VRSEL.

  Consider GDG(CYCLEBY(CRDATE)), where a higher generation might have already been created before the same generation is created again, as shown in Figure 10-30.

Creation Sequence	Data Set Name	Matching VRS	Processing sequence	Duplicate
	A.B.G0003V00	DSN('A.B.*') CYCLES		YES
	A.B.G0003V00	DSN('DELETE') DAYS		NO (3)
	A.B.G0003V00	DSN('A.B.*') CYCLES		NO (1)
	A.B.G0004V00	DSN('A.B.*') CYCLES		NO (1)
	A.B.G0003V00	DSN('A.B.*') CYCLES		NO (1)
	A.B.G0003V00	DSN('DELETE') DAYS		NO (2,3)

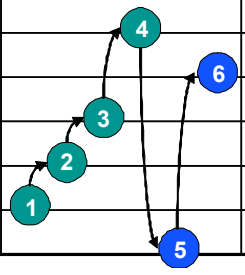


Figure 10-30 Process data set as duplicate generation

- The VRSEL has the same matching VRS as the previous VRSEL (same subchain).

The reason why data sets with the same name and generation number match to a different VRS could be because of one of the following options:

- DELETE VRS, OPEN VRS, ABEND VRS
- VRS with Management Class
- VRS with JOBNAME()

DUPLICATE processing only applies within a single VRS sub chain.

- The VRS retention type is CYCLES or BYDAYSCYCLE.
- The VRSEL met other retention criteria (such as UNTILEXPIRED).

The table in Figure 10-31 shows the different results of the DUPLICATE operands BUMP, DROP, KEEP, and COUNT.

VRS Subchains	GDG DUPLICATE(...)			
	BUMP	DROP	KEEP	COUNT
CYCLES COUNT(2)	A.B.G0003V00 A.B.G0002V00	A.B.G0003V00 A.B.G0002V00	A.B.G0003V00 A.B.G0003V00 A.B.G0002V00	A.B.G0003V00 A.B.G0003V00
DAYS COUNT(100)	A.B.G0003V00			A.B.G0002V00

Figure 10-31 Different results of the DUPLICATE operands BUMP, DROP, KEEP, and COUNT

Let us assume that we have three generation data sets in a cycle group, with the “red” generation 3 being processed as a “duplicate”.

The matching VRS has two subchains, where the first one keeps two cycles, and the second one has a DAYS retention:

BUMP The duplicate is not retained by the CYCLES subchain, but the next subchain retains it (assuming the retention criteria are met).

DROP The duplicate is dropped from the whole VRS chain.

KEEP The duplicate is retained by the CYCLES subchain as part of the same cycle as its original, so the three data sets are kept by COUNT(2).

COUNT The duplicate is retained by the CYCLES subchain, but as its own cycle, so that generation 2 cannot be kept in this subchain anymore.

DUPLICATE processing works basically the same way for CYCLES and BYDAYSCYCLE retention.

If BYDAYSCYCLE is true, and if the duplicate is created on the same day as its original, KEEP and COUNT make no difference. The reason is that when you use BYDAYSCYCLE, data sets created on the same day are treated as one cycle

Use the RMM TSO LISTCONTROL option to determine which parmlib option is currently set. Figure 10-32 shows you the correct use of the command.

```
RMM LISTCONTROL OPTION
```

Figure 10-32 LISTCONTROL OPTION

Figure 10-33 shows the result of the LISTCONTROL command.

```
System options:
PARMLIB Suffix  = CC
Operating mode  = P    Retention period: Default = 0       Maximum = NOLIMIT
                                         Catalog = 6     hours
Control data set name      = RMM.CONTROL.DSET
Journal file data set name = RMM.JOURNAL.DSET
Journal threshold          = 75%               Journal transaction = NO
Catalog SYSID              = Notset
Scratch procedure name     = EDGXPROC
Backup procedure name      = EDGCDSBK
IPL date check  = N        Date format    = J       RACF support    = N
SMF audit       = 248      SMF security   = 249     CDS id          = SC70
MAXHOLD value   = 100      Lines per page = 54      System ID       = SC70
BLP             = RMM      TVEXT purge    = RELEASE Notify          = N
Uncatalog       = Y        VRS job name   = 2       Message case    = M
MASTER overwrite= USER     Accounting     = J       VRS selection   = NEW
VRS change      = INFO     GDG duplicate  = BUMP    GDG cycle by = GENERATION
VRSMIN action   = INFO     VRSMIN count   = 1
VRSDROP action  = INFO     VRSDROP count  = 0               percent = 10
VRSRETAIN action= INFO     VRSRETAIN count= 0               percent = 80
EXPDTDROP action= INFO     EXPDTDROP count= 0               percent = 10
Disp DD name    = DISPDD   Disp msg ID    = EDG4054I
Retain by       = SET      Move by        = SET     CMDAUTH Owner   = NO
PREACS          = NO       SMSACS         = NO      CMDAUTH Dsn     = YES
Reuse bin       = CONFIRMMOVE                       Media name      = 3480
                                                    Local tasks     = 10
PDA: ON
 Block count    = 0        Block size     = 0       Log             = ON
SMSTAPE:
 Update scratch = YES      Update command = YES     Update exits    = YES
 Purge          = ASIS
Client/Server:
 Subsystem type = STANDARD Port           = 0
 Server                    Server tasks   = 0
  host name  =
  IP address =
```

Figure 10-33 Result of LISTCONTROL

In your activity log, you can now find two additional fields, as shown in Figure 10-34, that shows you the active GDG definitions for this run.

```
----+----1----+...---7----+----8----+----9----+----0----+----1----+----2-
H   2009/223  1...SC70    NSS         0 10I         0 80I         0 10IGB
```

Figure 10-34 Activity header record

Where:

GDGCYCLEBY On column 120:

```
ACTRC_HDR_GDGCYCLEBY      DS  C     GDG(CYCLEBY()
ACTRC_HDR_GDGC_GENERATION EQU C'G'
ACTRC_HDR_GDGC_CRDATE     EQU C'C'
```

GDGDUPLICATE On column 121:

```
ACTRC_HDR_GDGDUPLICATE    DS  C     GDG(DUPLICATE()
ACTRC_HDR_GDGD_BUMP       EQU C'B'
ACTRC_HDR_GDGD_DROP       EQU C'D'
ACTRC_HDR_GDGD_KEEP       EQUC'K'
ACTRC_HDR_GDGD_COUNT      EQU C'C'
```

10.4.1 Generation data set and DSN vital record specification creation

To test and show you how this new functionality works, we have created 12 data sets on tape and use these data sets in our test cases. Table 10-3 shows the details of these data sets. We have created each generation data set twice with different creation days. You can also see that the G0003V00 generation data sets are created after the G0004V00 generation data sets.

Table 10-3 Created data set to test the new GDG parmlib option

DSNAME	VOLSER	CRDATE
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00	NS0001	2009/200
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00	NS0011	2009/201
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00	NS0002	2009/207
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00	NS0012	2009/208
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00	NS0003	2009/221
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00	NS0013	2009/222
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00	NS0004	2009/214
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00	NS0014	2009/215
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00	NS0005	2009/228
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00	NS0015	2009/229
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00	NS0006	2009/235
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00	NS0016	2009/236

We have a DSNAME vital record specification that use the attributes shown in Figure 10-35. These attributes match the data sets shown in Table 10-3 on page 360.

```
                        DFSMSrmm Display Data Set VRS
Command ===>

Data set mask . : 'RMM.TEST.NEW.GDG.PARMLIB.OPTION.**'          GDG . : NO
     Job name mask . :

Count . . . : 5                     Retention type  . . . . . : CYCLES
                                    While cataloged . . . . . : NO
Delay . . . : 0   Days              Until expired . . . . . . : NO

Location  . . . . . : HOME
Number in location  : 5
Priority  . . . . . : 0
                                    Release options:
Next VRS in chain . : GDGOPT        Expiry date ignore  . . . : NO
    Chain using . . :               Scratch immediate . . . . : NO

Owner . . . . . : SCHLUM
Description . . :
Delete date . . : 1999/365    ( YYYY/DDD )
```

Figure 10-35 DSNAME vital record specification to test the new GDG parmlib option

To better show the differences, we add a management value/management class vital record specification. Figure 10-36 shows the details of this additional DSNAME vital record specification.

```
                        DFSMSrmm Display Name VRS
Command ===>

Name  . . . . . : GDGOPT


Count . . . : 6                  Retention type  . . . . . : CYCLES
                                 While cataloged . . . . . : NO
                                 Until expired . . . . . . : NO

Location  . . . . . : HOME
Number in location  : 365


Next VRS in chain . :
    Chain using . . :

Owner . . . . . : MHLRES7
Description . . : TEST NEW GDG PARMLIB OPTION
Delete date . . : 1999/365    ( YYYY/DDD )
```

Figure 10-36 Management value / management class VRS definition

10.4.2 CYCLEBY GENERATION and DUPLICATE BUMP

Figure 10-37 shows the normal GDG processing, where it matched the GDG(CYCLEBY(GENERATION) DUPLICATE(BUMP)) GDG parmlib option, which is the default if you have not specified the GDG option in your EDGRMMnn parmlib. You can also specify CYCLEBY(CRDATE) to fulfil the old processing if you have created the data sets in sequence.

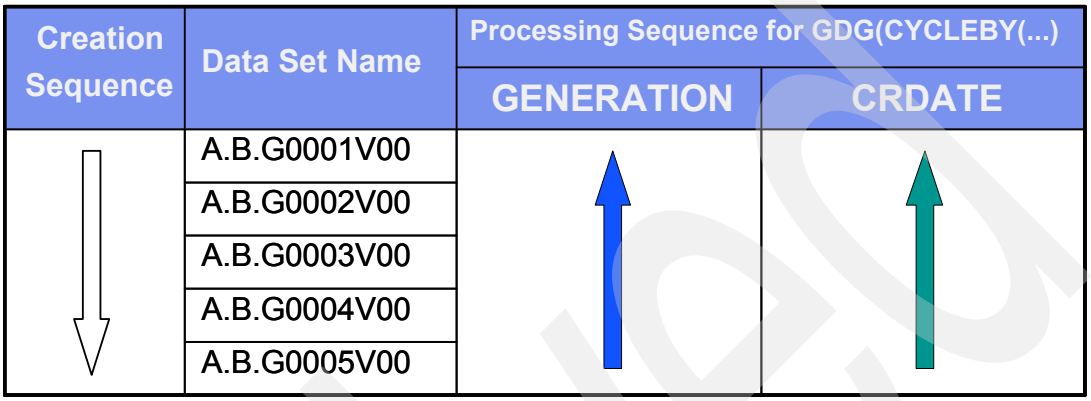


Figure 10-37 Generations created consecutively

In our first test case, we create 12 files and specify a DSNAME vital record specification with the CYCLES option and a count of 5. Figure 10-38 shows the detail DSNAME VRS options.

```
REMOVABLE MEDIA MANAGER                    VITAL RECORDS RETENTION REPORT                                    PAGE          1
    Copyright IBM Corp. 1993,2007          ----- ------- --------- ------                                TIME 14:07:04 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK                OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF
_____________________________________________________________________________________________________________________________
          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**   SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME           2009/223
          GDGOPT                               MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME           2009/223
JOB NAME  DATA SET NAME                          FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME
_____________________________________________________________________________________________________________________________
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0006    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0005    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0004    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0003    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0002    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                      11     0
```

Figure 10-38 Results of test case one

We have created each generation data set twice to show you that the actual generation data set is retained, but the older one is not retained. Table 10-3 on page 360 shows you the data sets we created and Figure 10-38 shows you how a generation data set is retained and how it is not retained. As you can see, only one instance of the same generation data set is retained.

Table 10-4 shows the drop reason and why a data set is retained.

Table 10-4 Details of test case one

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G0001V00	NS0001	2009/200	No	12	*Cycles exceeded*	*Duplicate GDS*
RMM.....G0001V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 6
RMM.....G0002V00	NS0002	2009/207	Next	10	*Duplicate GDS*	CYCL/00006 5
RMM.....G0002V00	NS0012	2009/208	Prim	9	CYCL/00005 5	
RMM.....G0003V00	NS0003	2009/221	Next	8	*Duplicate GDS*	CYCL/00006 4
RMM.....G0003V00	NS0013	2009/222	Prim	7	CYCL/00005 4	
RMM.....G0004V00	NS0004	2009/214	Next	6	*Duplicate GDS*	CYCL/00006 3
RMM.....G0004V00	NS0014	2009/215	Prim	5	CYCL/00005 3	
RMM.....G0005V00	NS0005	2009/228	Next	4	*Duplicate GDS*	CYCL/00006 2
RMM.....G0005V00	NS0015	2009/229	Prim	3	CYCL/00005 2	
RMM.....G0006V00	NS0006	2009/235	Next	2	*Duplicate GDS*	CYCL/00006 1
RMM.....G0006V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition.

10.4.3 CYCLEBY CRDATE and DUPLICATE BUMP

If generation data sets are created out of sequence, you can use the GDG(CYCLEBY(CRDATE)) operand of the parmlib OPTION command to specify whether VRSEL should process the data sets in generation sequence from highest number to lowest, or you can specify GDG(CYCLEBY(CRDATE)) to process the data sets in creation date sequence starting from the youngest. Figure 10-39 shows you the sequence in which the data sets are created.

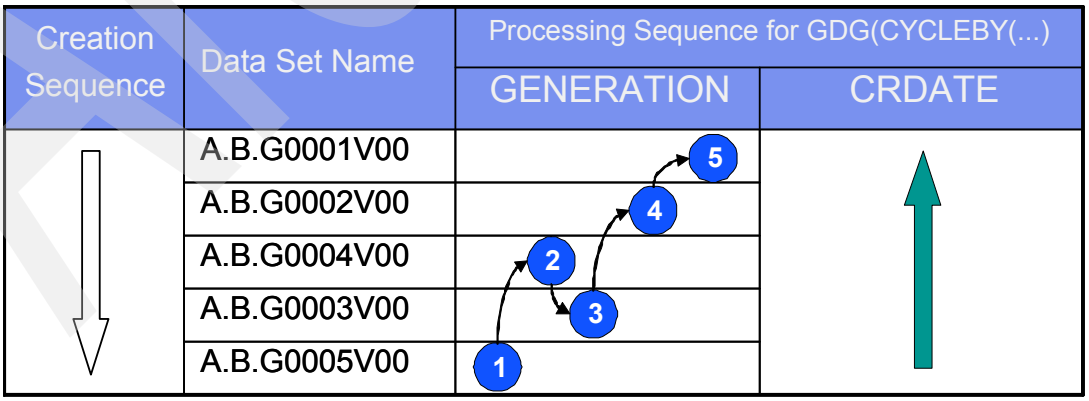


Figure 10-39 Generations out of sequence

In our second test case, we used the same 12 files as before, but we changed the GDG option from CYCLESBY(GENERATION) to CYCLESBY(CRDATE). Figure 10-40 shows you the result of our test.

```
REMOVABLE MEDIA MANAGER                    VITAL RECORDS RETENTION REPORT                                        PAGE          1
    Copyright IBM Corp. 1993,2007          ----- ------- --------- ------                                TIME 14:09:45 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF

          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**   SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME           2009/223
          GDGOPT                               MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME           2009/223
JOB NAME  DATA SET NAME                          FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME

TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0006    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0005    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0003    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0004    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0002    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                      11     0
```

Figure 10-40 Results of test case two

Table 10-5 shows the drop reason and why a data set is retained. The G0003V00 generation data set is retained before the G0004V00 generation data set, depending on the creation date.

Table 10-5 Details of test case two

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G0001V00	NS0001	2009/200	No	12	*Cycles exceeded*	*Duplicate GDS*
RMM.....G0001V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 6
RMM.....G0002V00	NS0002	2009/207	Next	10	*Duplicate GDS*	CYCL/00006 5
RMM.....G0002V00	NS0012	2009/208	Prim	9	CYCL/00005 5	
RMM.....G0003V00	NS0003	2009/221	Next	*6*	*Duplicate GDS*	CYCL/00006 3
RMM.....G0003V00	NS0013	2009/222	Prim	*5*	CYCL/00005 3	
RMM.....G0004V00	NS0004	2009/214	Next	*8*	*Duplicate GDS*	CYCL/00006 4
RMM.....G0004V00	NS0014	2009/215	Prim	*7*	CYCL/00005 4	
RMM.....G0005V00	NS0005	2009/228	Next	4	*Duplicate GDS*	CYCL/00006 2
RMM.....G0005V00	NS0015	2009/229	Prim	3	CYCL/00005 2	
RMM.....G0006V00	NS0006	2009/235	Next	2	*Duplicate GDS*	CYCL/00006 1
RMM.....G0006V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition.

10.4.4 CYCLEBY GENERATION and DUPLICATE BUMP with generation wrap

We simulate the generation wrap up, which means that we have reached the maximum count of 9999 of a generation data set and the next generation is created.

In our third test case, we create 12 files similar to the files we created before, as shown in Table 10-6, but we started with the G9997V00 generation data set so that we have a generation wrap. Figure 10-35 on page 361 shows the detail DSNAME VRS options, and Figure 10-36 on page 361 shows the NAME VRS details.

Table 10-6 Created data set with a generation wrap to test new GDG parmlib option

DSNAME	VOLSER	CRDATE
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9997V00	NS0001	2009/200
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9997V00	NS0011	2009/201
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9998V00	NS0002	2009/207
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9998V00	NS0012	2009/208
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9999V00	NS0003	2009/221
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9999V00	NS0013	2009/222
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00	NS0004	2009/214
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00	NS0014	2009/215
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00	NS0005	2009/228
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00	NS0015	2009/229
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00	NS0006	2009/235
RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00	NS0016	2009/236

Figure 10-41 shows a generation wrap, which means that you have reached the maximum of 9999 for the generation data set, and the next created data set receives the number 0001.

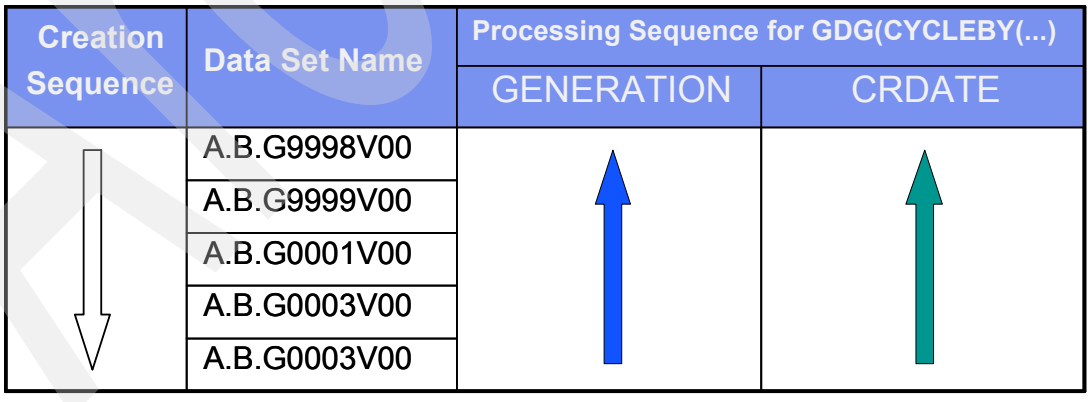


Figure 10-41 Generation wrap

As with the first test case, only the actual generation data set is retained by the primary VRS definition. Figure 10-42 shows the data set retained in this test case.

```
REMOVABLE MEDIA MANAGER                    VITAL RECORDS RETENTION REPORT                                    PAGE          1
    Copyright IBM Corp. 1993,2007          ----- ------- --------- ------                          TIME 14:08:14 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF

          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**   SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME          2009/223
          GDGOPT                               MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME          2009/223
JOB NAME  DATA SET NAME                          FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME

TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0006    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0005    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0004    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9999V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9999V00  1     1 NS0003    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9998V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9998V00  1     1 NS0002    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9997V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                      11     0
```

Figure 10-42 Results of test case three

Table 10-7 shows the drop reason and why a data set is retained.

Table 10-7 Details of test case three

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G9997V00	NS0001	2009/200	No	12	*Cycles exceeded*	*Duplicate GDS*
RMM.....G9997V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 6
RMM.....G9998V00	NS0002	2009/207	Next	10	*Duplicate GDS*	CYCL/00006 5
RMM.....G9998V00	NS0012	2009/208	Prim	9	CYCL/00005 5	
RMM.....G9999V00	NS0003	2009/221	Next	8	*Duplicate GDS*	CYCL/00006 4
RMM.....G9999V00	NS0013	2009/222	Prim	7	CYCL/00005 4	
RMM.....G0001V00	NS0004	2009/214	Next	6	*Duplicate GDS*	CYCL/00006 3
RMM.....G0001V00	NS0014	2009/215	Prim	5	CYCL/00005 3	
RMM.....G0001V00	NS0005	2009/228	Next	4	*Duplicate GDS*	CYCL/00006 2
RMM.....G0001V00	NS0015	2009/229	Prim	3	CYCL/00005 2	
RMM.....G0001V00	NS0006	2009/235	Next	2	*Duplicate GDS*	CYCL/00006 1
RMM.....G0001V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition.

10.4.5 CYCLEBY CRDATE and DUPLICATE BUMP with generation wrap

In our fourth test case, we used the same 12 files as before, but we changed the GDG option from CYCLESBY(GENERATION) to CYCLESBY(CRDATE). Figure 10-43 shows that the G9999V00 generation data set is retained before the G0001V00 generation data set, depending on the creation date.

```
REMOVABLE MEDIA MANAGER                    VITAL RECORDS RETENTION REPORT                                    PAGE          1
    Copyright IBM Corp. 1993,2007          ----- ------- --------- ------                          TIME 14:08:14 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF

          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**   SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME          2009/223
          GDGOPT                               MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME          2009/223
JOB NAME  DATA SET NAME                          FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME

TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0006    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0005    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9999V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9999V00  1     1 NS0003    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0004    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9998V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9998V00  1     1 NS0002    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G9997V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                        11     0
```

Figure 10-43 Results of test case four

Table 10-8 shows the drop reason and why a data set is retained. The G9999V00 generation data set is retained before the G0001V00 generation data set, depending on the creation date.

Table 10-8 Details of test case four

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G9997V00	NS0001	2009/200	No	12	*Cycles exceeded*	*Duplicate GDS*
RMM.....G9997V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 6
RMM.....G9998V00	NS0002	2009/207	Next	10	*Duplicate GDS*	CYCL/00006 5
RMM.....G9998V00	NS0012	2009/208	Prim	9	CYCL/00005 5	
RMM.....G9999V00	NS0003	2009/221	Next	6	*Duplicate GDS*	CYCL/00006 3
RMM.....G9999V00	NS0013	2009/222	Prim	5	CYCL/00005 3	
RMM.....G0001V00	NS0004	2009/214	Next	8	*Duplicate GDS*	CYCL/00006 4
RMM.....G0001V00	NS0014	2009/215	Prim	7	CYCL/00005 4	
RMM.....G0001V00	NS0005	2009/228	Next	4	*Duplicate GDS*	CYCL/00006 2
RMM.....G0001V00	NS0015	2009/229	Prim	3	CYCL/00005 2	
RMM.....G0001V00	NS0006	2009/235	Next	2	*Duplicate GDS*	CYCL/00006 1
RMM.....G0001V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition.

10.4.6 CYCLEBY GENERATION and DUPLICATE DROP

In our fifth test, we change the GDG parmlib option to GDG(CYCLEBY(GENERATION) DUPLICATE(DROP)). Figure 10-44 shows that only the newer version of each generation data set is retained. Depending on the DROP option we have specified, the older versions are dropped when the cycles are exceeded.

```
REMOVABLE MEDIA MANAGER                    VITAL RECORDS RETENTION REPORT                                    PAGE          1
    Copyright IBM Corp. 1993,2007          ----- ------- --------- ------                          TIME 14:09:45 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF

          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**   SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME          2009/223
          GDGOPT                               MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME          2009/223
JOB NAME  DATA SET NAME                          FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME

TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                                 6      0
```

Figure 10-44 Results of test case five

Table 10-9 shows the drop reason and why a data set is retained.

Table 10-9 Details of test case five

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G0001V00	NS0001	2009/200	No	12	*Cycles exceeded*	*Duplicate GDS*
RMM.....G0001V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 1
RMM.....G0002V00	NS0002	2009/207	No	10	*Duplicate GDS*	
RMM.....G0002V00	NS0012	2009/208	Prim	9	CYCL/00005 5	
RMM.....G0003V00	NS0003	2009/221	No	8	*Duplicate GDS*	
RMM.....G0003V00	NS0013	2009/222	Prim	7	CYCL/00005 4	
RMM.....G0004V00	NS0004	2009/214	Not	6	*Duplicate GDS*	
RMM.....G0004V00	NS0014	2009/215	Prim	5	CYCL/00005 3	
RMM.....G0005V00	NS0005	2009/228	No	4	*Duplicate GDS*	
RMM.....G0005V00	NS0015	2009/229	Prim	3	CYCL/00005 2	
RMM.....G0006V00	NS0006	2009/235	No	2	*Duplicate GDS*	
RMM.....G0006V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition.

10.4.7 CYCLEBY CRDATE and DUPLICATE DROP

In our sixth test, we change the GDG parmlib option to GDG(CYCLEBY(GENERATION) DUPLICATE(DROP)). Figure 10-45 shows that only the newer version of each generation data set is retained on the DROP specification. Depending on the DROP option we have specified, the older versions are dropped when the cycles are exceeded, but generation data set 4 will be retained before generation data set 3.

```
REMOVABLE MEDIA MANAGER                    VITAL RECORDS RETENTION REPORT                                    PAGE          1
    Copyright IBM Corp. 1993,2007          ----- ------- --------- ------                          TIME 14:09:45 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF

          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**   SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME           2009/223
          GDGOPT                               MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME           2009/223
JOB NAME  DATA SET NAME                          FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME

TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                                 6     0
```

Figure 10-45 Results of test case six

Table 10-10 shows the drop reason and why a data set is retained. The G0003V00 generation data set is retained before the G0004V00 generation data set, depending on the creation date.

Table 10-10 Details of test case six

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G0001V00	NS0001	2009/200	No	12	*Cycles exceeded*	*Duplicate GDS*
RMM.....G0001V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 1
RMM.....G0002V00	NS0002	2009/207	No	10	*Duplicate GDS*	D
RMM.....G0002V00	NS0012	2009/208	Prim	9	CYCL/00005 5	
RMM.....G0003V00	NS0003	2009/221	No	*6*	*Duplicate GDS*	
RMM.....G0003V00	NS0013	2009/222	Prim	*5*	CYCL/00005 4	
RMM.....G0004V00	NS0004	2009/214	No	*8*	*Duplicate GDS*	
RMM.....G0004V00	NS0014	2009/215	Prim	*7*	CYCL/00005 3	
RMM.....G0005V00	NS0005	2009/228	No	4	*Duplicate GDS*	
RMM.....G0005V00	NS0015	2009/229	Prim	3	CYCL/00005 2	
RMM.....G0006V00	NS0006	2009/235	No	2	*Duplicate GDS*	
RMM.....G0006V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition.

10.4.8 CYCLEBY GENERATION and DUPLICATE KEEP

In our seventh test, we change the GDG parmlib option to GDG(CYCLEBY(GENERATION) DUPLICATE(KEEP)). Figure 10-46 shows that both versions of the current five generation data sets are retained by the primary VRS definition and that both versions of the oldest generation data sets are retained by the next VRS definition GDGOPT.

```
REMOVABLE MEDIA MANAGER                  VITAL RECORDS RETENTION REPORT                                    PAGE          1
    Copyright IBM Corp. 1993,2007        ----- ------- --------- ------                        TIME 14:09:18 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF
_____________________________________________________________________________________________________________________
          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**    SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME           2009/223
          GDGOPT                                MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME           2009/223
JOB NAME  DATA SET NAME                         FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME
_____________________________________________________________________________________________________________________
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0006    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0005    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0004    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0003    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0002    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0001    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                      12     0
```

Figure 10-46 Results of test case seven

Table 10-11 shows that all 12 data sets are retained.

Table 10-11 Details of test case seven

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G0001V00	NS0001	2009/200	Next	12	*Cycles exceeded*	CYCL/00006 1
RMM.....G0001V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 1
RMM.....G0002V00	NS0002	2009/207	Next	10	CYCL/00005 5	
RMM.....G0002V00	NS0012	2009/208	Prim	9	CYCL/00005 5	
RMM.....G0003V00	NS0003	2009/221	Prim	8	CYCL/00005 4	
RMM.....G0003V00	NS0013	2009/222	Prim	7	CYCL/00005 4	
RMM.....G0004V00	NS0004	2009/214	Prim	6	CYCL/00005 3	
RMM.....G0004V00	NS0014	2009/215	Prim	5	CYCL/00005 3	
RMM.....G0005V00	NS0005	2009/228	Prim	4	CYCL/00005 2	
RMM.....G0005V00	NS0015	2009/229	Prim	3	CYCL/00005 2	
RMM.....G0006V00	NS0006	2009/235	Prim	2	CYCL/00005 1	
RMM.....G0006V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition

10.4.9 CYCLEBY CREATION and DUPLICATE KEEP

In our eight test, we change the GDG parmlib option to GDG(CYCLEBY(CREATION) DUPLICATE(KEEP)). Figure 10-47 shows that both versions of the last five generation data sets are retained by the primary VRS definition and that both versions of the oldest generation data sets are retained by the next VRS definition GDGOPT, but generation data set 4 will be retained before generation data set 3, depending on the creation date.

```
REMOVABLE MEDIA MANAGER                  VITAL RECORDS RETENTION REPORT                                    PAGE          1
    Copyright IBM Corp. 1993,2007        ----- ------- --------- ------                        TIME 14:09:18 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF

          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**    SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME          2009/223
          GDGOPT                                MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME          2009/223
JOB NAME  DATA SET NAME                         FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME

TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0006    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0005    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0003    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0004    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0002    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0001    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                      12     0
```

Figure 10-47 Results of test case eight

Table 10-12 shows that all 12 created data sets are retained. The G0003V00 generation data set is retained before the G0004V00 generation data set, depending on the creation date.

Table 10-12 Details of test case eight

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G0001V00	NS0001	2009/200	Next	12	*Cycles exceeded*	CYCL/00006 1
RMM.....G0001V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 1
RMM.....G0002V00	NS0002	2009/207	Prim	10	CYCL/00005 5	
RMM.....G0002V00	NS0012	2009/208	Prim	9	CYCL/00005 5	
RMM.....G0003V00	NS0003	2009/221	Prim	6	CYCL/00005 3	
RMM.....G0003V00	NS0013	2009/222	Prim	5	CYCL/00005 3	
RMM.....G0004V00	NS0004	2009/214	Prim	8	CYCL/00005 4	
RMM.....G0004V00	NS0014	2009/215	Prim	7	CYCL/00005 4	
RMM.....G0005V00	NS0005	2009/228	Prim	4	CYCL/00005 2	
RMM.....G0005V00	NS0015	2009/229	Prim	3	CYCL/00005 2	
RMM.....G0006V00	NS0006	2009/235	Prim	2	CYCL/00005 1	
RMM.....G0006V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition

10.4.10 CYCLEBY GENERATION and DUPLICATE COUNT

In our ninth test, we change the GDG parmlib option to GDG(CYCLEBY(GENERATION) DUPLICATE(COUNT)). Figure 10-48 shows that the newest five data sets are retained by the primary VRS definition, an additional six versions of the next current created generation data sets are retained by the next VRS definition GDGOPT, and the oldest generation data set is dropped when the cycles are exceeded.

```
REMOVABLE MEDIA MANAGER                  VITAL RECORDS RETENTION REPORT                                    PAGE          1
    Copyright IBM Corp. 1993,2007        ----- ------- --------- ------                        TIME 14:09:18 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF

          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**    SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME           2009/223
          GDGOPT                                MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME           2009/223
JOB NAME  DATA SET NAME                         FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME

TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0006    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0005    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0004    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0003    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0002    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                      11     0
```

Figure 10-48 Results of test case nine

Table 10-13 shows that 11 data sets are retained.

Table 10-13 Details of test case nine

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G0001V00	NS0001	2009/200	No	12	*Cycles exceeded*	*Cycles exceeded*
RMM.....G0001V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 6
RMM.....G0002V00	NS0002	2009/207	Next	10	*Cycles exceeded*	CYCL/00006 5
RMM.....G0002V00	NS0012	2009/208	Next	9	*Cycles exceeded*	CYCL/00006 4
RMM.....G0003V00	NS0003	2009/221	Next	8	*Cycles exceeded*	CYCL/00006 3
RMM.....G0003V00	NS0013	2009/222	Next	7	*Cycles exceeded*	CYCL/00006 2
RMM.....G0004V00	NS0004	2009/214	Next	6	*Cycles exceeded*	CYCL/00006 1
RMM.....G0004V00	NS0014	2009/215	Prim	5	CYCL/00005 5	
RMM.....G0005V00	NS0005	2009/228	Prim	4	CYCL/00005 4	
RMM.....G0005V00	NS0015	2009/229	Prim	3	CYCL/00005 3	
RMM.....G0006V00	NS0006	2009/235	Prim	2	CYCL/00005 2	
RMM.....G0006V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition.

10.4.11 CYCLEBY CREATION and DUPLICATE COUNT

In our tenth test, we change the GDG parmlib option to GDG(CYCLEBY(GENERATION) DUPLICATE(COUNT)). Figure 10-49 shows that the newest five data sets are retained by the primary VRS definition, an additional six versions of the next generation data sets are retained by the next VRS definition GDGOPT, and the oldest generation data set is dropped when the cycles are exceeded. Depending on the creation date, generation data set 4 will be retained before generation data set 3.

```
REMOVABLE MEDIA MANAGER                  VITAL RECORDS RETENTION REPORT                                        PAGE          1
    Copyright IBM Corp. 1993,2007        ----- ------- --------- ------                          TIME 14:09:18 DATE   2009/223
JOB MASK  DATA SET OR VOLUME MASK               OWNER    TYPE RETN   C X   DELETE   DLY COUNT STNUM LOCATION RLSE  LASTREF

          RMM.TEST.NEW.GDG.PARMLIB.OPTION.**    SCHLUM   DSN  CYCLES N N   1999/365   0     5     5 HOME           2009/223
          GDGOPT                                MHLRES7  NEXT CYCLES N N   1999/365         6     6 HOME           2009/223
JOB NAME  DATA SET NAME                         FSEQ  DSEQ  VOLUME VSEQ OWNER    CURRENT  REQUIRED PRTY RETDATE    RETNAME

TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0016    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0006V00  1     1 NS0006    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0015    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0005V00  1     1 NS0005    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0013    1 SCHLUM   SHELF    LIB2     4800 CYCL/00005 *
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0003V00  1     1 NS0003    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0014    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0004V00  1     1 NS0004    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0012    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0002V00  1     1 NS0002    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
TESTGDG   RMM.TEST.NEW.GDG.PARMLIB.OPTION.G0001V00  1     1 NS0011    1 SCHLUM   SHELF    LIB2     4800 CYCL/00006 GDGOPT
NUMBER OF DATA SETS RETAINED (GROUP STORE) =                                                          11     0
```

Figure 10-49 Results of test case ten

Table 10-14 shows that 11 data sets are retained. The G0003V00 generation data set is retained before the G0004V00 generation data set, depending on the creation date.

Table 10-14 Details of test case ten

DSNAME	VOLSER	CRDATE	VRS	Cycle no.	Reason for *(non)* retention	
					Primary	Next[a]
RMM.....G0001V00	NS0001	2009/200	No	12	*Cycles exceeded*	*Cycles exceeded*
RMM.....G0001V00	NS0011	2009/201	Next	11	*Cycles exceeded*	CYCL/00006 6
RMM.....G0002V00	NS0002	2009/207	Next	10	*Cycles exceeded*	CYCL/00006 5
RMM.....G0002V00	NS0012	2009/208	Next	9	*Cycles exceeded*	CYCL/00006 4
RMM.....G0003V00	NS0003	2009/221	Next	6	*Cycles exceeded*	CYCL/00006 3
RMM.....G0003V00	NS0013	2009/222	Prim	5	CYCL/00005 5	
RMM.....G0004V00	NS0004	2009/214	Next	8	*Cycles exceeded*	CYCL/00006 2
RMM.....G0004V00	NS0014	2009/215	Next	7	*Cycles exceeded*	CYCL/00006 1
RMM.....G0005V00	NS0005	2009/228	Prim	4	CYCL/00005 4	
RMM.....G0005V00	NS0015	2009/229	Prim	3	CYCL/00005 3	
RMM.....G0006V00	NS0006	2009/235	Prim	2	CYCL/00005 2	
RMM.....G0006V00	NS0016	2009/236	Prim	1	CYCL/00005 1	

a. These data sets are retained by the next VRS definition.

10.5 DFSMSrmm report generator

Extensive changes have been made to the DFSMSrmm report generator in z/OS V1.11, and these changes have improved usability, enabled more customization of reports, and simplified the way that selection information can be specified. The changes have been made by exploiting recent changes to DFSORT and ICETOOL, data typing, and report type inheritance, The changes further improve the reporting available for DFSMSrmm, DFSMShsm, and other DFSMS components.

The following changes were made:

- You can now override a data type. 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.
- You can now import changes in report types into existing report definitions. This action allows changes in data types, comments, and other criteria to be imported into pre-existing report definitions, thereby improving their usability and allowing your reports to benefit from improved report types shipped with DFSMSrmm.
- You can now select which fields should not be included in totals and break totals. When you specify that a field used for a report column should not 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 they should be excluded.
- You can now 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 values, you must provide a new value to which the field will be changed for the report. Only those fields that have a change value are used for report generation.
- When you specify the column width to be used, the generated ICETOOL statements now override the default ICETOOL processing. If no override is provided, the default is to set the width to a maximum of the column header text and the data size, which is:
  - 10 characters for 4 byte fields, five for 2 byte fields, and three for 1 byte fields of binary data.
  - For numeric fields declared with Z (zoned) or P (packed), the default is set to the number of possible digits.
- You can now use the equated assembler symbols instead of the absolute value. An option in the window displays the available equates ready for you to use. When equates are available, the report generator uses them as a basic set to enable support for the specification of an alternate value for use in the report. This setup is called a *change value*. For example, “I” can be changed to INFO and so on.
- We can now place guidance 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 to make things clearer, such as what variables to set in JCL, how to run HSM preprocessor to convert data, and so on. The help information is split into three parts:
  - Type
  - Report
  - JCL help. It is browsed or edited as a single set of help information.

- You can now request 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 function. In fact, you can perform this action 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 the new DFSORT ICETOOL capability to specify multiple report TITLE strings.
- A new reporting tool is available that enables records to be manipulated with DFSORT. The output of this tool is not a report, but rather reformatted records.

  When using the new reporting tool, the report generator can be considered as a partial replacement for the now withdrawn DFSORT ISPF panels. 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.
- 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.
- You can now use substring for record selection criteria regardless of the data type of the field.

10.5.1 Report Migration Tasks panel

Using the Report Migration Tasks panel, you can compare report types, reports, check for inheritance of new information and criteria, and merge report types. You might use these migration tasks as you migrate to a new release of z/OS or after maintenance has been installed, which enhances or corrects the distributed report types and reports. Use the line commands to process items in the list. Select option 4 MIGRATION from the DFSMSrmm REPORT Generator primary selection panel, as shown in Figure 10-50.

```
  Panel  Help
------------------------------------------------------------------------------
                          DFSMSrmm Report Generator
Option ===> 4

0  OPTIONS         - Specify dialog options and defaults
1  REPORT          - Work with reports
2  REPORT TYPE     - Work with report types
3  REPORTING TOOL  - Work with reporting tools
4  MIGRATION       - Migration tasks for reporting

Enter selected option or END command. For more info., enter HELP or PF1.

5694-A01  COPYRIGHT 1993,2008 IBM CORPORATION
```

Figure 10-50 DFSMSrmm Report Generator primary selection panel

In the Report Migration Task panel, select the report type and the library you would like to check, as shown in Figure 10-51.

```
  Panel  Help
------------------------------------------------------------------------------
                        DFSMSrmm Report Migration Task
Command ===>


2  1. Compare report types
   2. Compare report definitions
   3. Check report type inheritance
   4. Merge report types from Installation
      library to User library

Select Source (S) and Compare (C) Libraries: Default is compare with Product
 S  C  Libraries:     Currently defined libraries:
       User           MHLRES7.REPORT.LIB
       Installation
       Product        SYS1.SAMPLIB

Enter selected option or END command. For more info., enter HELP or PF1.
```

Figure 10-51 DFSMSrmm Report Migration Task panel

As a result of the check, you receive a list of all processed members, as shown in Figure 10-52, and you can use one of the available line commands.

```
  Panel  Help
------------------------------------------------------------------------------
                    DFSMSrmm Report Definition Compare       Row 1 to 3 of 3
Command ===>                                                Scroll ===> PAGE

The following line commands are valid:
   L - List the results of the compare
   S - Display the source report definition
   C - Display the compare report definition
   U - Update the report definition with the new report definition data

 S Report       Differences                          Report title
           Lev Help Typ Mac Sel Sort Rep Dsn Ttl
 - -------- --- ---- --- --- --- ---- --- --- --- ----------------------------
                                                     +
 L ARCGAR01   N    N   N   N   N    N   N   N   N ABARS ARECOVER Statistics
   ARCGDB01   N    N   N   N   N    N   N   N   N DCOLLECT BACKUP DATA
   MYOWNREP   N    N   N   N   N    N   N   N   N DCOLLECT BACKUP DATA
******************************* Bottom of data ********************************
```

Figure 10-52 DFSMSrmm Report Definition Compare panel

If you select L to view the results of the comparison in a detailed list, you see a report like the one shown in Figure 10-53.

```
Report Generator - Report Definition differences       2009/08/17 15:19

                                     Level    Library
Source:  Report definition  ARCGAR01 none     User         MHLRES7.REPORT.LIB
Compare: Report definition  ARCGAR01 none     User         MHLRES7.REPORT.LIB

==> No differences found with: Help information
==> No differences found with: Data type information
==> No differences found with: Macros, macro libraries, and keywords
==> No differences found with: RDW setting
==> No differences found with: Record selection criteria
==> No differences found with: Record sort criteria
==> No differences found with: Report information
==> No differences found with: Input data set name
==> No differences found with: Report title

N- Last change userid: MHLRES7
O- Last change userid: MHLRES7
```

Figure 10-53 DFSMSrmm Report Definition comparison result

10.5.2 Creating a report with multiple title lines using variables

You can specify a report title that has up to three lines. In Figure 10-55 on page 378, we define a title with three lines. Each line must be separated by a comma. In the first line we use the default title information “DCOLLECT DASD CAPACITY PLANNING”, in the second line we have the information “for IBM internal use only”, and in the third line we have the date and time of the report’s creation.

```
                 DFSMSrmm Report Definition - ARCGDD01   Row 16 to 24 of 24
Command ===>                                               Scroll ===> CSR

Report title . . . DCOLLECT DASD CAPACITY PLANNING,for IBM internal use on   +
Report footer  . .
Reporting tool . : ICETOOL                                 Report width:  118
Enter "/" to select option
   Edit the help information for this report

Use END to save changes or CANCEL
The following line commands are valid: S, and R

S CO SO  Field name          Column header text                   CW  Len Typ
- -- --- ------------------- ----------------------------------- --- --- ---
        DCUSYSID            SYSTEM ID FOR THIS OPERATION          9   4 C
        DCUTMSTP            TIMESTAMP FIELD                       8   8 C
....
```

Figure 10-54 DFSMSrmm Report Definition panel

In the help panel shown in Figure 10-55, you can see detailed information about how you can specify variable fields and how you can specify the use of up to three title lines.

```
EDGHG052 -------------------- Report Title ----------------------- HELP

                                                              More:     +
Use this field to specify the Title to appear on each page. You can
specify up to 3 title strings separated by a comma (,). You can also
specify variables instead of a title string. The variable can be any
value you wish prefixed by '&' to represent the field name in the record
of the input file to be used in the title. For example you could use the
date and time as variables.

Identify the field to be used as follows: (r,p,m,f), where:

  r   - is the record number, and must be the same for all
        variables
  p   - is the position of the field in the record
  m   - is the length of the field
  f   - is the format of the field, and will be evaluated by the
        reporting tool

Possible values:
            Up to 3 strings of 1 to 59 characters.
Examples:
   Report title . . . Report Extract created on, &RHCRDATE(1,49,10,CH),
&RHCRTIME(1,59,6,CH)+

Use ENTER to continue, END to exit Help.
```

Figure 10-55 Report Title help panel

The Report Title is a scrollable field, so you can use the LEFT and RIGHT commands when the cursor has been placed within this field to move the field. Figure 10-56 shows that we have placed our cursor before the word and scrolled to the right to provide more space, and then we have added the date and time to the third title line.

Tip: These commands apply when the cursor has been placed within a scrollable field:

- LEFT: Scroll the specified scroll amount to the left.
- RIGHT: Scroll the specified scroll amount to the right.

```
                  DFSMSrmm Report Definition - ARCGDD01   Row 16 to 24 of 24
Command ===>                                               Scroll ===> CSR

Report title . . . _only, &RHCRDATE(1,49,10,CH)  &RHCRTIME(1,59,6,CH)       -+
Report footer  . .
Reporting tool . : ICETOOL                                Report width:  118
Enter "/" to select option
   Edit the help information for this report

Use END to save changes or CANCEL
The following line commands are valid: S, and R

S CO SO  Field name          Column header text                    CW  Len Typ
- -- --- ------------------- ------------------------------------ --- --- ---
         DCUSYSID            SYSTEM ID FOR THIS OPERATION           9   4 C
         DCUTMSTP            TIMESTAMP FIELD                        8   8 C
....
```

Figure 10-56 DFSMSrmm Report Definition panel after using the scroll command

Instead of using the LEFT and RIGHT commands, you can use the ZEXPAND command to display the variable in a dynamic area in a panel, as shown in Figure 10-57. If you can enter input into the scrollable field, then you will be able to update the variable in the expand window.

Tip: The expand panel displays the variable in a scrollable dynamic area. Standard up and down scrolling is supported. You can display the variable in character and hexadecimal using the HEX primary command.

```
                              EDGGRDES+0
                                                        Line   1 of      3
 Command ===>                                           Scroll ===> PAGE

 DCOLLECT DASD CAPACITY PLANNING,for IBM internal use only, &RHCRDATE(1,49,
 10,CH), &RHCRTIME(1,59,6,CH)_____________________________________________
 _____________________________
```

Figure 10-57 DFSMSrmm Report Definition panel

10.5.3 Report generator comparison operands

There are new logical operators that you can use to compare the field contents with the values in the compare value(s) column.

Here is a list of all possible values:

EQ or **=**	Equal.
NE or **<>**	Not equal.
GT or **>**	Greater than.
GE or **>**	Greater than or equal.
LT or **<**	Less than.
LE or **<**	Less than or equal to.
IN	In (Specify a list of possible values, separated by a comma.).
BW	Between (Specify two inclusive values, separated by a comma.).
SE	Substring Equal (The compare value is anywhere in the field.).
SN	Substring Not equal (the compare value).
BO	Compare the field with the specified bit string for all bits are one/all on.
BM	Compare the field with the specified bit string for the bits are mixed/some are one/on.
BZ	Compare the field with the specified bit string for all bits are zero/off/none on.
NO	Compare the field with the specified bit string for all bits are not one/all off.
NM	Compare the field with the specified bit string for all bits are as specified/not some.
NZ	Compare the field with the specified bit string for all bits are not zero/off/not none.

Figure 10-58 shows how you can select a field in the DFSMSrmm Report Definition panel to add the field to your report or to select a field to use this field for report selection decisions only.

```
                    DFSMSrmm Report Definition - ARCGDD01   Row 16 to 24 of 24
Command ===>                                                  Scroll ===> CSR

Report title . . . DCOLLECT DASD CAPACITY PLANNING                           +
Report footer  . .
Reporting tool . : ICETOOL                                  Report width:  118
Enter "/" to select option
   Edit the help information for this report

Use END to save changes or CANCEL
The following line commands are valid: S, and R

S CO SO  Field name           Column header text                  CW  Len Typ
- -- --- -------------------- ----------------------------------- --- --- ---
         DCUSYSID             SYSTEM ID FOR THIS OPERATION         9   4 C
         DCUTMSTP             TIMESTAMP FIELD                      8   8 C
         DCUTIME              TIME IN SMF HEADER FORMAT            8   4 B
         DCUDATE              DATE IN SMF FORMAT (CCYYDDDF)        9   4 C
         DCUDATA              END OF HEADER SECTION                7   1 C
         UCCAPD               DASD CAPACITY PLANNING RECORD       23  23 C
s        UCFLAG1              INFORMATION FLAG #1                  6   1 B
         UCOCCUP              Occupancy information: See guide    10   7 C
         UCCAPDE              END OF DCCCAPD                       4   1 C
```

Figure 10-58 DFSMSrmm Report Definition panel

In the DFSMSrmm Report Criteria panel, use the I line command in front of the new selected field, as shown in Figure 10-59, to get the selection details of this field.

```
  Panel  Help
ssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                    DFSMSrmm Report Criteria - ARCGDD01       Row 1 to 3 of 3
Command ===>                                                 Scroll ===> CSR

Report title : DCOLLECT DASD CAPACITY PLANNING

Use END to save changes or CANCEL
The following line commands are valid: B,D,N,P,R,T, and I (for details)
Operators: EQ = NE <> GT > GE >= LT < LE <= IN BW SE SN BO BM BZ NO NM NZ
Conjunction: AND, OR, AND(, )AND

S Field name           Op Compare value(s)                       Conj Len Typ
- -------------------- -- ---------------------------------------- ---- --- ---
                         +
  DCURCTYP             EQ C                                          2 C
  UCCOLDT              LE &TODAY                                     4 N
I UCFLAG1                                                          1 B
****************************** Bottom of data ********************************
```

Figure 10-59 DFSMSrmm Report Criteria panel

Figure 10-60 shows you the DFSMSrmm Report Criteria Details panel. Enter / in front of the “Select from available equated values” text field to get the DFSMSrmm Report Criteria Equates panel.

```
               DFSMSrmm Report Criteria Details - ARCGDD01

Field name . . . . : UCFLAG1
Operation  . . . . .
Enter "/" to select additional options:
/  Select from available equated values
Or enter compare value(s):
Compare value(s) . .                                                  +
Conjunction  . . . .
Substring position
Substring length . .
Orig field length  :   1
Type . . . . . . . . B           Original field type . . . : B
```

Figure 10-60 DFSMSrmm Report Criteria Details panel

Note: If there are no equate values available, you receive the following message:

```
No equate values exist for variable UCTOTAL
```

If an equate value is available, the DFSMSrmm Report Criteria Equates panel opens. Select the value you want. Sometimes there is only one selection possible, as shown in Figure 10-61.

```
  Panel  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
             DFSMSrmm Report Criteria Equates - ARCGDDSD  Row 1 to 1 of 1
Command ===>                                            Scroll ===> CSR

Field name . . . : UCFLAG1

Enter S to select Equates:
 S  Equate name                 Description                Value
--- -------------------------- -------------------------- -----------
 S  UCLEVEL                     LEVEL OF VOLUME (LO, L1)   B'11000000'
****************************** Bottom of data ********************************
```

Figure 10-61 DFSMSrmm Report Criteria Equates panel

Select a field by typing an S in front of the line and use the end key to save this information and go back to the previous panel. Figure 10-62 shows the result if you are back in the DFSMSrmm Report Criteria Details panel.

```
               DFSMSrmm Report Criteria Details - ARCGDD01

Field name . . . . : UCFLAG1
Operation  . . . . . NO
Enter "/" to select additional options:
   Select from available equated values
Or enter compare value(s):
Compare value(s) . . &EQU=UCLEVEL                                      +
Conjunction  . . . .
Substring position
Substring length . .
Orig field length  :   1
Type . . . . . . . . B           Original field type . . . : B
```

Figure 10-62 DFSMSrmm Report Criteria Details panel

If you agree with the selection, press the End key once more to go back to the DFSMSrmm Report Criteria panel. Figure 10-63 shows you the panel and the correct values for selection in the UCFLAG1 line.

```
  Panel  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
                    DFSMSrmm Report Criteria - ARCGDD01       Row 1 to 3 of 3
Command ===>                                                 Scroll ===> CSR

Report title : DCOLLECT DASD CAPACITY PLANNING

Use END to save changes or CANCEL
The following line commands are valid: B,D,N,P,R,T, and I (for details)
Operators: EQ = NE <> GT > GE >= LT < LE <= IN BW SE SN BO BM BZ NO NM NZ
Conjunction: AND, OR, AND(, )AND

S Field name          Op Compare value(s)                        Conj Len Typ
- ------------------- -- ---------------------------------------- ---- --- ---
                        +
  DCURCTYP            EQ C                                             2 C
  UCCOLDT             LE &TODAY                                        4 N
  UCFLAG1             NO &EQU=UCLEVEL                                  1 B
******************************** Bottom of data ********************************
```

Figure 10-63 DFSMSrmm Report Criteria specification result

You can select as many fields as you want and you can make a comparison multiple times for the same field, as shown in Figure 10-64.

```
EDGPG060             DFSMSrmm Report Criteria - EDGGDSNM     Row 1 to 16 of 30
Command ===>                                                  Scroll ===> CSR

Report title : Mixed Case data sets Retained by VRS,                        Extr

Use END to save changes or CANCEL
The following line commands are valid: B,D,N,P,R,T, and I (for details)
Operators: EQ = NE <> GT > GE >= LT < LE <= IN BW SE SN BO BM BZ NO NM NZ
Conjunction: AND, OR, AND(, )AND

S Field name           Op Compare value(s)                       Conj Len Typ
- -------------------- -- ---------------------------------------- ---- --- ---
                          +
  RXTYPE               EQ X                                          1 C
  XVVRS                EQ Y                                          1 C
  XDVRSR               EQ N                                          1 C
  XDDSNAME             SE a                                     OR   44 C
  XDDSNAME             SE b                                     OR   44 C
  XDDSNAME             SE c                                     OR   44 C
  XDDSNAME             SE d                                     OR   44 C
  XDDSNAME             SE e                                     OR   44 C
  XDDSNAME             SE f                                     OR   44 C
```

Figure 10-64 DFSMSrmm Report Criteria panel with multiple selections for one field

10.6 DFSMSrmm usability items

In this section, we give you an overview about the new DFSMSrmm usability items.

10.6.1 Using MATCHVRS in the ISPF data set panel

The MATCHVRS fastpath command and “M” line command processing now use the matching VRS details, if any, from the data set information to list the matching VRSes. The primary and secondary VRSes are listed if they exist. If there is no matching VRS, dialog processing uses the fully qualified data set name to search for a matching VRS. In addition, the specific volume VRS is also listed if it exists. The results are displayed in a search VRS results list; the order in the list is initially primary, then secondary, or a specific data set, then a volume VRS. The initial list is not sorted, but you can use SORT to reorder the list.

Important: Only after inventory management vital record processing has completed can you use the DFSMSrmm panel to determine how a data set is being retained.

After inventory management vital record processing has completed, the matching and grouping information is included in the `EDGHSKP REPORT` file (see *DFSMSrmm Implementation and Customization Guide,* SC26-7405 for more information). However, you can use the DFSMSrmm panel interactively to determine how a data set is being retained. Using the panel, search for the data set and then, from the data set search results list, use the M line command to view the matching VRS information. Alternatively, use the I or S line commands in the data set search results list to display the data set details. You can then place the cursor on either the “VRS name” or the “Value or class” highlighted point and shoot panel text for a matching VRS and press Enter (or use the equivalent MATCHVRS primary command) to see what retention policies are used. The M line command and the MATCHVRS primary command use the matching VRS details, if any, from the data set information to list the matching VRSes. The primary and secondary VRSes are listed, if they exist. If there are no matching VRS details, dialog processing uses the fully qualified data set name to search for a matching VRS. In addition, the specific volume VRS is also listed, if it exists.

This new function lists VRSes that match the data set or volume displayed on the panel. You can use it as a line command in front of the data set name or volume, as shown in Figure 10-65.

```
EDGPDO2O        DFSMSrmm Data Sets (Page 1 of 2)           Row 1 to 1 of 1
Command ===>                                              Scroll ===> PAGE


Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
                                                 Volume          File
S  Data set name                                 serial Owner    seq
-- ------------------------------------------- ------ -------- -----
M  SCHLUM7.TEST.PRTITION.RULES                   THSO21 MHLRES7  1
```

Figure 10-65 MATCHVRS line command

Also, you can use this function in the data set detail panel if you write MATCHVRS in the command line and press Enter to display the matching VRS results list. This is equivalent to using the MATCHVRS as a primary command in this panel. Figure 10-66 shows you the use of this command. You can abbreviate MATCHVRS to “M” or any other valid matching abbreviation.

```
  Panel  Help
------------------------------------------------------------------------------
EDGPD110                 DFSMSrmm Data Set Details
Command ===> MATCHVRS

Data set name . . : 'MHLRES7.TEST.PRTITION.RULES'
Volume serial . . : THS021      Physical file sequence number . . : 1
Owner . . . . . . : MHLRES7     Data set sequence number  . . . . : 1
                                                                More:   -
Date last written : 2008/095               Management class  . . : MCDB22
                                           Data class  . . . . . :
Retention date  . : CYCL/99999             Storage class . . . . : SCLIB2
VRS retained  . . : YES                    Storage group . . . . : SGLIB2

Security name . . :                        BES key index . . . . : 0
Classification  . :

Primary VRS details:  (Use MATCHVRS primary command to display matching VRSes)
 VRS name . . . : SCHLUM.TEST.**
 Job name . . . :                            VRS type  . . . . . : DSN/MV
 Subchain name  :                            Subchain start date :

Secondary VRS details:
 Value or class : CATALOG
 Job name . . . :
 Subchain name  :                            Subchain start date :

Catalog status  . : YES
Closed by Abend . : NO                     Deleted . . . . . . . : NO
```

Figure 10-66 Use MATCHVRS in the data set detail panel

The panel shows you all the VRS definitions matching the selected data set name or volume serial number, as shown in Figure 10-67. You can see that a VRS management class (MC) or management vale (MV) of CATALOG is assigned to the tested data set.

```
EDGPV020                DFSMSrmm VRSs (Page 1 of 4)        Row 1 to 3 of 3
Command ===>                                              Scroll ===> PAGE

Enter HELP or PF1 for the list of available line commands.
Use the LEFT and RIGHT commands to view other data columns.

S  Volume/Data set/Name specification           Job name Type Location Prty
-- ---------------------------------------- -------- ---- -------- ----
I  SCHLUM.TEST.**                                     DSN  HOME     0
   THS001                                             VOL  HOME     0
```

Figure 10-67 MATCHVRS result

Note: In the VRS panel, you can place the cursor on a matching VRS, specify “I” or “S”, and press Enter to see what retention policies it specifies. You can use any other possible line command in the VRS panel.

To get the same result when using the MATCHVRS command, place your cursor on the new point and shoot fields VRS name or Value or class and press Enter, as shown in Figure 10-68. We have placed the cursor on the “V” of the VRS name field.

```
  Panel  Help
------------------------------------------------------------------------------
EDGPD110                  DFSMSrmm Data Set Details
Command ===>

Data set name . . : 'MHLRES7.TEST.PRTITION.RULES'
Volume serial . . : THS021      Physical file sequence number . . : 1
Owner . . . . . . : MHLRES7     Data set sequence number  . . . . : 1
                                                                More:   -
Date last written : 2008/095               Management class  . . : MCDB22
                                           Data class  . . . . . :
Retention date  . : CYCL/99999             Storage class . . . . : SCLIB2
VRS retained  . . : YES                    Storage group . . . . : SGLIB2

Security name . . :                        BES key index . . . . : 0
Classification  . :

Primary VRS details:  (Use MATCHVRS primary command to display matching VRSes)
 VRS name . . . : SCHLUM.TEST.**
 Job name . . . :                            VRS type  . . . . . : DSN/MV
 Subchain name  :                            Subchain start date :

Secondary VRS details:
 Value or class : CATALOG
 Job name . . . :
 Subchain name  :                            Subchain start date :

Catalog status  . : YES
Closed by Abend . : NO                     Deleted . . . . . . . : NO
```

Figure 10-68 Use the point and shoot function in the data set detail panel

10.6.2 New ADDVOLUME subcommand fields

The RMM ADDVOLUME subcommand has been enhanced with following new abilities:

- You can now specify the volume type as physical/logical when adding scratch or private volumes.
- You can now specify the storage group name when adding scratch volumes.
- You can now specify the creation date and time when adding volumes. This enables you to specify any value you wish and enables DFSMSrmm to have the correct value, such as when the volume was manufactured.

ADDVOLUME ISPF panel

The ADDVOLUME ISPF panel is enhanced to support the new functions. Figure 10-69 shows the new fields for Storage group, Media information, and Create date and time.

```
  Panel  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDGPT210                    DFSMSrmm Add Volume
Command ===>
                                                                        More:     +
Volume . . . . . . . VT0000       Pool  . . . . . . . .
Count  . . . . . . .               or
Volume type  . . . . PHYSICAL     Rack  . . . . . . . .
Media name . . . . . 3592
Vendor . . . . . . .              Media information   IBM
Storage group  . . .              Location name . . . SHELF
Retention period . .              Expiration date . . 2009/365    YYYY/DDD
Create date  . . . . 2009/222     Create time . . . . 13:49:45

Status . . . . . . . MASTER       Initialize  . . . .
Description  . . . .
Data set name  . . . 'RMM.F00001'
```

Figure 10-69 ADDVOLUME panel part 1

As you can see in Figure 10-70, there is no change.

```
  Panel  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDGPT210                    DFSMSrmm Add Volume
Command ===>
                                                                        More:     +
Media type . . . . . ETC          Release actions
Label  . . . . . . . SL             Return to SCRATCH pool . . YES
  Current version                   Replace volume . . . . . . NO
  Required version                  Return to owner  . . . . . NO
Density  . . . . . . IDRC           Initialize volume  . . . . NO
Recording format . . EFMT1          Erase volume . . . . . . . NO
Compaction . . . . . YES            Notify owner . . . . . . . NO
Attributes . . . . . NONE
Owner  . . . . . . .                            Owner access . . . . ALTER

Assigned date  . . . 2009/224    YYYY/DDD       Assigned time  . . . 23:34:50
```

Figure 10-70 ADDVOLUME panel part 2

In the last page of the ADDVOLUME panel there also no changes, as shown in Figure 10-71.

```
  Panel  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDGPT210                    DFSMSrmm Add Volume
Command ===>

                                                                        More:   -
Assigned date  . . . 2009/224    YYYY/DDD      Assigned time  . . . 23:34:50
Last read date . . . 2009/225                  Last write date  . . 2009/225
Security name  . . .

Account number . . . 999,POK

Jobname  . . . . . . TEST9999                  MVS use  . . . . . . . YES
                                               VM use . . . . . . . . NO
Loan location  . . .                           IRMM use . . . . . . . NO
Previous volume  . .

Volume access list:                           Access . . . . . . . . NONE
  User . .           User . .           User . .           User . .
  User . .           User . .           User . .           User . .
  User . .           User . .           User . .           User . .
```

Figure 10-71 ADDVOLUME panel part 3

Figure 10-72 shows you the new fields for Storage group, Media information, and Create date and time, which can be specified in the Add Scratch Volume ISPF panel.

```
  Panel  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDGPT230                DFSMSrmm Add Scratch Volumes
Command ===>

Volume . . . . . VT0000     Pool  . . . . . .
                             or
Volume type  . . . . PHYSICAL   Rack  . . . . . .
Media name . . . . . 3592
Vendor . . . . . .           Media information   IBM
Storage group  . . .         Location name . . . SHELF

Count  . . . . . .        ( Default is 1 )

Description  . . . .
Account number . . . 999,POK
Assigned date  . . . 2009/224    YYYY/DDD        MVS use . . . . . . YES
Assigned time  . . . 23:34:50                    VM use  . . . . . . NO
Create date  . . . . 2009/222                    IRMM use  . . . . . NO
Create time  . . . . 13:49:45

Media type . . . . . ETC
Label  . . . . . . . SL      ( AL, NL or SL )
  Current version          Label version number( for example 3 )
  Required version         Label version number( for example 4 )
Density  . . . . . . IDRC    ( 1600, 3480, 6250 or * )
Initialize . . . . .         ( Default is YES )

Press ENTER to ADD one or more SCRATCH volumes, or END command to CANCEL.
```

Figure 10-72 ADDVOLUME panel to add a scratch volume

In the Add Stacked Volume ISPF panel, you can specify the new fields for Storage group, Media information, and Create date and time here as well, as shown in Figure 10-73.

```
  Panel  Help
sssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssss
EDGPT240                DFSMSrmm Add Stacked Volume
Command ===>

Volume . . . . . VT0000
Count  . . . . . .

Location name  . . . SHELF                    Pool prefix  . . . .
Create date  . . . . 2009/222    YYYY/DDD      or
Create time  . . . . 13:49:45                  Rack number  . . . .

Description  . . . .
Owner  . . . . . . . MHLRES7

Loan location  . . .

Media name . . . . . 3592                     Media information    IBM
Media type . . . . . ETC                      Vendor . . . . . .
Recording format . . EFMT1

Security name  . . .
Account number . . . 999,POK

Press ENTER to ADD one or more STACKED volumes, or END command to CANCEL.
```

Figure 10-73 ADDVOLUME panel to add a stacked volume

RMM TSO ADDVOLUME subcommand

In the following section, we describe how to use the new RMM TSO ADDVOLUME subcommands enhancements.

CRDATE

This enhancement specifies the date when the volume was created. Supply the year and day in one of two forms. As a best practice, use the yyyy/ddd format rather than the yyddd format for dates.

The formats are:

- yyyy/ddd, where yyyy is the four-digit number for the year. The maximum allowable value for yyyy is 9799. ddd is the three-digit number for the day of the year, such as 2004/001. The slash is required. You can specify a date in the range between 0000/000 to 9799/365.
- yyddd, where yy is the last two-digit number for the year and ddd is the three-digit number for the day of the year, such as 04001. When you use the yyddd format, DFSMSrmm determines the century by using a date window:
  - DFSMSrmm uses the current century if the difference between the current year and the specified year is not greater than 50.
  - DFSMSrmm uses the previous or next century if the difference between the current year and the specified year is greater than 50.

  You can specify a date in the range between 00000 to 99366.

The default is the date you issue the ADDVOLUME subcommand.

Figure 10-74 shows the correct use of the CREATEDATE operand.

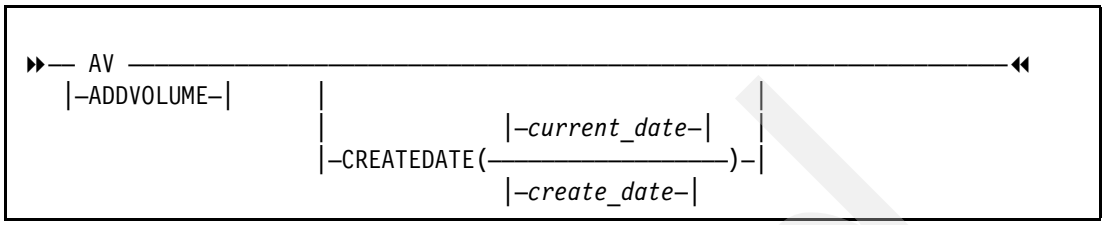


Figure 10-74 ADDVOLUME subcommand using the CREATEDATE syntax

CRTIME

This enhancement specifies the time when the volume was created. The CRTIME format is hhmmss, where:

- hh is hours
- mm is minutes
- ss is seconds

For example, nine o'clock in the morning is 090000.

You can specify a time in the range between 000000 to 235959.

The default is the time when you issue the ADDVOLUME subcommand.

Figure 10-75 shows the correct use of the CREATETIME operand.

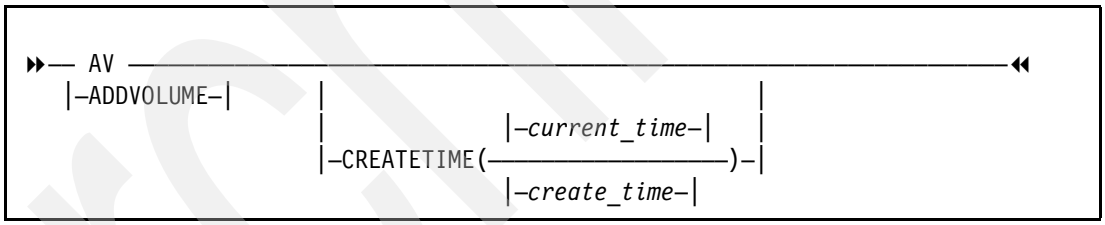


Figure 10-75 ADDVOLUME subcommand using the CREATETIME syntax

MEDINF

This enhancement specifies the assigned installation-defined media information to the volume. This value is one to eight alphanumeric characters and must be defined once in your installation. You can use the LISTCONTROL subcommand with the MEDINF operand to display the media information. For more information about MEDINF, see *z/OS DFSMSrmm Implementation and Customization Guide*, SC26-7405. The default media information is IBM.

Figure 10-76 shows you the correct use of the MEDIAINF operand.

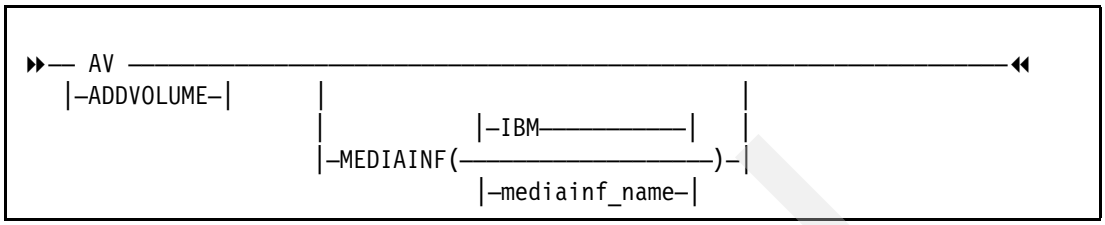


Figure 10-76 ADDVOLUME subcommand using the MEDIAINF syntax

STORAGEGROUP

This enhancement specifies the SMS-defined storage group ADDVOLUME subcommand to which the volume belongs. A storage group name is one to eight characters, other than blanks, commas, and semicolons. A storage group name can be a value that matches to a VLPOOL NAME value but does not need to be defined on a VLPOOL definition. STORAGEGROUP may be abbreviated as STORGRP.

For volumes in a system-managed library, DFSMSrmm uses the current location or the location specified in the command to validate the specified storage group. For volumes with a system-managed home location defined, DFSMSrmm uses the home location for validation. For other volumes, any value you specify is accepted as long as the value is valid in the current SMS configuration.

You can set the storage group even if the TCDB already contains a storage group name.

A storage group name can be assigned to any volume, even a scratch volume. The storage group name can be used for scratch pooling, except when the volume is in a system-managed automated tape library. For system-managed scratch volumes, the storage group name is not maintained in the TCDB, because it is not supported by SMS tape processing.

For system-managed manual tape library volumes, the storage group name is used for scratch pool validation only when you request that a specific storage group name is used for pooling. For all non-system managed scratch pooling validation, the storage group name is significant and is always used to ensure that a volume from the correct pool is mounted.

When you do not specify a storage group name, DFSMSrmm assigns a storage group name by using the matching EDGRMMxx VLPOOL NAME value. If the VLPOOL NAME value is a valid storage group name, DFSMSrmm uses the VLPOOL NAME value as the default value for volumes added to the pool.

If you use STATUS(VOLCAT), the value you specify is overridden by information in the TCDB. It is ignored if the volume is already defined in a TCDB with a different storage group name.

Figure 10-77 shows the correct use of the STORAGEGROUP operand.

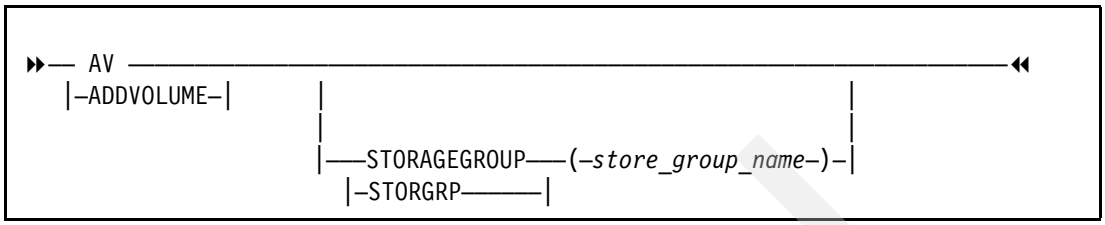


Figure 10-77 ADDVOLUME subcommand using the STORAGEGROUP syntax

10.6.3 Extended SEARCHVOLUME subcommand

The RMM TSO SEARCHVOLUME subcommand's functionality has ben expanded to give you more flexibility and scope for querying a wider range of volume attributes such as dates, actions, options, and flag settings. The new functions are:

- Additional operands are added enabling more extensive searches, including many for specific date ranges.
- There is new support to ensure that specific flag combinations can be set.

SEARCHVOLUME ISPF panel

Figure 10-78, Figure 10-79 on page 396, and Figure 10-80 on page 396 show you the new ISPF SEARCHVOLUME panel format to use the new SEARCHVOLUME operands available with this release. In the first part of the panel, you can see the new selection of the date formats you can use, the enhanced actions, and release selections.

```
  Panel  Help  Scroll
------------------------------------------------------------------------------
                           DFSMSrmm Volume Search
Command ===>

                                                                    More:     +
Volume  . . . . VT*        May be generic. Leave blank for all volumes
Owner . . . . . *          Owned by a specific user. Default is your userid
Job name  . . .            May be generic
Limit . . . . . *          Limit search to first nnnn volumes
Media name  . .            Limit to a single media name
Vendor  . . . .            Supplier of media
Pool prefix . .            or to a particular pool
Status  . . . .
                           Select volume status. Default is ALL
Dates            Start          End       Date, date range or relative value
  Assigned  . .  yyyy/mm/dd . .  yyyy/mm/dd
  Create  . . .              . .
  Reference . .  2008/01/01 . .  -1M
  Read  . . . .              . .
  Write . . . .              . .
  Changed . . .              . .
  Moved . . . .              . .

Actions . . . .
                           Specify one or more pending actions
Release
  Actions . . .
                           Specify one or more release actions

Enter HELP or PF1 for the list of available line commands
```

Figure 10-78 RMM SEARCHVOLUME panel 1 of 3

In the second part of the panel, you can see that it is now possible to limit your search by setting an original expiration date selection.

```
   Reference . .              . .
  Read  . . . .              . .
  Write . . . .              . .
  Changed . . .              . .
  Moved . . . .              . .

Actions . . . .
                           Specify one or more pending actions
Release
  Actions . . .
                           Specify one or more release actions
  Options . . .
                           Specify one or more release options
System use  . .
                           Select system use. Default is ALL
Since . . . . .            Volumes assigned since YYYY/DDD
Retention . . .            Volumes retained up to YYYY/DDD
Original EXPDT             YES, NO, or a specific date YYYY/DDD
Clist . . . . .            YES to create a data set, or NO, or blank

Home  . . . . .            Limit to volumes with this home location name
Location  . . .            Limit to volumes by location. May be generic
In container               Stacked volser
```

Figure 10-79 RMM SEARCHVOLUME panel 2 of 3

In the third part of the new ISPF SEARCHVOLUME panel, you can limit your search for specific media information, for example, if the volume is part of a multi volume set, if the volume is retained by a multi-volume set or not, or if the FORCE flag is set for a volume or not.

```
In container               Stacked volser

Volume type . .            ( LOGICAL , PHYSICAL or STACKED )
MedInf  . . . .            Media information name ( IBM or MEDINF defined)
Media type  . .            Tape media type ( for example HPCT )
Label . . . . .            Tape label standard ( for example SL )
  Current version          Label version number( for example 3 )
  Required version         Label version number( for example 4 )
VOL1 volser . .            Volser in the VOL1 label ( NONE or volser )
Density . . . .            Tape recording density
Format  . . . .            Tape recording format ( for example 256TRACK )
Worm  . . . . .            YES or NO to limit volumes based on WORM, or blank

Compaction  . .            Limit to volumes containing compacted data
Attributes  . .            Tape special attributes ( NONE or RDCOMPAT )

Destination . .            Limit by destination
Required  . . .            Limit by required location
Move mode . . .            Limit by move mode ( AUTO or MANUAL )
Intransit . . .            Limit to volumes which are moving ( YES or NO )
Volume set  . .            Volumes in a set/chain ( YES, NO, FIRST, or LAST )
Set retained               Volumes retained BYSET ( YES or NO )
Force . . . . .            Volumes used with FORCE ( YES or NO )

Enter HELP or PF1 for the list of available line commands
```

Figure 10-80 RMM SEARCHVOLUME panel 3 of 3

Figure 10-82 on page 398 shows the expanded help functions available in the ISPF panel.

```
EDGHT010 ------------------ DFSMSrmm Volume Search ----------------------HELP
COMMAND ===>


   Use the DFSMSrmm Volume Search panel to generate a list of volumes defined
   to DFSMSrmm that match the criteria you specify.

   Use ENTER to see the following topics in sequence or choose them by number.

     1    Volume search - Overview
     2    Volume                          30    Medinf
     3    Owner                           31    Media type
     4    Job name                        32    Label
     5    Limit                           33    Label version
     6    Media name                      34    VOL1 volser
     7    Vendor                          35    Density
     8    Pool                            36    Format
     9    Status                          37    Worm
    10    Dates                           38    Compaction
    11    Actions and Release             39    Attributes
    12    Release Options                 40    Destination
    13    System use                      41    Required
    14    Since                           42    Move mode
    15    Retention                       43    Intransit
    16    Original EXPDT                  44    Volume set
    17    Clist                           45    Set retained
    18    Home                            46    Force
    19    Location                        47    Line commands
    20    In container                    48    Volume list
    21    Volume type                     49    Volume list sort order
   Use ENTER to continue, END to exit Help.
```

Figure 10-81 RMM SEARCHVOLUME primary help panel

Figure 10-82 shows the correct use of the start and end dates in the ISPF panel.

```
HELP ------------------------------ DATES ------------------------------ HELP
Command ===>
                                                                      More:   -
   Possible values for Start and End dates:
       A date in the currently selected date format:
                The current date format type is JULIAN
                The current date format style is YYYY/DDD
       A relative day, month, year:
                The value range for relative values is 0 to 99999, with a
                required leading '-' and an optional suffix of M!Y.
                -n     - a date n days before the current date. 0 means
                         the current day, current month, current year.
                -nM    - a date n months before the current month and the
                         current day in the month is as the current date.
                -nY    - a date n years before the current year and the
                         current day in the year is as the current date.

   For example:
       Dates            Start          End
         Assigned  . . 2008/001    . . -5
                                     (DFSMSrmm lists volumes assigned between
                                      1 January 2008 and 5 days ago)
   Use ENTER to continue, END to exit Help.
```

Figure 10-82 RMM SEARCHVOLUME help panel for start and end dates

Figure 10-83 shows the use of the retention field.

```
EDGHT01B ------------------------ RETENTION ----------------------------HELP
COMMAND ===>

   Use the Retention field to specify that DFSMSrmm lists only MASTER and USER
   volumes which will expire up to and including this date. Specify true dates
   or DFSMSrmm special retention date formats.
   To list volumes with a permanent expiration date and not retained by VRS
   you have to specify the expiration dates 1999/365 or 1999/366.
   Search is based on the retention date for volumes retained by VRS, and the
   expiration date for volumes not retained by VRS.

   Possible values:
       A date in the currently selected date format:
                The current date format type is JULIAN
                The current date format style is YYYY/DDD
       CATRETPD      - Limit to those retained but not yet catalogued.
       PERMANENT     - Limit to those retained permanently by VRS.
       WHILECATLG    - Limit to those retained by catalog control.
       CYCL/nnnnn    - Limit to the same number or fewer cycles.

   For example:
       Retention   ===>  1991/074      (DFSMSrmm lists volumes which have a
                                        retention date prior to 16 March 1991)
   Use ENTER to continue, END to exit Help.
```

Figure 10-83 RMM SEARCHVOLUME retention help panel

RMM TSO SEARCHVOLUME subcommand

In this section, we describe how you can use the new RMM TSO SEARCHVOLUME subcommands enhancements.

ACTION and RELEASEACTION

Figure 10-84 on page 400 shows the correct use of the expanded ACTION and RELEASEACTION operand. You can check for specific combinations of actions by using the NOxxxx values, or by using ONLY with a list of actions that must be set. When you do not specify an ONLY or a NOxxxx value, DFSMSrmm lists volumes that have one or more of the specified actions pending. To limit the list to a specific combination, use ONLY or include a list of NOxxx values.

When you specify ONLY with no other operand values, DFSMSrmm selects volumes that have no release action set. This is the same as specifying NOERASE, NOINIT, NONOTIFY, NOREPLACE, NORETURN, or NOSCRATCH.

Use the RELEASEACTION operand to search for volumes that have the specified release action set. Volumes are returned if any of the values you specify are set in the volume. Refer to the help for the ACTION operand for information about how to search for volumes with pending actions.

Use the ACTION(pending_actions) operand to search for volumes that have the specified release action set. Volumes are returned if any of the values you specify are set in the volume.

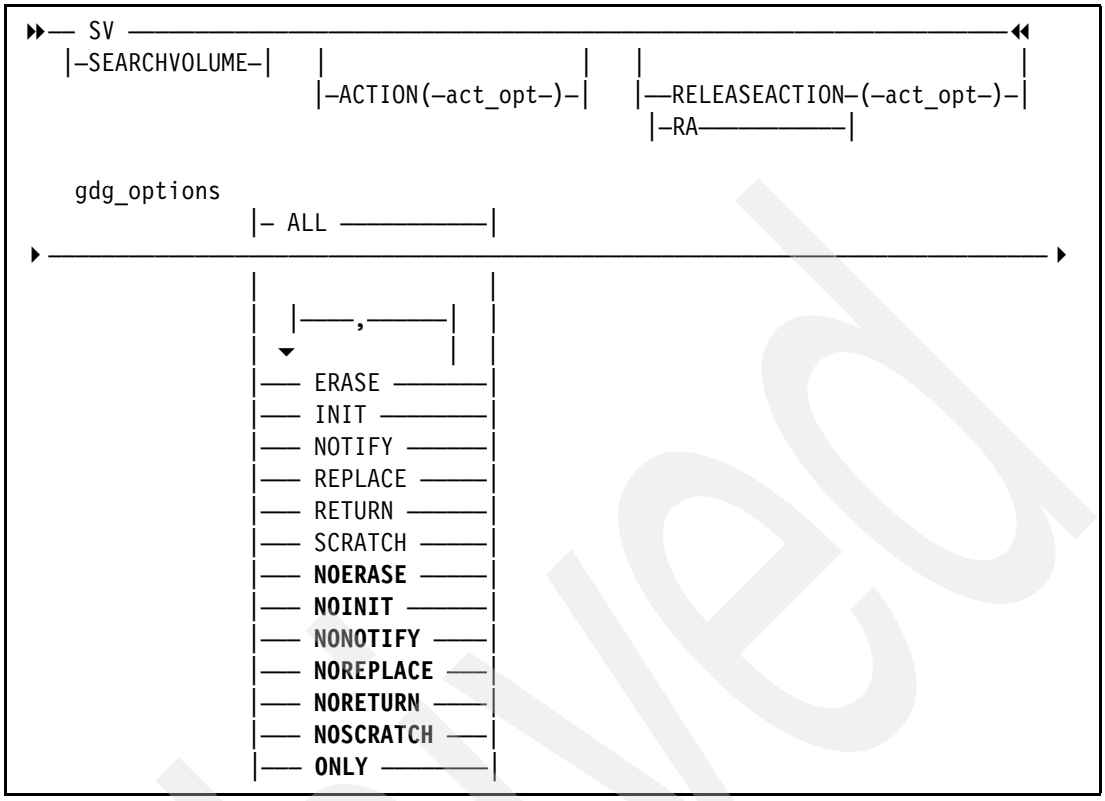


Figure 10-84 RMM SEARCHVOLUME subcommand using ACTION and RELEASEACTION

Where:

ALL Used to list all volumes with any pending action.

ERASE Used to list only volumes that require erasing.

INIT Used to list only volumes that require initialization.

NOTIFY Used to list only volumes for which owners must be notified.

REPLACE Use to list only volumes that must be replaced by new volumes and returned to the scratch pool.

RETURN Used to list only volumes that should be returned to their owner.

SCRATCH Used to list all volumes to be returned to scratch status.

NOERASE Used in combination with ALL or other operands to exclude volumes that require erasing.

NOINIT Used in combination with ALL or other operands to exclude volumes that require initialization.

NONOTIFY Used in combination with ALL or other operands to exclude volumes for which owners must be notified.

NOREPLACE Used in combination with ALL or other operands to exclude volumes that must be replaced by new volumes and returned to the scratch pool.

NORETURN Used in combination with ALL or other operands to exclude volumes that should be returned to their owner.

NOSCRATCH Used in combination with ALL or other operands to exclude volumes to be returned to scratch status.

ONLY Used in combination with other operands to limit the list to a specific combination. For example, ACTION(RETURN,NOTIFY,ONLY) will list only those volumes for which both the RETURN and NOTIFY pending release action apply. If ONLY were omitted, then DFSMSrmm would list all the volumes for which either pending release action applied.

When you specify ONLY with no other operand value, DFSMSrmm selects volumes that have no pending release action set. This is the same as specifying NOERASE, NOINIT, NONOTIFY, NOREPLACE, NORETURN, or NOSCRATCH.

RELEASEOPTION

Figure 10-85 shows how you can use the RELEASEOPTION operand of the RMM SEARCHVOLUME subcommand. Use the RELEASOPTION subcommand to list volumes based on their release options. Release options are set by VRSEL processing when data sets match to, or are retained by, a VRS.

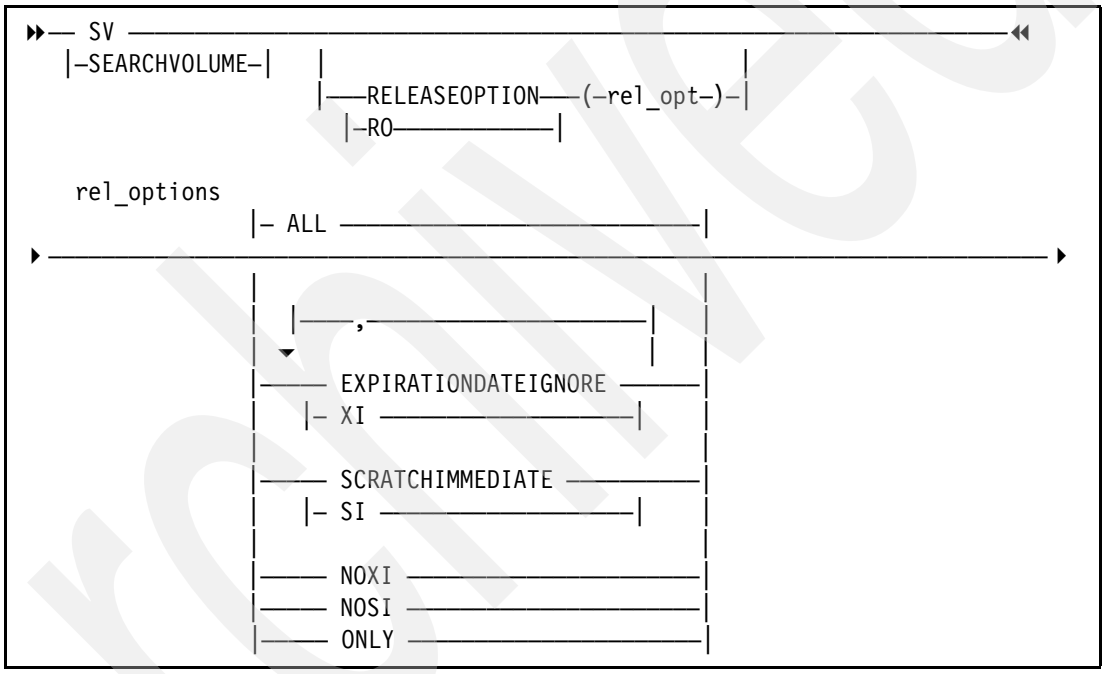


Figure 10-85 RMM SEARCHVOLUME subcommand using RELEASEOPTION

Where:

ALL List all volumes that have any release options of any kind. Specifying the ALL operand value is the same as specifying RELEASEOPTION(EXPIRYDATEIGNORE,SCRATCHIMMEDIATE).

EXPIRYDATEIGNORE List all volumes that have the “expiry date ignore” release option. EXPIRYDATEIGNORE may be specified as XI.

NOSI List all volumes that do not have the “scratch immediate” release option. Can be used in combination with ALL or other operands to exclude volumes that have the “scratch immediate” release option.

NOXI List all volumes that do not have the “expiry date ignore” release option. Can be used in combination with ALL or other operands to exclude volumes that have the “expiry date ignore” release option.

ONLY Can be used in combination with other operands to limit the list to a specific combination. For example, RELEASEOPTION(SI,XI,ONLY) will list only those volumes for which both the SI and XI release options apply. If ONLY were omitted, then DFSMSrmm would list all the volumes for which either release option applied. When you specify ONLY with no other operand value, DFSMSrmm selects volumes that have no release options set. This is the same as specifying RELEASEOPTION(NOSI,NOXI).

SCRATCHIMMEDIATE List all volumes that have the “scratch immediate” release option. SCRATCHIMMEDIATE may be specified as SI.

We have two volumes, NS0001 and NS0002, with a RELEASEACTION of INIT, but only the NS0001 volume has the additional RELEASEACTION RETURN. Figure 10-86 shows you the result if you using the RELEASEACTIO(INIT,RETURN) operand.

```
RMM SV VOLUME(*) OWNER(*) LIMIT(*) RELEASEACTION(INIT,RETURN)


Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                      date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
NS0001 SCHLUM         2008/200   1999/365   SHELF    1     U
NS0002 SCHLUM         2008/207   1999/365   SHELF    1     UV
EDG3012I 2          ENTRIES LISTED
```

Figure 10-86 RMM SEARCHVOLUME example using the RELEASEACTION operand

In Figure 10-87, we have used the same RMM TSO SEARCHVOLUME subcommand as before, but we have added the ONLY option to the RELEASEACTION operand. As you can see, only the NS0001 volume is listed, because for this volume we have both RELEASEACTIONS set.

```
RMM SV VOLUME(*) OWNER(*) LIMIT(*) RELEASEACTION(INIT,RETURN,ONLY)


Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                      date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
NS0001 SCHLUM         2008/200   1999/365   SHELF    1     U
EDG3012I 1          ENTRIES LISTED
```

Figure 10-87 RMM SEARCHVOLUME example using the RELEASEACTION operand with ONLY

FORCE

Figure 10-88 shows how you can use the FORCE operand of the RMM SEARCHVOLUME subcommand.

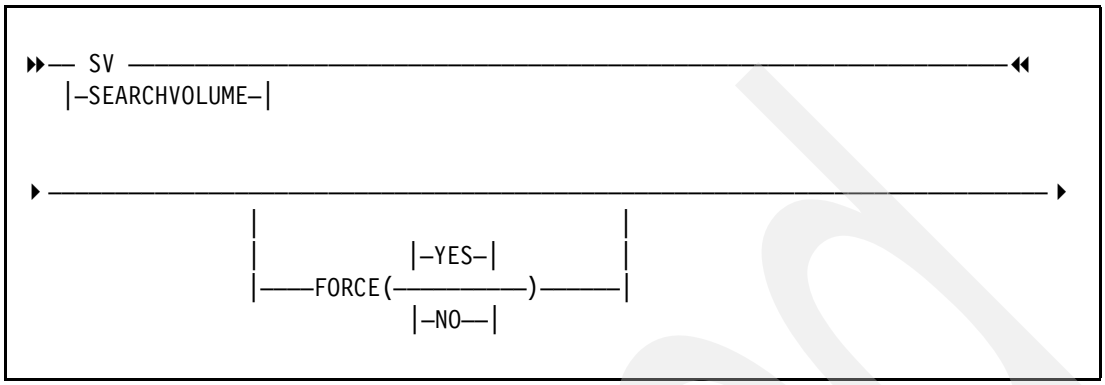


Figure 10-88 RMM SEARCHVOLUME subcommand using FORCE

Where:

YES Use to list volumes that have the FORCE attribute set.

NO Use to list volumes that do not have the FORCE attribute set.

BYSET

Figure 10-89 shows you how you can use the BYSET operand of the RMM SEARCHVOLUME subcommand.

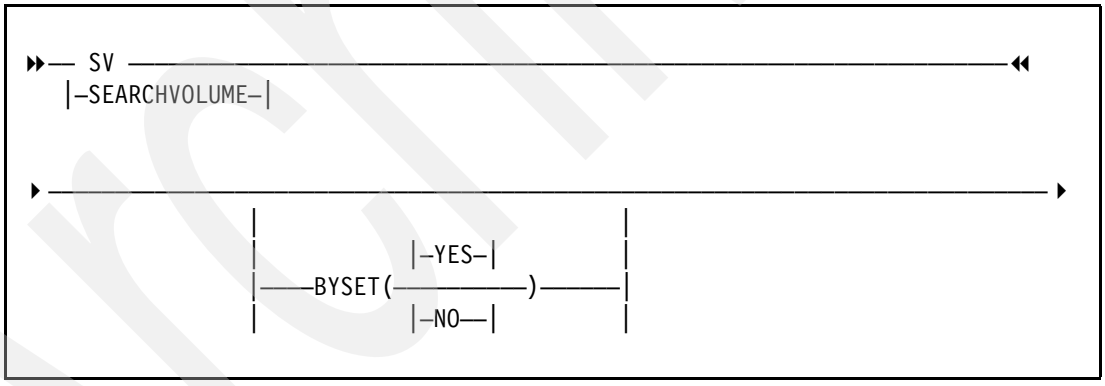


Figure 10-89 RMM SEARCHVOLUME subcommand using BYSET

Where:

YES Used to list volumes that have the SET RETAINED attribute set.

NO Used to list volumes that do not have the SET RETAINED attribute set.

INSET

Use the INSET operand of the RMM SEARCHVOLUME subcommand, as shown in Figure 10-90, to get a list of volumes based on whether or not a volume is part of a multi-volume chain or set. The INSET operand checks the previous and next volume information in the volume record.

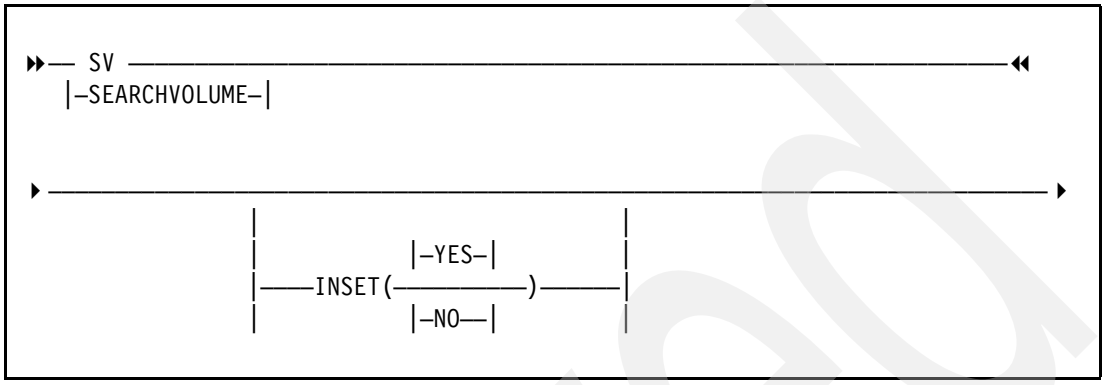


Figure 10-90 RMM SEARCHVOLUME subcommand using INSET

Where:

YES Used to list volumes that have the SET RETAINED attribute set.

NO Used to list volumes that do not have the SET RETAINED attribute set.

DATE operands using a date range

Use the ASDATE, CRDATE, LASTREFDATE, READDATE, WRITEDATE, LASTCHANGEDATE, or MOVEDATE|STOREDATE operand to specify a date range for your search. A date_range consists of a start date and an end date. Each date can be an absolute date in either yyyy/ddd or yyddd format, or it can be a relative value from which DFSMSrmm calculates the date.

Figure 10-91 shows a list of volumes based on whether or not a volume is part of a multi-volume chain or set. The INSET operand checks the previous and next volume information in the volume record.

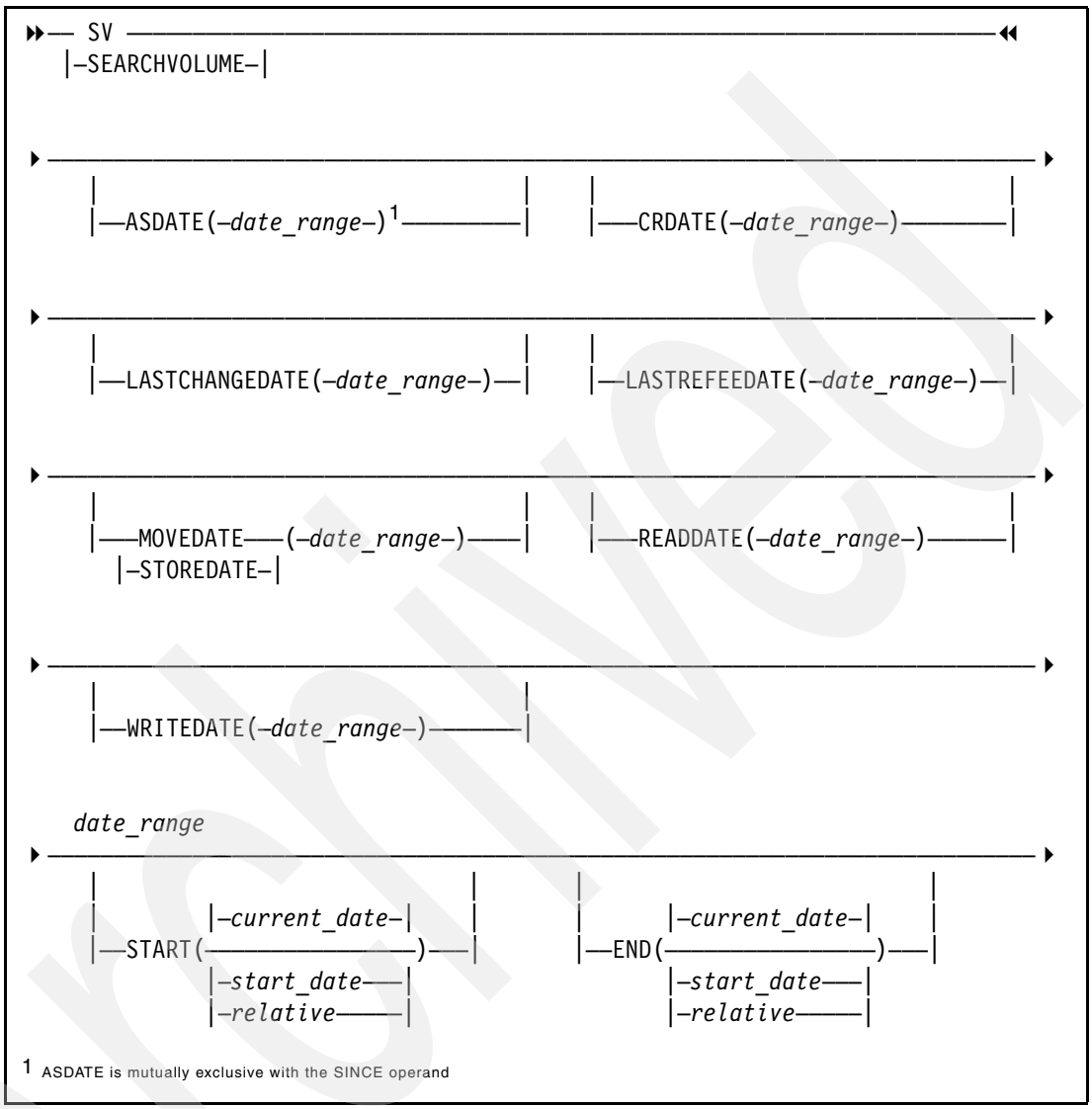


Figure 10-91 RMM SEARCHVOLUME subcommand using INSET

Where:

- Absolute dates are specified as either yyyy/ddd or yyddd format. For example, January 3, 2003 may be specified as 2003/003 or 03003.

current_date Only volumes whose assigned date is the current date are listed. Specify ASDATE or ASDATE() to get the current system date.

end_date Only volumes whose assigned date is on or before the specified end date are listed, where end_date is either an absolute date or relative date. Note that because START defaults to the current date, the specified end date must be equal to or greater than the current date when START is omitted.

start_date Only volumes whose assigned date is on or after the specified start date are listed, where start_date is either an absolute date or relative date.

start_date end_date Only volumes whose assigned date is within the range delimited by the specified start and end dates are listed, where both start_date and end_date are either an absolute date or relative date. The specified end date must be equal to or greater than the specified start date.

- Relative dates are specified as a number of days, months, or years prior to the current date.

  relative A relative value is specified as a negative number of days, months, or years.

  -0 Specifies the current day, current month, and current year.

  -n Specifies that the date is n days before the current date.

  -nM Specifies that the date is n months before the current month and the current day in the month is the current date.

  -nY Specifies that the date is n years before the current year and the current day in the year is the current date.

  The value range for n is 0 to 99999, with a required leading '-' and an optional suffix of M or Y.

We have 12 volumes owned by the SCHLUM user. This volumes have all an assigned date in the range between are 2008/200 and 2008/236. Figure 10-92 shows you the result if you use the SEARCHVOLUME command with the LIMIT(*) and OWNER(SCHLUM) operands only.

```
RMM SV VOLUME(*) OWNER(SCHLUM) LIMIT(*)

Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- -------
NSO001 SCHLUM         2008/200   1999/365   SHELF    1     U
NSO002 SCHLUM         2008/207   1999/365   SHELF    1     UV
NSO003 SCHLUM         2008/221   1999/365   SHELF    1     UV
NSO004 SCHLUM         2008/214   1999/365   SHELF    1     UV
NSO005 SCHLUM         2008/228   1999/365   SHELF    1     UV
NSO006 SCHLUM         2008/235   1999/365   SHELF    1     UV
NSO011 SCHLUM         2008/201   1999/365   SHELF    1     UV
NSO012 SCHLUM         2008/208   1999/365   SHELF    1     UV
NSO013 SCHLUM         2008/222   1999/365   SHELF    1     UV
NSO014 SCHLUM         2008/215   1999/365   SHELF    1     UV
NSO015 SCHLUM         2008/229   1999/365   SHELF    1     UV
NSO016 SCHLUM         2008/236   1999/365   SHELF    1     UV
EDG3012I 12         ENTRIES LISTED
```

Figure 10-92 RMM SEARCHVOLUME example

In our next example, shown in Figure 10-93, we use the SINCE operand and a value of 2008/220. This operand was available prior to z/OS V1.11. You can see that all volumes with an assign date equal to or higher than 2008/220 are listed.

```
RMM SV VOLUME(*) OWNER(schlum) LIMIT(*) SINCE(2008/220)

Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                      date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
NS0003 SCHLUM          2008/221   1999/365   SHELF    1     UV
NS0005 SCHLUM          2008/228   1999/365   SHELF    1     UV
NS0006 SCHLUM          2008/235   1999/365   SHELF    1     UV
NS0013 SCHLUM          2008/222   1999/365   SHELF    1     UV
NS0015 SCHLUM          2008/229   1999/365   SHELF    1     UV
NS0016 SCHLUM          2008/236   1999/365   SHELF    1     UV
EDG3012I 6          ENTRIES LISTED
```

Figure 10-93 RMM SEARCHVOLUME example using the SINCE operand

The third example, shown in Figure 10-94, shows how you can use the start_date and end_date options to limit your search results and get only the volumes listed in the selected time period. We use the time command to show the date that we use the command. We use the RMM TSO SEARCHVOLUME subcommand with the ASDATE operand. The start_date is specified in an absolute date format and the end_date is specified in a relative date format. August 15,2009 is 2009/227 in the Julian date format, so only volumes in the date range between 2008/220 and 2008/227 are listed.

```
TIME
TIME-03:33:12 PM. CPU-00:00:01 SERVICE-45252 SESSION-01:00:57 AUGUST 15,2009

RMM SV VOLUME(*) OWNER(schlum) LIMIT(*) ASDATE(START(2008/220) END(-12M))

Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                      date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
NS0003 SCHLUM          2008/221   1999/365   SHELF    1     UV
NS0013 SCHLUM          2008/222   1999/365   SHELF    1     UV
EDG3012I 2          ENTRIES LISTED
```

Figure 10-94 RMM SEARCHVOLUME example using the ASDATE operand with start and end dates

10.6.4 VRS location definition handling

DFSMSrmm vital record specifications (VRS) handling is simplified by allowing location definitions to be changed after a VRS has been defined.

To change the bin management for a storage location, perform the following steps:

1. Identify the storage locations information you want to change.

2. Update the DFSMSrmm EDGRMMxx parmlib member LOCDEF command MANAGEMENTTYPE operand for a storage location. Figure 10-95 shows an example of this task.

```
LOCDEF          LOCATION(TEST) MEDIANAME(*) TYPE(STORAGE)                -
                MANAGEMENTTYPE(NOBINS)

LOCDEF          LOCATION(TEST) MEDIANAME(*) TYPE(STORAGE)                -
                MANAGEMENTTYPE(BINS)
```

Figure 10-95 EDGRMMnn PARMLIB member

3. Restart the DFSMSrmm subsystem using the MVS command, as shown in Figure 10-96.

```
F DFRMM,M=xx
```

Figure 10-96 Restart the DFSMSrmm subsystem

4. Use the RMM ADDBIN subcommand, shown in Figure 10-97, to add the required bins to the storage location if the new MANAGEMENTTYPE is BIN.

```
RMM ADDBIN TOOOOO LOCATION(TEST) COUNT(500) MEDIANAME(*)
```

Figure 10-97 RMM TSO ADDBIN subcommand

5. If you changed the management type for a location defined with TYPE(STORE,HOME), perform the following additional steps:

   a. Issue the RMM CHANGEVOLUME subcommand with the HOME operand for all the volumes that have the changed location defined as their home location. This is done to correctly update the home location information previously saved in the volume records. Use the command show in Figure 10-98.

```
RMM SEARCHVOLUME VOLUME(*) OWNER(*) LIMIT(*) LOCATION(TEST) -
    CLIST('          RMM CV ',' HOME(TEST)')
```

Figure 10-98 RMM TSO CHANGEVOLUME subcommand

   b. Execute the created RMM TSO CHANGEVOLUME subcommands stored in the prefix.EXEC.RMM.CLIST data set. Normally, the prefix is the user ID of the user. Figure 10-99 shows the sample job control used to process the create RMM TSO CHANGEVOLUME subcommands in a batch job.

```
//TESTCASE EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   DISP=SHR,DSN=prefix.EXEC.RMM.CLIST
```

Figure 10-99 Execute CHANGEVOLUME subcommands in a batch job

6. Perform inventory management vital record processing and storage location management to move volumes to the updated storage location. Figure 10-100 shows a sample job control that can be used to process these two functions in one single job step.

```
//STEP0001 EXEC PGM=EDGHSKP,
//         PARM=('VRSEL,DSTORE,DATEFORM(J)')
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   DSN=RMM.HSKP.MESSAGE,DISP=SHR
//REPORT   DD   DSN=RMM.HSKP.REPORT,DISP=SHR
//ACTIVITY DD   DSN=RMM.HSKP.ACTIVITY,DISP=SHR
```

Figure 10-100 Sample housekeeping job control

10.6.5 Rexx variables

Only the key variable for each SEARCH, and LISTCONTROL subcommand has a ".0" stem variable. Our code is updated to only create the ".0" stem variable for the key variables and special stem variables, as shown in Table 10-15. All other non-key and non-special stem.0 variables are deleted.

Table 10-15 shows all the stem.0 variables that can be used with DFSMSrmm z/OS V1.11 or higher. All other previously created stem.0 variables are no longer supported.

Table 10-15 List of stem.0 variables

Command	Key stem	Special stem
LISTCONTROL ACTIONS	edg@act	
LISTCONTROL LOCDEF	edg@ldlc	edg@ldmn
LISTCONTROL SECLVL	edg@dnm	
LISTCONTROL MEDINF	edg@mdnf	
LISTCONTROL MNTMSG	edg@mid	
LISTCONTROL MOVES	edg@mfr	
LISTCONTROL OPENRULE	edg@ortp	
LISTCONTROL PARTITION	edg@pttp	
LISTCONTROL REJECT	edg@grk	
LISTCONTROL SECCLS	edg@cls	
LISTCONTROL VLPOOL	edg@pid	
LISTPRODUCT		edg@vol
SEARCHBIN	edg@rck	
SEARCHDATASET	edg@dsn	
SEARCHOWNER	edg@own	
SEARCHPRODUCT	edg@pnum	
SEARCHRACK	edg@rck	

Command	Key stem	Special stem
SEARCHVOLUME	edg@vol	
SEARCHVRS	edg@vrs	

Note: Removing the .0 stem variables benefits performance in multiple ways, for example, there are fewer calls to Rexx to create and update variables, and there is less virtual storage used, which should lead to lower CPU and elapsed time. The difference will be small and likely not measurable as a significant benefit.

10.6.6 DFSMSrmm journaling

Using a new parmlib option, DFSMSrmm now writes the previous level of the updated record to the journal along with the updated record and updated, related records.

JRNLTRAN(NO|YES)

Use the JRNLTRAN operand to specify whether the unchanged copy of a record is journaled, as well as the updated copy. You can specify one of the following items:

NO Only the updated records are written to the journal file.

YES Additional journal records are written to the journal file. The additional record is a pre-update copy of the CDS record being updated. You should only set this option on a test or recovery system when you plan to exploit the EDGUPDT utility to duplicate CDS record updates in the production CDS. As a result of using this option, you should plan on providing up to 33% more journal data set space to accommodate the additional records.

Figure 10-101 shows the correct use of the JRNLTRAN operand.

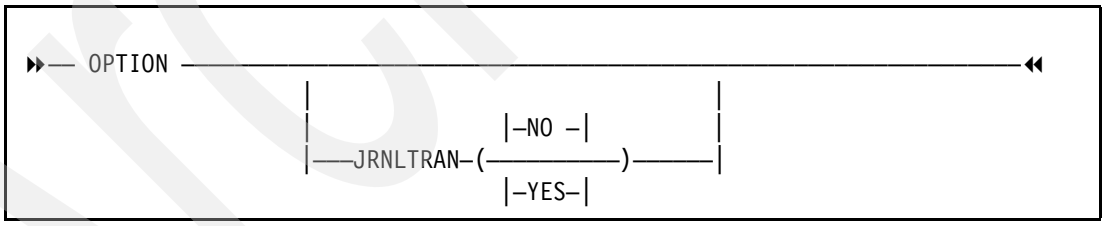


Figure 10-101 JRNLTRAN operand

You can use the RMM TSO LISTCONTROL OPTION subcommand to see the current setting, as shown in Figure 10-102.

```
System options:
PARMLIB Suffix  = NS
Operating mode  = P    Retention period: Default = 0       Maximum = NOLIMIT
                                         Catalog = 6     hours
Control data set name      = RMM.CONTROL.DSET
Journal file data set name = RMM.JOURNAL.DSET
Journal threshold          = 75%               Journal transaction = NO
```

Figure 10-102 LISTCONTROL OPTION subcommand

10.6.7 Using the EDGUPDT utility

Use the EDGUPDT utility with the UPDATE parameter to update the active DFSMSrmm CDS with record updates created during testing or recovery. EDGUPDT uses a journal or journal backup or a concatenation of these data sets, which were created by DFSMSrmm running with the JRNLTRAN(YES) option. Using EDGUPDT ensures that, even after testing or recovery exercises, that the DFSMSrmm CDS reflects the actual content of the tape volumes in your library. The DFSMSrmm started procedure must be active when you run this utility.

When you prepare to run a test or recovery system with JRNLTRAN(YES), you create a copy of the current production CDS, perhaps by using the EDGHSKP/EDGBKUP utility with the BACKUP(COPY) parameter. Before you start DFSMSrmm on the test/recovery system, create a new journal data set so that only the journal records created during the current test/recovery are available for use with EDGUPDT after all or part of testing/recovery is completed. You could, for example, back up the test/recovery system journal regularly and use the backup of the journal with EDGUPDT to maintain the production CDS. At the end of the test, complete any required updates using the active test/recovery journal.

The EDGUPDT utility can only apply updates from the journal if the records have not been changed in the original CDS. Utility processing ensures that the CDS record has not been changed, deleted, or added since the CDS copy was created.

If any errors occur when running EDGUPDT as a result of records being changed in the original CDS, processing continues, but skips the update from the journal. Information messages are issued. At the end of processing, a summary of record updates is written to the SYPRINT file. If you rerun EDGUPDT using the same JOURNAL file to recover from a previous run error, you should expect an error message for each record that was successfully processed on the previous run.

DFSMSrmm processing ensures that only the volume and data set information in the CDS reflects processing in the test/recovery environment. DFSMSrmm does not make any changes to or validates that other, related system data sets are correct. For example;

- TCDB: There is no consideration of whether a volume entry exists or not, or that the volume status remains in sync with the DFSMSrmm status of the volume.
- UCAT: There is no consideration of whether data sets that were cataloged on the test/recovery system have corresponding catalog entries in the production environment.

After your EDGUPDT processing is complete, you can optionally cross-check the DFSMSrmm CDS with related system data by:

1. Running EDGUTIL with PARM='VERIFY(SMSTAPE)'

   You can optionally limit the scope of the cross-checking by using the SYSIN DD statement to specify include and exclude conditions for the volumes to be processed.

2. Running EDGHSKP with PARM='VERIFY,CATSYNCH'

   Messages are issued during processing for data sets that are missing from the user catalogs but defined as cataloged in DFSMSrmm.

The IDCAMS utility can be used to maintain either volume entries in the TCDB or data set catalog entries in the user catalogs.

Figure 10-103 shows you the sample job control can be used to update the active DFSMSrmm control data set (CDS) based on journal records coming from a test system.

```
//UPDATE   EXEC PGM=EDGUPDT,PARM=UPDATE
//SYSPRINT DD   SYSOUT=*
//JOURNAL  DD   DISP=SHR,DSN=TEST.SYSTEM.JRNL
//SYSIN    DD   *
UPDATE TARGETDATE(2009/123,12:30:00)
/*
```

Figure 10-103 Sample EDGUPDT job control

Where:

JOURNAL Identifies the journal backups and the journal to be used during update processing. When you concatenate multiple data sets, do so from oldest to newest.

SYSPRINT Contains the utility program messages issued when updating the DFSMSrmm control data set. This data set can be a SYSOUT file.

SYSIN Use the SYSIN file to select the journal records used for updating the CDS.

SYSIN file for EDGUPDT

Use this file to select the target point in time for update processing from the journal input records.

Update processing starts at the first record in the JOURNAL. Update processing continues until the point in time you select, or to the end of the input records, whichever occurs first.

Figure 10-104 shows the correct use of the JRNLTRAN operand.

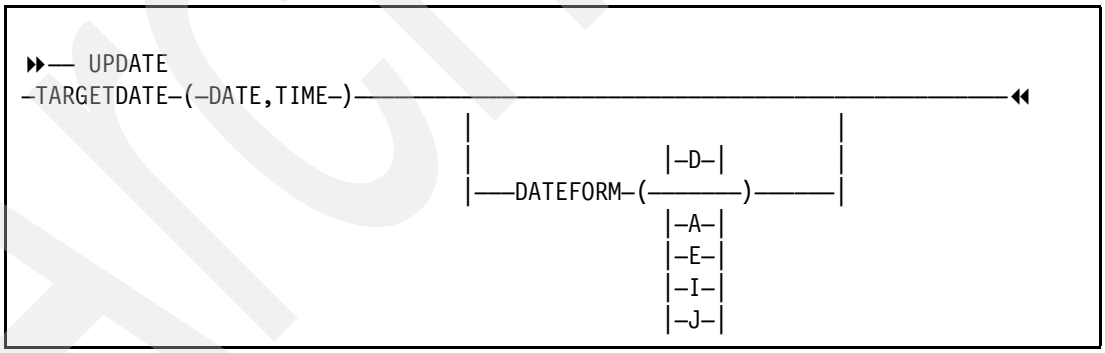


Figure 10-104 EDGUPDT syntax

Where:

UPDATE Use this command in the SYSIN file to select your chosen endpoint for journal update processing. When you do not specify UPDATE, DFSMSrmm uses all the available complete sets of records to perform update processing to the latest available point in time.

TARGETDATE Use the TARGETDATE(Date, Time) operand to specify the date and time that you have selected to be your endpoint in time for update processing. You must specify both a date and a time. The date and time are the local date and time recorded in the journal records used for update processing. Any input records created at a time higher than the value you specify are skipped.

Date Specify the date in the format set by the DATEFORM operand or (if the DATEFORM operand is not specified) by the DATEFORM parmlib value. For example, if your installation is set to DATEFORM(J), specify TARGETDATE(2008/123,23:22:10.0).

Time Specify the target local time in the format hh:mm:ss.t The .t (tenths of a second) is optional.

DATEFORM Use the optional DATEFORM (A | D| E | I | J) operand to specify the format for the date you enter in the TARGETDATE operand. Table 10-16 shows the correct spelling of the different date formats.

Table 10-16 DFSMSrmm supported date formats

Value	Language	Format	Example
A	American	mm/dd/yyyy	12/08/2009
E	European	dd/mm/yyyy	15/08/2009
I	ISO	yyyy/mm/dd	2009/08/15
J	Julian	yyyy/ddd	2009/227
D	Default	The installation's default in EDGRMMxx	Initially set to Julian

The default date format for all date fields is the value specified in the parmlib member EDGRMMxx. The value is initially set to J for Julian. To change the date format for each run of EDGUPDT, use the DATEFORM parameter.

Important: EDGBKUP RESTORE uses journal records to forward recover the CDS, but ignores the additional, pre-update level of the changed record. EDGUPDT with the UPDATE parameter uses a journal or journal backup or a concatenation of these data sets to update the active DFSMSrmm CDS with record updates created during testing or recovery.

RACF authorization

You can protect this new functionality by adding a new resource to the FACILITY RACF class. Table 10-17 shows the new resources that you can define and the updates to them.

Table 10-17 RACF access to class FACILITY

When you define	With access	Then
STGADMIN.EDG. EDGUPDT.UPDATE	Entity not defined	Same as the UPDATE access
	NONE	No authority is granted to use the EDGUPDT UPDATE function.
	UPDATE	You can use the EDGUPDT UPDATE function.

10.6.8 API multi-entry return

You can use a new keyword with 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, use the REQUEST=CONTINUE keyword.

The getBufferXml and getBufferSfi methods can return multiple resources in a buffer; be sure to process all the returned data (XML or SFIs) before using the getNextEntry method, as more entries may exist.

The SSOB extension size is set to 64K-8, allowing at least double the number of volume entries to be returned.

EDGXCI parameters

You can use a new 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.

The parameters are explained as follows:

MULTI When OUTPUT=FIELDS and OPERATION=BEGIN are specified, an optional parameter that specifies whether a single resource group is to be returned in the buffer, or whether as many resources as fit in the buffer, are to be returned. The default is MULTI=NO.

- **,MULTI=NO** Indicates that only a single entry can be handled by the API caller.
- **,MULTI=YES** Indicates that multiple entries can be handled by the API caller.

Requesting multiple resources for SEARCH subcommands

The DFSMSrmm API can return resources either one at a time or multiples at a time when you specify one of the DFSMSrmm TSO RMM SEARCHDATASET, SEARCHBIN, SEARCHOWNER, SEARCHPRODUCT, SEARCHRACK, SEARCHVOLUME, or SEARCHVRS subcommands together with OUTPUT=FIELDS. Use the MULTI keyword to notify the API about which type of output you can handle. In order to specify MULTI=YES, your application must be able to handle multiple resources, each separated by the begin/end group SFIs. When you specify MULTI=YES, your output buffer can have one or more resource groups returned in a single call of the API. Using MULTI=YES helps reduce the system resources used for API processing.

10.7 VRSEL(OLD) parmlib option

The VRSEL(OLD) option is no longer supported. All information related to the use of VRSEL(OLD) has been removed. If you have not yet migrated to the VRSEL(NEW) option, refer to a previous edition of this book (SC26-7404-08 or earlier) for information about migrating to VRSEL(NEW), or you can use the information in this section.

The DFSMSrmm parmlib option VRSEL(OLD) has been removed in z/OS V1.11 DFSMSrmm. You should migrate from VRSEL(OLD) to VRSEL(NEW) before moving to z/OS V1.11. Each time you run VRSEL processing and VRSEL(OLD) is in use, the message EDG2317E and a minimum return code of 4 are issued:

```
EDG2317E MIGRATION FROM VRSEL(OLD) TO VRSEL(NEW) IS RECOMMENDED.
```

Where:

Severity	Warning.
Explanation	During inventory management vital record processing, DFSMSrmm checks the VRSEL option that you have defined in the EDGRMMxx parmlib. The VRSEL(OLD) option will be removed in a future release of z/OS. Please migrate to VRSEL(NEW).
Source	DFSMSrmm.
Detecting Module	EDGMHKP.
System Action	Processing continues. A minimum return code of 4 is set.
Operator Response	None.
Sysprog Response	Plan a migration to VRSEL(NEW). Refer to the migration planning steps documented in the *DFSMSrmm Implementation and Customization Guide,* SC26-7405.
Routing Codes	11.
Descriptor Codes	None.

10.7.1 Migrating from VRSEL(OLD) to VRSEL(NEW)

Because the VRSEL(OLD) option is being removed in a future release, and there is a new warning message when you run VRSEL processing on z/OS V1.8, migrate from VRSEL(OLD) to VRSEL(NEW) before moving to z/OS V1.8. If you do not perform this migration, you will get a warning message (EDG2221E) each time VRSEL is run, and EDGHSKP processing ends with job step return code 4.

Any data sets that match to an incomplete chain are retained by a special broken vital record specification. The special broken vital record specification uses the name *broken* and is listed in the REPORT and ACTIVITY files and in the data set matching vital record specification information. With VRSEL(NEW), the broken vital record specification uses a permanent retention date. With VRSEL(OLD), the broken vital record specification ensures that the data set is retained to the maximum of the COUNT value in the first vital record specification.

Use VRSEL(NEW) to:

- Have more flexibility in defining retention and movement policies
- Override a volume expiration date when a volume is dropped from vital record specification retention and the data set retained on it has the release option EXPIRYDATEIGNORE
- Return a volume to scratch status in a single inventory management run

Before using VRSEL(NEW), understand that your existing policies might be applied differently under VRSEL(NEW). You can perform the following steps to avoid problems that might occur when the DFSMSrmm performs VRSEL(NEW) processing:

1. Before using VRSEL(NEW), back up your DFSMSrmm control data set using EDGBKUP or EDGHSKP.
2. Perform a cleanup of the name vital record specifications by making sure that any retention information in them is correct. DFSMSrmm provides the EDGRVCLN REXX exec described in the section “EDGRVCLN REXX” in Chapter 5, “Defining retention and movement policies”, in *DFSMSrmm Managing and Using Removable Media*, SC26-7404, to report and clean up problems with name vital record specifications.
3. Run DFSMSrmm inventory management vital record processing so that the DFSMSrmm control data set reflects the cleanup you have done.
4. Update the DFSMSrmm parmlib OPTION VRSEL(NEW) operand.
5. Make sure that all systems sharing DFSMSrmm control data sets have the same parmlib options.
6. Run the inventory management VERIFY function against the control data set without introducing any of the new vital record specification functions. When you run VERIFY, changes are not actually made to the DFSMSrmm control data set, so you can look at the results before any changes are made.
7. Inspect the inventory management VERIFY ACTIVITY file by looking at changes in the matching vital record specification information, vital record status, and retention date. DFSMSrmm provides a sample job (EDGJACTP) that you can use with DFSORT to format and print fields in the ACTIVITY file. If you cannot clean up to your satisfaction, you can revert to VRSEL(OLD) at this time.
8. Correct vital record specifications as needed to make sure that the policies you want are in place.
9. Continue running the VERIFY function and inspect the results until you get the results you expect using the new functions.
10. Begin defining vital record specifications that include the new release options or the use of ANDVRS. We suggest starting slowly until you gain more experience with using the new vital record specification functions. We recommend using the RMM ISPF panel to add the new vital record specifications or to make changes to existing vital record specifications.
11. Repeat the process from step 7, step 9, and step 10 until you are satisfied that the results are what you expect.
12. Run inventory management production run processing.
13. Check the vital records retention report to make sure that data sets and volumes are retained as you intended.

EDGRVCLN REXX exec

Before you can use VRSEL(NEW), you need to clean up the retention information in your existing vital record specifications. Use the EDGRVCLN REXX procedure to report and update existing vital record specifications that were created before VRSEL(NEW) was introduced. The procedure is intended for use only during the implementation of name vital record specification retention information and for cleanup of vital record specifications, even if name vital record specification retention information is not being implemented. EDGRVCLN provides options that you can use to identify retention information that needs to be corrected and options that you can use to correct the information.

The REXX exec has following functions:

LIST(DSNCHAIN) Lists all vital record specification chains.

LIST(CURRENT) Lists all vital record specifications that specify LOCATION(CURRENT).

LIST(CYCLES) Lists all vital record specifications that specify a retention type of CYCLES.

LIST(ERROR) Lists all name vital record specifications that contain incorrect or incomplete retention information.

LIST(FILTER) Lists all data set VRSs that are candidates for exploitation of the use of COUNT(0).

FIX(CYCLEBYDAYS) Changes all CYCLES vital record specifications to use the CYCLEBYDAYS retention type.

FIX(ERROR) Corrects all name vital record specifications that contain incorrect or incomplete retention information found by LIST(ERROR).

FIX(FILTER) Implements the changes recommend by LIST(FILTER).

EDGRVCLN LIST file

When you use one of the LIST parameters, the procedure produces the DFSMSrmm ADDVRS subcommands that can be issued to define corrected vital record specifications. When you use the LIST(CURRENT), LIST(ERROR), or LIST(CYCLES) parameters, the `LIST` file includes an RMM DELETEVRS subcommand for each vital record specification in addition to the ADDVRS subcommand. When you use the LIST(FILTER) parameter, the `LIST` file includes an RMMCHANGEVRS and one or more ADDVRS subcommands for each vital record specification that is a candidate for exploitation of the use of COUNT(0). If you do not want to use the FIX parameters to correct all the identified vital record specifications, you can use the `LIST` file as input to your own processing. You can edit the file to remove or modify the commands to meet your specific requirements.

EDGRVCLN LIST(DSNCHAIN) parameter

Use this parameter to list all vital record specification chains.

The procedure searches for all data set vital record specifications. The procedure then searches for all name vital record specifications in the chain.

Use the JCL shown in Example 10-3 to list all vital record specification chains.

Example 10-3 Sample JCL to use EDGRVCLN with parameter LIST(DSNCHAIN)

```
//EDGRVCLN JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.CHAINP    NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.CHAIN    NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CHAINP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CHAIN,
```

```
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(DSNCHAIN)
/*
```

Figure 10-105 shows you the messages EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(DSNCHAIN)
SEARCHVRS DSNAME 'SCHLUM.TAPE.**' CHAIN  failed with Return Code 4, Reason Code
Error in the CDS
READY
END
```

Figure 10-105 Sample SYSTSPRINT messages using EDGRVCLN with the LIST(DSNCHAIN) parameter

The commands are generated using the LIST(DSNCHAIN) parameter are shown in Figure 10-106.

```
        RMM ADDVRS DSNAME('A.B.C') COUNT(99999) DELETEDATE(1999/365)+
            PRIORITY(0) OWNER(SCHLUM) STORENUMBER(99999) +
            LOCATION(HOME) CYCLES
...
        RMM ADDVRS DSNAME('SCHLUM.RMMDEMO.**') COUNT(1) +
            DELETEDATE(1999/365) +
            DESCRIPTION('MVSSSC SYSTEM BACKUP (PROD)') +
            PRIORITY(0) NEXTVRS(EXTRACAT) OWNER(SCHLUM) STORENUMBER(1) +
            LOCATION(HOME) DAYS +
        RMM ADDVRS NAME(EXTRACAT) DELETEDATE(1999/365) +
            DESCRIPTION('TEST') OWNER(SCHLUM) STORENUMBER(99999) +
            LOCATION(CURRENT)
        RMM ADDVRS DSNAME('SCHLUM.RMMTEST.MOVE.**') COUNT(99999) +
            DELETEDATE(1999/365) +
            DESCRIPTION('RETAIN AND MOVE DATA SETS') +
            PRIORITY(0) OWNER(SCHLUM) STORENUMBER(1) WHILECATALOG +
            LOCATION(REMOTE) CYCLES
```

Figure 10-106 Commands created using EDGRVCLN with the LIST(DSNCHAIN) parameter

EDGRVCLN LIST(CURRENT) parameter

Use this parameter to list all vital record specifications that specify LOCATION(CURRENT).

The procedure searches for all vital record specifications and lists each one that includes the LOCATION(CURRENT).

Use the JCL shown in Example 10-4 to list all vital record specification that specify LOCATION(CURRENT).

Example 10-4 Sample JCL to use EDGRVCLN with parameter LIST(CURRENT)

```
//EDGRVCLN JOB ,14O.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
```

```
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.CURRENTP NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.CURRENT  NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CURRENTP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CURRENT,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(CURRENT)
/*
```

Figure 10-107 shows you the messages EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(CURRENT)
4 VRSs with the LOCATION(CURRENT) LISTed.
READY
END
```

Figure 10-107 Sample SYSTSPRINT messages using EDGRVCLN with the LIST(CURRENT) parameter

The commands that are generated using the LIST(CURRENT) parameter are shown in Figure 10-108.

```
        RMM DELETEVRS DSNAME('DATA.SET.BACKUP')
        RMM ADDVRS DSNAME('DATA.SET.BACKUP') +
            COUNT(99999) DELETEDATE(1999/365) DESCRIPTION('TEST') +
            PRIORITY(0) OWNER(SCHLUM) STORENUMBER(99999) WHILECATALOG +
            LOCATION(CURRENT) CYCLES
        RMM DELETEVRS DSNAME('D99003')
        RMM ADDVRS DSNAME('D99003') COUNT(3) +
            DELETEDATE(1999/365) DESCRIPTION('TEST') +
            PRIORITY(0) OWNER(SCHLUM) STORENUMBER(3) +
            LOCATION(CURRENT) LASTREFERENCEDAYS
        RMM DELETEVRS NAME(CATALOG)
        RMM ADDVRS NAME(CATALOG) DELETEDATE(1999/365) +
            DESCRIPTION('TEST') OWNER(SCHLUM) STORENUMBER(99999) +
            LOCATION(CURRENT)
        RMM DELETEVRS NAME(EXTRACAT)
        RMM ADDVRS NAME(EXTRACAT) DELETEDATE(1999/365) +
            DESCRIPTION('TEST') OWNER(SCHLUM) STORENUMBER(99999) +
            LOCATION(CURRENT)
```

Figure 10-108 Commands created using EDGRVCLN with the LIST(CURRENT) parameter

EDGRVCLN LIST(CYCLES) parameter

Use this parameter to list all vital record specifications that specify a retention type of CYCLES. This includes those vital record specifications that include WHILECATALOG where CYCLES is used as the default retention type.

The procedure searches for all data set vital record specifications and lists each one that includes the CYCLE retention type.

Use the JCL shown in Example 10-5 to list all vital record specification that specify a retention type of CYCLES.

Example 10-5 Sample JCL to use EDGRVCLN with parameter LIST(CYCLES)

```
//EDGRVCLN JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.CYCLESP  NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.CYCLES   NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CYCLESP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.CYCLES,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(CYCLES)
/*
```

Figure 10-109 shows you the messages EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(CYCLES)
24 DSNAME VRSs with CYCLES retention type LISTed.
READY
END
```

Figure 10-109 Sample SYSTSPRINT messages using EDGRVCLN with the LIST(CYCLES) parameter

The commands that are generated using the LIST(CYCLES) parameter are shown in Figure 10-110.

```
        RMM DELETEVRS DSNAME('A.B.C')
        RMM ADDVRS DSNAME('A.B.C') COUNT(99999) +
            DELETEDATE(1999/365) PRIORITY(0) OWNER(SCHLUM) +
            STORENUMBER(99999) +
            LOCATION(HOME) BYDAYSCYCLE
....
        RMM DELETEVRS DSNAME('SCHLUM.EDG*.**')
        RMM ADDVRS DSNAME('SCHLUM.EDG*.**') COUNT(6) +
            DELETEDATE(1999/365) DESCRIPTION('TEST') +
            PRIORITY(0) OWNER(SCHLUM) STORENUMBER(6) +
            WHILECATALOG RELEASE(EXPIRYDATEIGNORE SCRATCHIMMEDIATE) +
            LOCATION(HOME) BYDAYSCYCLE
        RMM DELETEVRS DSNAME('SCHLUM.MULTIPLE.DATASET.**')
        RMM ADDVRS DSNAME('SCHLUM.MULTIPLE.DATASET.**') +
            COUNT(1) DELETEDATE(1999/365) DESCRIPTION('TEST')+
            PRIORITY(0) OWNER(SCHLUM) STORENUMBER(1) +
            LOCATION(HOME) BYDAYSCYCLE
        RMM DELETEVRS DSNAME('SCHLUM.RMMTEST.MOVE.**')
        RMM ADDVRS DSNAME('SCHLUM.RMMTEST.MOVE.**') COUNT(99999) +
            DELETEDATE(1999/365) +
            DESCRIPTION('RETAIN AND MOVE DATA SETS') +
            PRIORITY(0) OWNER(SCHLUM) STORENUMBER(1) WHILECATALOG +
            LOCATION(REMOTE) BYDAYSCYCLE
        RMM DELETEVRS DSNAME('SCHLUM.TAPE.**')
        RMM ADDVRS DSNAME('SCHLUM.TAPE.**') COUNT(1) +
            DELETEDATE(1999/365) DESCRIPTION('TEST NEXT VRS') +
            PRIORITY(0) NEXTVRS(EXTRA) OWNER(SCHLUM) STORENUMBER(1) +
            LOCATION(HOME) BYDAYSCYCLE
        RMM DELETEVRS DSNAME('SCHLUM.TEST1')
        RMM ADDVRS DSNAME('SCHLUM.TEST1') COUNT(1) +
            DELETEDATE(1999/365) PRIORITY(0) OWNER(SCHLUM) +
            STORENUMBER(1) +
            LOCATION(HOME) BYDAYSCYCLE
....
```

Figure 10-110 Commands created using EDGRVCLN with the LIST(CYCLES) parameter

EDGRVCLN LIST(ERROR) parameter

Use this parameter to list all name vital record specifications that contain incorrect or incomplete retention information. The procedure assumes that any name vital record specification containing a retention type or count value is in error. This is the default value.

The procedure searches for all name vital record specifications and lists each one that includes any retention type or count value.

Use the JCL shown in Example 10-6 to list all vital record specification that contain incorrect or incomplete retention information.

Example 10-6 Sample JCL to use EDGRVCLN with the LIST(ERROR) parameter

```
//EDGRVCLN JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
```

```
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.ERRORP   NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.ERROR    NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.ERRORP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.ERROR,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(ERROR)
/*
```

Figure 10-111 shows you the messages EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(ERROR)

VRS NAME(CATALOG) is a valid location-only VRS

VRS NAME(D99000) is a valid location-only VRS

VRS NAME(EXTRACAT) is a valid location-only VRS
VRS NAME(TESTAND) had retention values and has been recreated as a
    location-only VRS

VRS NAME(TESTNEXT) is a valid location-only VRS

VRS NAME(TESTX) is a valid location-only VRS
1 NAME VRSs in error are LISTed.
5 NAME VRSs were valid location-only VRSs.
READY
END
```

Figure 10-111 Sample SYSTSPRINT messages using EDGRVCLN with the LIST(ERROR) parameter

The commands that are generated using the LIST(ERROR) parameter are shown in Figure 10-112.

```
      RMM DELETEVRS NAME(TESTAND)
      RMM ADDVRS NAME(TESTAND) DELETEDATE(1999/365) +
          DESCRIPTION('TEST LCLVRS1 AND LCLVRS2 FUNCT') +
          ANDVRS(TESTX) OWNER(SCHLUM) STORENUMBER(99999) +
          LOCATION(SHELF)
```

Figure 10-112 Commands created using EDGRVCLN with the LIST(ERROR) parameter

EDGRVCLN LIST(FILTER) parameter

Use this parameter to list all data set VRSs that are candidates for use by COUNT(0).

The procedure searches for all data set name VRSs that do not specify COUNT(0) and groups the ones that have common delay, retention, and movement requirements, and NEXT/ANDVRS specified in the first VRS in the chain.

Use the JCL shown in Example 10-7 to list all data set VRSs that are candidates for use by COUNT(0).

Example 10-7 Sample JCL to use EDGRVCLN with the LIST(DSNCHAIN) parameter

```
//EDGRVCLN JOB ,140.SCHLUMBERGER,MSGCLASS=H,REGION=6M,
//         MSGLEVEL=(1,1),NOTIFY=SCHLUM
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE RMM.EDGRVCLN.LIST.FILTERP  NONVSAM PURGE
  DELETE RMM.EDGRVCLN.LIST.FILTER   NONVSAM PURGE
  SET MAXCC=0
/*
//TMPCHAIN EXEC PGM=IKJEFT01,DYNAMNBR=30
//SYSPROC  DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//SYSTSPRT DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.FILTERP,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//LIST     DD   DISP=(,CATLG),DSN=RMM.EDGRVCLN.LIST.FILTER,
//         SPACE=(TRK,(15,5),RLSE),UNIT=SYSDA,
//         DCB=(LRECL=255,RECFM=VB,BLKSIZE=0)
//SYSTSIN  DD   *
     %EDGRVCLN LIST(FILTER)
/*
```

Figure 10-113 shows you the messages EDGRVCLN writes to the SYSTSPRT DD.

```
READY
     %EDGRVCLN LIST(FILTER)

Dsname VRSs with retention type CYCLE/BYDAYSCYCLE with the combination of
DELAY() and STORENUMBER() can not be converted.
The following have the same retention criteria and could be converted if
you manually change the VRSs not to have DELAY and STORENUMBER in one VRS:
JOBNAME  DSNAME
TESTYY   TESTYY
TEST*    TESTZZ

RMM commands prepared for the exploitation of the use of COUNT(0) are LISTed.
READY
END
```

Figure 10-113 Sample SYSTSPRINT messages using EDGRVCLN with the LIST(FILTER) parameter

The commands that are generated using the LIST(FILTER) parameter are shown in Figure 10-114.

```
RMM CHANGEVRS DSNAME('TEST.LCLVRSC.FUNCTION') +
              COUNT(0) +
              NEXTVRS(A0000000)

RMM CHANGEVRS DSNAME('TEST.LCLVRSC.FUNCTION.**') +
              COUNT(0) +
              NEXTVRS(A0000000)

RMM ADDVRS    NAME(A0000000) +
              COUNT(99999) LASTREFERENCEDAYS +
              LOCATION(DISTANT) STORENUMBER(12345) +
              WHILECATALOG UNTILEXPIRED


RMM CHANGEVRS DSNAME('ABEND') +
              JOBNAME(SCHLUM) +
              COUNT(0) +
              NEXTVRS(A0000004)

RMM CHANGEVRS DSNAME('ABEND') +
              JOBNAME(SCHLUX) +
              COUNT(0) +
              NEXTVRS(A0000004)

RMM CHANGEVRS DSNAME('OPEN') +
              JOBNAME(SCHLUM) +
              COUNT(0) +
              NEXTVRS(A0000004)

RMM ADDVRS    NAME(A0000004) +
              COUNT(1) DAYS +
              LOCATION(HOME)


RMM CHANGEVRS DSNAME('SIEGEL.TEST.VRSEL.CATYN') +
              COUNT(0) +
              NEXTVRS(A0000005)

RMM CHANGEVRS DSNAME('SIEGEL.TEST1') +
              COUNT(0) +
              NEXTVRS(A0000005)

RMM ADDVRS    NAME(A0000005) +
              COUNT(3) DAYS +
              LOCATION(HOME)
```

Figure 10-114 Commands created using EDGRVCLN with the LIST(FILTER) parameter

FIX(CYCLEBYDAYS)

Use this parameter to change all CYCLES vital record specifications to use the CYCLEBYDAYS retention type.

FIX(ERROR)

Use this parameter to correct all name vital record specifications that contain incorrect or incomplete retention information found by LIST(ERROR). The corrections are made by deleting and re-adding the vital record specifications that contain errors. Only use this option when you are ready to correct the errors listed by the LIST(ERROR) option.

FIX(FILTER)

Use this parameter to implement the changes recommended by LIST(FILTER).

The procedure creates one or more retention NAME VRSs for each of the common groups of retentions found, and then uses the CHANGEVRS subcommand to set COUNT(0) and the NEXTVRS to chain to the new NAME VRS(es) created. If DELAY was in use, a retention name VRS is created for the DELAY using DAYS since creation and a NEXTVRS to the retention name VRS(es) for the retention and location pulled from the DSNAME VRS.

10.8 Migration considerations

In this section, we give you an overview about what has changed in z/OS V1.11 and what actions you must perform initially loading the program.

10.8.1 Report generator

A migration action is required if you want to exploit the new capability within the report types for existing report definitions.

After the program is initially loaded, you can use the DFSMSrmm report generator Report Migration Tasks panel to import new information shipped in report types into existing report definitions.

After you have completed this migration task, do not share the updated report definitions with a lower release of the DFSMSrmm report generator.

10.8.2 VRSEL GDG options

You should consider these options before the program is initially loaded.

The defaults for the GDG parmlib options are the same as the processing performed by previous releases. However, any customer with a USERMOD applied to the EDGVREC load module which sets the ZpSwitches to influence duplicate GDG handling must now use the parmlib options instead.

The USERMOD may be called RMDUPGD. The previous version of USERMOD no longer works with EDGVREC in z/OS V1.11 because the switch setting initialization has been changed. This means any attempt to use SMP/E APPLY or to use AMASPZAP will fail and you will need to refit the USERMOD. Section “DFSMSrmm: Use DFSMSrmm parmlib options instead of USERMOD to specify how VRSEL processing handles duplicate GDGs” in *z/OS V1R11 Migration*, GA22-7499 provides more information about refitting the RMDUPGD USERMOD.

If the usermod is refitted and used, a message is issued to the MESSAGE file that warns you that the usermod is no longer supported. Figure 10-115 shows this message.

```
EDGX004I USERMOD HAS BEEN DETECTED AND IGNORED – USE THE GDG(CYCLEBY(GEN)
         DUP(COUNT)) PARMLIB OPTION INSTEAD
```

Figure 10-115 EDGX004I message

10.8.3 CBRUXVNL user exit

There are no specific migration considerations for this function.

In order to exploit new choices within the CBRUXVNL exit, you can perform an optional migration action the program is initially loaded. You can update the assembly time variables via a SMP/E USERMOD. See 10.3, “Returning volumes to the system-managed library” on page 353 for more information.

There are no specific coexistence considerations for this function.

10.8.4 Rexx variables

You should consider these variables before the program is initially loaded.

If you remove DFSMSrmm Rexx stem “.0” variables, you must verify that any Rexx execs used with the DFSMSrmm subcommands are not impacted.

Use the supplied sample JCL from SAMPLIB (EDGJSTM0) shown in Figure 10-116 to check if your execs are affected.

```
//SEARCH  EXEC PGM=ISRSUPC,PARM=(SRCHCMP,'ANYC')
//NEWDD   DD   DISP=SHR,DSN=RMM.ADDONS.CEXEC
//        DD   DISP=SHR,DSN=RMM.LCLTCT.CEXEC
//OUTDD   DD   SYSOUT=*
//SYSIN   DD   *
    SRCHFOR  'EDG§OVOL.0'
....
/*
```

Figure 10-116 EDGJSTMO JCL sample

The function ended with a return code of 1. The report also tells you that one of the specified variables was found and that you can see the member and the line where the variable was found. Figure 10-117 shows you a sample report of a scan of two Rexx exec libraries.

```
  ISRSUPC   -   MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF FOR z/OS
 LINE-#  SOURCE SECTION                    SRCH DSN: SCHLUM.LCLTCT.CEXEC.SOURCE


 TESTLC   CONCAT#(2)       ----------- STRING(S) FOUND ------------------

      2  00030000edg§ovol.0

 TESTUC   CONCAT#(2)       ----------- STRING(S) FOUND ------------------

      2  00020000EDG§OVOL.0

  ISRSUPC   -   MVS/PDF FILE/LINE/WORD/BYTE/SFOR COMPARE UTILITY- ISPF FOR z/OS
     SEARCH-FOR SUMMARY SECTION           SRCH DSN: SCHLUM.LCLTCT.CEXEC.SOURCE

LINES-FOUND  LINES-PROC  MEMBERS-W/LNS  MEMBERS-WO/LNS  COMPARE-COLS
LONGEST-LI
        2       218425            2            327          1:251          89

PROCESS OPTIONS USED: ANYC

THE FOLLOWING PROCESS STATEMENTS (USING COLUMNS 1:72) WERE PROCESSED:
       SRCHFOR  'EDG§OVOL.0'
```

Figure 10-117 EDGJSTMO sample report

Important: If any Rexx exec is identified as affected, you must replace the affected variable with the appropriate stem .0 variable for the processed subcommand. Refer to 10.8.4, “Rexx variables” on page 426 for more information about the stem.0 variables.

10.8.5 Dynamic installation exits

You should consider these exits before the program is initially loaded.

The migration action required for this support is to educate operators and systems programmers about the differences between how DFSMSrmm used to work and how it works now. Items that may need updating include:

- Operator procedures; Ensure that any mention of F DFRMM,REFRESH EXITS is replaced with use of the MVS operator commands that can be used with dynamic exits
- System Automation; Ensure that any automated processing you have for the deleted messages EDG0301I, EDG0302I, EDG0303D, EDG0304I, EDG0305I, and EDG0306I is updated to instead process the EDG0311I-EDG0314I messages.

If you are using DFSMSrmm installation exits, you do not need to change the exits; just install them as you always have done. DFSMSrmm automatically exploits the dynamic exit services using your installation exits as the default exit modules at the first start up of the DFRMM procedure.

You do not need to change the load module names used as exit routines with DFSMSrmm exits, but you may do so if you want.

Refer to 10.2, “Dynamic Exits Services” on page 343 for more information.

Controlling the exit routine through the z/OS dynamic exits facility

IBM has defined the EDG_EXITn00 exit to the dynamic exits facility. You can refer to the exit by the name EDG_EXITn00. You can use the EXIT statement of the PROGxx parmlib member, the SET PROG=xx operator command, the SETPROG EXIT operator command, or the CSVDYNEX macro to control this exit and its exit routines. Figure 10-118 shows a sample of the SETPROG to add the EDGUX300 user exit after the program is initially loaded.

```
SETPROG EXIT,ADD,EXITNAME=EDG_EXIT300,MODNAME=EDGUX300
```

Figure 10-118 SETPROG EXIT syntax

10.8.6 VRSEL(OLD) parmlib option

You should consider this option before the program is initially loaded.

Do not use the VRSEL(OLD) parmlib option. Prior z/OS V1.11 releases, EDGHSKP issued a warning message when VRSEL(OLD) was in use.

Tip: Refer to Chapter 8, “DFSMSrmm enhancements”, in *z/OS V1R8 DFSMS Technical Update*, SG24-7435 for more information.

10.9 DFSMSrmm VTFM setup

In the following section, we describe the steps you must perform to set up DFSMSrmm to support a VTFM Library. You need to perform two actions:

1. Define a VLPOOL.
2. Add the volumes to the DFSMSrmm.

10.9.1 Defining a VLPOOL

Define a new volume pool operand (VLPOOL) in your EDGRMMnn parmlib member that matches the volumes you have specified in the HVTUTAP utility to add virtual tapes to the VTFM volume data base (VDB), as shown in Figure 10-119.

```
VLPOOL PREFIX(VT*)                    /* volume prefix             */ -
       AUTOSCRATCH(YES)               /* return to scratch auto    */ -
       DESCRIPTION('VTFM 3592 POOL')  /* pool description          */ -
       EXPDTCHECK(N)                  /* do not check VOL1 expdt   */ -
       MASTEROVERWRITE(LAST)          /* control DSN overwriting   */ -
       MEDIANAME(3592)                /* media name used           */ -
       NAME($$DFLT)                   /* pool name                 */ -
       RACF(N)                        /* Type of RACF tape support */ -
       TYPE(S)                        /* satisfy scratch requests  */
```

Figure 10-119 DFSMSrmm VLPOOL definition for a VTFM volume pool

If you do not follow these rules, your scratch request cannot be resolved, because DFSMSrmm will reject the scratch request, as shown in Figure 10-120.

```
EDG4021I VOLUME VT0016 REJECTED. IT IS NOT IN AN ACCEPTABLE SCRATCH POOL.
         POOL= REQUESTED, RACK=VT* MOUNTED
EDG4005E VOLUME VT0016 ON 040F REJECTED FOR USE BY HVTJIVPA, STEP03, SYSUT2
```

Figure 10-120 EDG4021I volume reject message

Where:

```
EDG4021I VOLUME volser REJECTED. IT IS NOT IN AN ACCEPTABLE SCRATCH POOL,
rtype=rvalue REQUESTED mtype=mvalue MOUNTED
```

Explanation	DFSMSrmm found that the specified volume serial number is not assigned to an acceptable scratch pool. Only scratch tapes from acceptable pools can be used for scratch tape mounts, so this volume is rejected. DFSMSrmm indicates in the mount message what caused the volume to be rejected.
System action	If you are running DFSMSrmm in warning mode, the volume specified in this message can be used and DFSMSrmm issues the EDG4004I message. If you are running DFSMSrmm in protect mode, DFSMSrmm issues the EDG4005E or EDG4006E message.
Operator response	None.
System programmer response	If you are using the DFSMSrmm EDGUX100 installation exit to implement exit-based scratch pools, check that the implementation steps have been correctly followed. Otherwise, use the information in the message to determine why the volume is rejected. You need to consider how a scratch pool is selected: by SMS ACS processing, EDGUX100 installation exit, or by DFSMSrmm system-based pooling. You can use the RMM LISTCONTROL VLPOOLS subcommand to list the pool definitions. If the volume was incorrectly rejected because of incorrect installation options, update the DFSMSrmm start up options and restart DFSMSrmm.

If you have specified a TYPE of “R”, as described in the *IBM VTF Mainframe Customization & Reference*, GC53-1190, you must customize and use the EDGUX100 user exit to make a valid pool selection. If you do not use the EDGUX100 exit, you have to specify a TYPE of “S” to satisfy scratch requests.

Important: You must specify a TYPE of “S” or you have to use the EDGUX100 user exit to make a volume pool selection.

Defining a location type HOME

To identify the volumes managed by a IBM System Storage VTF Mainframe (VTFM), define a storage location type HOME for them. Figure 10-127 on page 432 shows an example of this task for your EDGRMMxx PARMLIB member.

```
LOCDEF LOCATION(VTFM001)                                          -
       MANAGEMENTTYPE(NOBINS)                                     -
       MEDIANAME(*)                                               -
       TYPE(STORAGE,HOME)
```

Figure 10-121 EDGRMMxx LOCDEF command example

Defining MEDIAINF

To identify the volumes managed by a IBM System Storage VTF Mainframe (VTFM), define a storage location type HOME for them. Figure 10-127 on page 432 shows an example of this task for your EDGRMMxx PARMLIB member.

```
MEDINF NAME(VTFM)                    /* define the VTFM volumes   */ -
       MEDIATYPE(5,ETC)                                             -
       CAPACITY(1000)
MEDINF NAME(VTFM)                    /* define the VTFM volumes   */ -
       MEDIATYPE(5,ETC)                                             -
       RECORDINGFORMAT(5,EFMT1)                                     -
       CAPACITY(1000)
```

Figure 10-122 EDGRMMxx LOCDEF command example

After we have re-started DFSMSrmm, we check the current MEDINF settings by way of the RMM TSO LISTCONTROL MED subcommand, as shown in Figure 10-123.

```
PROF MSGID
READY
 RMM LISTCONTROL MED

Media Information :
                                                          Replace Policy
Name     Media type   Recording format Capacity (MB) Perm  Temp  WMC   Age
-------- ------------ ---------------- ------------- ----- ----- ----- -----
VTFM       5 ETC          5 EFMT1             1000       0     0     0     0
VTFM       5 ETC          0 *                 1000       0     0     0     0
VTFM       0 *            0 *                    0       0     0     0     0

READY
```

Figure 10-123 RMM LISTCONTROL MEDINF subcommand

Using EDGUX300 User Exit

You must use the EDG_EXIT300 installation exit to dynamically set media information for volumes that have a media information name (MEDINF operand) other than IBM. If you do not use the EDG_EXIT300 installation exit, only the volumes with an IBM MEDINF name are maintained dynamically.

Use the EDG_EXIT300 installation exit to supply media information for a specific volume for non-IBM media using the vendor API. The exit is called during O/C/EOV processing if the installation-defined media information for this volume is not IBM, and you have MEDINF definitions.

Refer to 10.2, “Dynamic Exits Services” on page 343 for more information about implementing this exit.

Changing media information

If you have already added your volume to DFSMSrmm CDS, you can change the media information, the recording format, and the MEFINF matching to the information you have specified in the MEDINF parmlib option by way of the RMM TSO CHANGEVOLUME subcommand, as shown in Figure 10-124.

```
RMM CHANGEVOLUME VT0001 TYPE(LOGICAL) MEDINF(VTFM)      -
    MEDIATYPE(VT3590G2) RECORDINGFORMAT(VT3590G2) FORCE
```

Figure 10-124 RMM TSO CHANGVOLUME subcommand to correct the media information

Figure 10-125 shows the volume information before correction.

```
                    DFSMSrmm Volume Details - VT0100
Command ===>

Volume . . . . . : VT0100     VOL1 volser :         Rack number :
Media name . . . . : 3480                           Status  . . : SCRATCH
                                                                  More:   - +
Volume use count . : 0              Volume usage (KB)  . . . . : 0
Capacity (MB)  . . : 1000           Percent full . . . . . . . : 0
Create date  . . . : 2009/233       Create time  . . . . . . . : 13:20:50
                                    System ID  . . . . . . . . : SC70
Date last read . . :                Date last written  . . . . :
Drive last used  . :                Write Mount count  . . . . : 0
```

Figure 10-125 DFSMSrmm volume capacity after the volume is added

After the volume is used the first time, the real volume capacity is updated by information coming back from the tape drive, as shown in Figure 10-126.

```
                    DFSMSrmm Volume Details - VT0100
Command ===>

Volume . . . . . : VT0100     VOL1 volser :         Rack number :
Media name . . . . : 3480                           Status  . . : MASTER
                                                                  More:   - +
Volume use count . : 1              Volume usage (KB)  . . . . : 28
Capacity (MB)  . . : 1000           Percent full . . . . . . . : 0
Create date  . . . : 2009/233       Create time  . . . . . . . : 13:20:50
                                    System ID  . . . . . . . . : SC70
Date last read . . : 2009/235       Date last written  . . . . : 2009/235
Drive last used  . : 0402           Write Mount count  . . . . : 1
```

Figure 10-126 DFSMSrmm volume capacity after first use

Adding volumes

To identifying the volumes managed by a IBM System Storage VTF Mainframe (VTFM), define a storage location type HOME for them. Figure 10-127 shows an example of how to perform this task in your EDGRMMxx PARMLIB member.

```
RMM ADDVOLUME VT0000 STATUS(SCRATCH) -
              COUNT(100) -
              INITIALIZE(N) -
              LOCATION(VTFM001) -
              MEDINF(VTFM) -
              MEDIATYPE(VT3590G2) -
              RECORDINGFORMAT(VT3590G2) -
              TYPE(LOGICAL)
```

Figure 10-127 EDGRMMxx LOCDEF command example

Where:

INITIALIZE(N) Indicates that the volume(s) does not need to be initialized.

LOCATION(VTFM001) Specifies the location where the volume resides other than in the removable media library or a storage location.

MEDINF(VTFM) Specifies the assigned installation-defined media information to the volume. This value is one to eight alphanumeric characters and must be defined once in your installation. You can use the LISTCONTROL subcommand with the MEDINF operand to display the media information. For more information about MEDINF, see *z/OS DFSMSrmm Implementation and Customization Guide*, SC26-7405.

MEDIATYPE(VT3590G2) Specifies the volume's physical media type. Specify a non-IBM media type if your installation definition contains media information for *medinf_mediatype* that matches the media information assigned to the volume.

RECORDINGFORMAT(VT3590G2) Specifies the basic recording format for tape volumes. Specify a non-IBM media recording format if your installation definition contains media information for *medinf_recordingformat* that matches the media information assigned to the volume.

TYPE(LOGICAL) DFSMSrmm does not assign a rack number to the logical volume because the logical volume's association is with a stacked volume container or library rather than a rack number or pool.

CBRUXVNL user exit considerations

If you are using the sample volume that is not in the CBRUXVNL library installation exit, you should change the code so that the VTFM,HOME location is supported correctly and so that you can specify your logical VTFM volumes with TYPE=LOGICAL in DFSMSrmm. Figure 10-128 shows you the update we have made to the sample code.

```
ACTTST1  CLI   LCSUP_LOCTYPE-LCSUP_HDR+LCSUPWA,LCSUP_TYPE_MTL  Current X09557600
               location MTL?                                            @11C 09578800
         BE    SETVNL2                       Yes -> want 2nd call       @04A 09600000
         CLI   LCSUP_LOCTYPE-LCSUP_HDR+LCSUPWA,LCSUP_TYPE_ATL  Current X09633300
               location ATL?                                            @11C 09666600
         BE    SETVNL2                       Yes -> want 2nd call       @04A 09700000
        CLI LCSUP_VOLUMETYPE-LCSUP_HDR+LCSUPWA,LCSUP_VOLUMETYPE_LOGICALX09720800
                                                                        @11C 09729100
         BNE   ACTTST2                       No -> test next            @mwA
         CLI LCSUP_HOMETYPE-LCSUP_HDR+LCSUPWA,X'02' ATL/MTL?            @mwA
         BO    SETVNL2                       Yes -> want 2nd call       @mwC 09737500
ACTTST2  CLI   LCSUP_DESTYPE-LCSUP_HDR+LCSUPWA,LCSUP_TYPE_MTL            X09783300
               Destination is   MTL?                                    @11C 09816600
         BE    SETVNL2                       Yes -> want 2nd call       @04A 09850000
         CLI   LCSUP_DESTYPE-LCSUP_HDR+LCSUPWA,LCSUP_TYPE_ATL            X09883300
               Destination is   ATL?                                    @11C 09916600
         BE    SETVNL2                       Yes -> want 2nd call       @04A 09950000
ACTTST3  CLI   LCSUP_HOMETYPE-LCSUP_HDR+LCSUPWA,LCSUP_TYPE_MTL  Home   X10083300
```

Figure 10-128 Update the CBRUXVNL sample exit to support VTFM logical volumes

After the update, you have to compile and link the exit to test the functionality, as shown in Figure 10-129.

```
//ASMCL    PROC SM=NONE,LM=NONE
//C        EXEC PGM=ASMA90,
//         PARM='NODECK,XREF(SHORT),LINECOUNT(58)'
//SYSLIN   DD   DSN=RMM.ADDONS.OBJ(&LM),DISP=SHR
//SYSLIB   DD   DSN=SYS1.MACLIB,DISP=SHR
//         DD   DSN=SYS1.MODGEN,DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   UNIT=SYSDA,SPACE=(7040,400)
//SYSUT2   DD   UNIT=SYSDA,SPACE=(3520,400)
//SYSUT3   DD   UNIT=SYSDA,SPACE=(3520,400)
//SYSIN    DD   DSN=SYS1.SAMPLIB(&SM),DISP=SHR
//L        EXEC PGM=HEWL,
//         PARM='LIST,MAP,XREF,REFR',
//         COND=(4,LT,C)
//SYSLMOD  DD   DSN=SYS1.LINKLIB(&LM),DISP=SHR
//SYSPRINT DD   SYSOUT=*
//SYSUT1   DD   UNIT=SYSDA,SPACE=(3520,400)
//SYSLIN   DD   DSN=RMM.ADDONS.OBJ(&LM),DISP=SHR
//         PEND
//COMPLNK0 EXEC ASMCL,LM=CBRUXVNL,SM=CBRUXVNL
```

Figure 10-129 Compile and link the CBRUXVNL user exit

Before you can test the functionally of the new volume that is not in the library exit, you must take an LLA refresh, as shown in Figure 10-130.

```
   /F LLA,REFRESH

   CSV210I LIBRARY LOOKASIDE REFRESHED
```

Figure 10-130 MVS command for a LLA refresh

Adding virtual volumes to the VTFM VDB

Use the HVTUTAP utility to add the previously defined volumes in the DFSMSrmm CDS as virtual tapes to the VTFM VDB. For example, the initialization statements shown in Figure 10-131 are supplied to the utility.

```
//UTAP     EXEC VTFMSRV,SERVICE=UTAP      <== PROCEDURE NAME
//HVTUTIL  DD *
 TAP     FUNCTION=ADD,
                      FIRST=VT0100,       <== FIRST VIRTUAL VOLUME
                      LAST=VT0199,        <== LAST VIRTUAL VOLUME
                      INCR=NUM
//
```

Figure 10-131 Add virtual volumes to the VTFM VDB

Figure 10-132 shows the report produced by the HVTUTAP utility when all the tapes have been added successfully.

```
15.33.35 HVT304I POL    REFRESH WAS OK, RC=00
15.33.35 HVT200I VTF Mainframe VER 3.0.0 UTILITY HVTUTAP  STARTED
15.33.35 HVT355I VOLUME=VT0100 FUNCTION=ADD      WAS OK, RC=00
15.33.35 HVT355I VOLUME=VT0101 FUNCTION=ADD      WAS OK, RC=00
...
15.33.35 HVT355I VOLUME=VT0198 FUNCTION=ADD      WAS OK, RC=00
15.33.35 HVT355I VOLUME=VT0199 FUNCTION=ADD      WAS OK, RC=00
15.33.35 HVT201I UTILITY HVTUTAP  ENDED, RC=00
```

Figure 10-132 HVTUTAP utility report

10.9.2 ICF user catalog considerations

All data sets are created inside a Virtual Tape Facility for Mainframe (VTFM) receives an x'44048083' catalog device type, as shown in Figure 10-133.

```
NONVSAM ------- VTFM.IVP001A
     IN-CAT --- UCAT.VTFM
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.224
       RELEASE----------------2     EXPIRATION------0000.000
     VOLUMES
       VOLSER------------VT0017     DEVTYPE------X'44048083'     FSEQN-----------------1
     ASSOCIATIONS--------(NULL)
     ATTRIBUTES
```

Figure 10-133 Catalog device type

Important: Use the correct catalog device type information to recatalog a data set residing on a VTFM virtual tape volume.

10.9.3 Scratching or reclaiming virtual tapes

Once a virtual tape is used by VTFM, it is marked as “active” in the VDB and it is not eligible for mount as scratch. These virtual tapes are scratched by the HVTUSCR utility (as directed by the tape management system). As the utility scratches the virtual tapes, it releases the disk space used by them while they were active. The actual scratch of the virtual tapes may be deferred according to the DEFERBY parameter’s value of the VTFM TPOOL initialization statement.

You can specify how long a volume should be retained by the VTFM before the status is really changed back to scratch, because all data sets on it are erased at this time. In our example, we specified that a volume will be “pending scratch” for 23 hours, as shown in Figure 10-134.

```
DPOOL NAME=DPOOL1, INTERNAL NAME FOR DISK POOL
UNIT=(,VTFM), STORAGE FOR VIRTUAL DS
THRESH=(85,1800,50) %FULL,SEC'S TO WARN,%LOWTHRESH
* UNIT=(#unit,#storclas,#mgmtclas)
*
TPOOL NAME=$$DFLT, DEFAULT TPOOL
DPOOL=DPOOL1,
RANGE1=(0,0), INDICATES IT IS DEFAULT POOL
TAPECAP=(1000,16),
THRESH=(90,1800), %FULL,SEC'S TO WARN
COMPRESS=0,
DEFERBY=23,
CKPINTVL=0
```

Figure 10-134 VTFM scratch pool definition

The DEFERBY parameter specifies if and by how much time virtual tapes scratching should be delayed by HVTUSCR utility.

hours Specifies the number of hours to pass from the time HVTUSCR first encountered a scratch-eligible virtual tape in this tapepool until the virtual tape can be actually scratched.

The virtual tape scratch is not done automatically when the deferred time expires, but on the next run of the HVTUSCR utility.

The following special values exist:

0 No defer-scratch occurs and the HVTUSCR utility immediately scratches any scratch eligible virtual tape from this tapepool.

9999 The HVTUSCR will never scratch virtual tapes from this tapepool. Use this value when:

- The virtual tapes from this tapepool are managed by an EDM and you did not define them to the local tape management system.
- This site is used as a remote off-site storage location for type4 e-Vault virtual tapes and the virtual tapes are not defined to the local tape management system.

Value range: Integer from 0 to 1800 or 9999.

Default: 0.

DEFERBY This parameter specifies if and by how much time virtual tapes scratching should be delayed by the HVTUSCR utility.

hours Specifies the number of hours to pass from the time HVTUSCR first encountered a scratch-eligible virtual tape in this tapepool until the virtual tape can be actually scratched. The virtual tape scratch is not done automatically when the deferred time expires, but on the next run of the HVTUSCR utility. The following special values exist:

0 No defer-scratch occurs and the HVTUSCR utility immediately scratches any scratch eligible virtual tape from this tapepool.

9999 The HVTUSCR will never scratch virtual tapes from this tapepool. Use this value when:

- The virtual tapes from this tapepool are managed by an EDM and you did not define them to the local tape management system.
- This site is used as a remote off-site storage location for type4 e-Vault virtual tapes and the virtual tapes are not defined to the local tape management system.

Value range: Integer from 0 to 1800 or 9999.

Default: 0.

Figure 10-135 shows a VTFM inventory report. Some of the volumes are retained by the DEFERBY setting. See the (0023H) information in column SCR.

```
118.40.47 HVT304I POL    REFRESH WAS OK, RC=00                                                          *HVTRFR  *
 18.40.47 HVT200I VTF Mainframe VER 3.0.0 UTILITY HVTUREP  STARTED                                    *HVTUREP *

 1VOLUME SCR       LOCID___(VD) LB U R SID  ALC_M* JOBNAME   JOBID     DATE     TIME   VIRTUAL TAPE DATASET NAME
  ------ --------- ------------ -- - - ---- ------ -------- -------- -------- -------- ------------------------------------
 0IVP001 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10503 2009.222 13:42:20 VTFMTAPE.VIVP001.HAIM0.TEST01
  VT0000 YES       *MAIN*       SL
  VT0001 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10505 2009.222 13:46:14 VTFMTAPE.VVT0001.HAIM0.TEST02
  VT0002 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10506 2009.222 13:54:37 VTFMTAPE.VVT0002.HAIM0.TEST02
  VT0003 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10507 2009.222 13:54:58 VTFMTAPE.VVT0003.HAIM0.TEST03
  VT0004 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10509 2009.222 14:01:07 VTFMTAPE.VVT0004.VTFM.IVP001A
  VT0005 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10510 2009.222 14:06:56 VTFMTAPE.VVT0005.HAIM0.TEST04
  VT0006 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10511 2009.222 14:08:05 VTFMTAPE.VVT0006.VTFM.IVP001A
  VT0007 NO(0023H) *MAIN*       SL     SC70 00062  IEBGENE1 JOB10513 2009.222 14:26:06 VTFMTAPE.VVT0007.HAIM0.TEST05
  VT0008 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10519 2009.222 14:51:29 VTFMTAPE.VVT0008.VTFM.IVP001A
  VT0009 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10519 2009.222 14:51:29 VTFMTAPE.VVT0009.VTFM.IVP002
  VT0010 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10520 2009.222 14:59:58 VTFMTAPE.VVT0010.VTFM.IVP001A
  VT0011 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10520 2009.222 14:59:58 VTFMTAPE.VVT0011.VTFM.IVP002
  VT0012 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10521 2009.222 15:01:48 VTFMTAPE.VVT0012.VTFM.IVP001A
  VT0013 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10521 2009.222 15:01:48 VTFMTAPE.VVT0013.VTFM.IVP002
  VT0014 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10522 2009.222 15:03:04 VTFMTAPE.VVT0014.HAIM0.IVP001A
  VT0015 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10522 2009.222 15:03:04 VTFMTAPE.VVT0015.HAIM0.IVP002
  VT0016 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10577 2009.222 20:40:28 VTFMTAPE.VVT0016.VTFM.IVP001A
  VT0017 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10732 2009.224 00:53:17 VTFMTAPE.VVT0017.VTFM.IVP001A
  VT0018 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10732 2009.224 00:53:17 VTFMTAPE.VVT0018.VTFM.IVP002
  VT0019 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10734 2009.224 01:27:30 VTFMTAPE.VVT0019.RMM.VTFM.IVP001A
  VT0020 NO(0023H) *MAIN*       SL     SC70 00062  HVTJIVPA JOB10734 2009.224 01:27:31 VTFMTAPE.VVT0020.RMM.VTFM.IVP002
  VT0021 YES       *MAIN*       SL
  VT0022 YES       *MAIN*       SL
....
```

Figure 10-135 VTFM volume inventory report

Where:

NO(0023H) The delay time before the tape is really scratched. This time is set by the DEFERBY parameter in your TPOOL settings.

Note: You can also bring back a single volume in status scratch by using the HVTUTIL DD statement in the VTFM procedure.

Using the VTFMSRV procedure to change a volume status

Instead of updating the whole VTFM data base, you can update a single volume to change the volume status from private to scratch or reclaim a volume from its deferred-scratch status back to private. Use the VTFMSRV VTFM procedure and add a SYNC control statement for each volume you want to process. Figure 10-136 shows an example.

```
//USCR     EXEC VTFMSRV,SERVICE=USCR     <== PROCEDURE NAME
//HVTUTIL  DD *
 SYNC      TYPE=VOLSER,VOLSER=VT0000,FORCE=YES
//
```

Figure 10-136 Use of SYNC for a single volume

The syntax of the SYNC operand is shown in Figure 10-137.

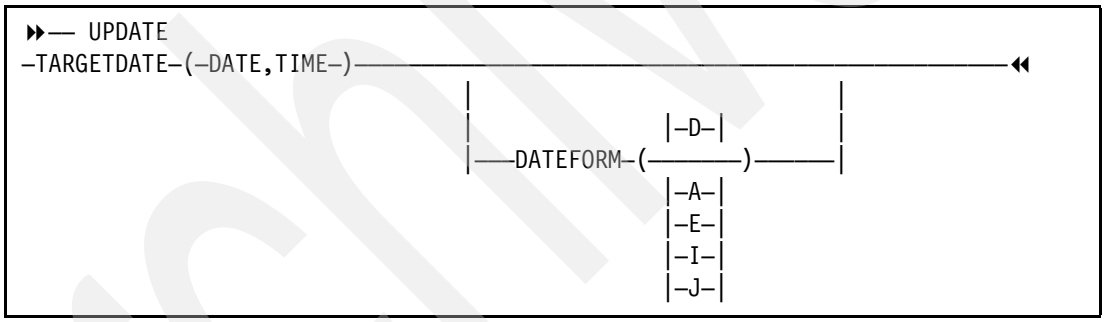


Figure 10-137 SYNC command syntax

Where:

SYNC The SYNC initialization statement requests that the scratch and e-Vault status of all the entries in the VDB will be synchronized with the tape management system database or MVS catalog, or that a specific virtual tape entry will be synchronized as per the user request.

TYPE The TYPE parameter defines the method by which the VDB synchronization will be performed.

VOLSER When using this option, HVTUSCR handles separately each virtual tape specified in the VOLSER parameter. The virtual tape is handled regardless of its status in the tape management system database or MVS catalog. Incorrect use of this option may cause discrepancies between the virtual tape status in the VDB and its status in the tape management system database (if used).

If LOCID=locid is not specified, refer to HVTUSCR process when TYPE=volser is coded and LOCID=locid is not specified. If LOCID=locid is specified, the virtual tape is not scratched. Refer to the LOCID=locid parameter in the *IBM VTF Mainframe Customization & Reference*, SG24-7768 for a description of the HVTUSCR process for that option.

If more than one specific virtual tape will be handled, the SYNC initialization statement should be repeated. A sample job to run HVTUSCR with TYPE=VOLSER can be found in the HVTUSCRV member in the VTFM SAMPLIB library.

CATALOG

This option is used by sites that do not use any tape management system and the data set residing on virtual tapes are cataloged (either in the master or user catalog(s)). A virtual tape, where all the data sets residing on it are not cataloged, is considered to be scratch.

During the MVS catalogs scan, the utility lists all relevant entries from the master and all connected user catalogs. User catalogs not connected to the master catalog are not scanned and the tape data sets cataloged in it are not listed. That means that virtual tapes that contain data sets cataloged in user catalogs accessed via JOBCAT or STEPCAT DD statements will be scratched. Accordingly, HVTUSCR with TYPE=CATALOG cannot be used when jobs catalog their data sets in user catalogs not connected to the master catalog.

If one or more user catalogs are not accessible, the utility terminates and no virtual tape is scratched. You may have connected user catalogs that are not accessible and do not really contain relevant entries. To allow the HVTUSCR process, you can exclude these irrelevant user catalogs from the utility catalogs scan by specifying their names in the XUCAT001- XUCAT999 DD statements. If the disks where these user catalogs reside are not accessible, you need to specify any existing disk volume serial number and unit along with the user catalog names on the XUCAT001-XUCAT999 DD statements. A sample job to run HVTUSCR with TYPE=CATALOG can be found in the HVTUSCRC member in the VTFM SAMPLIB library.

RMM

When using this option, the HVTUSCR utility scans an IBM DFSMSrmm EDGHSKP report produced in an earlier step of the same job. The report contains the current TCDB tapes from which the virtual tapes that are not scratch (active), along with their vault location, are filtered for processing.

All virtual tapes not listed in this report are considered to be scratch.

A sample job to run HVTUSCR with TYPE=RMM can be found in the HVTUSCRR member in the VTFM SAMPLIB library.

It is mandatory for the EDGHSKP report to be produced in the same job that runs the HVTUSCR utility, otherwise the utility may scratch active virtual tapes created after the job run starts. The report generator used by previous VTFM releases can be used with the current VTFM release.

Value range: N/A.

Default: None.

VOLSER This parameter is used only when TYPE=VOLSER is used and specifies the virtual tape volume serial number to be handled by the utility.

Value range: A one to six alphanumeric character string.

Default: None.

RECLAIM This parameter is used only when TYPE=VOLSER and indicates whether the specific virtual tape should be reclaimed from its deferred-scratch status. For other TYPE values, the reclamation is handled automatically and this parameter is ignored.

YES The virtual tape should be reclaimed from its deferred-scratch status. No action is taken if the virtual tape has already been scratched.

NO The virtual tape should not reclaimed from its deferred-scratch status.

Value range: N/A

Default: NO.

FORCE This parameter indicates whether virtual tapes eligible for scratch will be scratched regardless of their deferred-scratch conditions.

YES Virtual tape eligible for scratch will be scratched regardless of their deferred-scratch status.

NO Virtual tape eligible for scratch will be scratched while considering their deferred-scratch status.

Value range: N/A.

Default: NO.

LOCID This parameter is used only when TYPE=VOLSER. When TYPE is not VOLSER, the e-Vault location name is extracted from the tape management system database and this parameter is ignored.

When the LOCID parameter is specified, the virtual tape is not scratched. Rather, it is marked “to be vaulted” or “to be returned” (according to the locid value).

locid Specifies either the local vault location name or a remote off-site storage location name to which this virtual tape is to be moved.

- If locid is similar to the OWNLOCID parameter specified on the VLTOWN initialization statement and the virtual tape currently resides in a remote off-site storage location, the virtual tape is marked “to be returned”.
- If locid is similar to one of the LOCID parameters in the VLTRMT initialization statements and the virtual tape currently resides in the local location, the virtual tape is marked “to be vaulted” to that location.

Value range: A one to eight alphanumeric character string or blanks surrounded by quotes.

Default: None.

Using the SEARCHVOLUME subcommand

You can use the RMM TSO SEARCHVOLUME subcommand to create SYNC control statements that can be used to synchronize the VTFM Volume Data Base (VDB). Figure 10-138 shows a sample JCL using the SEARCHVOLUME subcommand to create control statements.

```
//CLEANUP  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
 DELETE RMM.VTFM.SYNC.D2009227 NONVSAM PURGE
 SET MAXCC=0
/*
//SEARCHV  EXEC PGM=IKJEFT01
//SYSTSPRT DD   SYSOUT=*
//RMMCLIST DD   DISP=(,CATLG),DSN=RMM.VTFM.SYNC.D2009227,
//         UNIT=3390,SPACE=(CYL,(1,1),RLSE),
//         DCB=(LRECL=80,BLKSIZE=0,RECFM=FB,DSORG=PS)
//SYSTSIN  DD   *
 RMM SV VOLUME(VT*) OWNER(*) LIMIT(*) STATUS(SCRATCH) -
     CLIST('SYNC      TYPE=VOLSER,VOLSER=',',FORCE=YES')
/*
```

Figure 10-138 Use SCEARCHVOLUME to create VTFM commands

Now you can use the previously created control cards to update your VTFM Volume Data Base (VDB) using the JCL shown in Figure 10-139.

```
//USCR     EXEC VTFMSRV,SERVICE=USCR     <== PROCEDURE NAME
//HVTUTIL  DD  DISP=SHR,DSN=RMM.VTFM.SYNC.D2009227
```

Figure 10-139 Sample JCL to update VTFM VDB

EDGHSKP considerations

Use the normal daily EDGHSKP report extract file to update the VTFM Volume Data Base. Depending on the VTFM requirement, you have to copy the RPTEXT or XRPTEXT file to a temporary one. If you are using XRPTEXT, you must be sure that the required V (Volume) records are included. Figure 10-140 shows a sample JCL.

Note: The input file to update the VTFM volume data base (VDB) must be a temporary file.

```
//STEP0001 EXEC PGM=EDGHSKP,
//         PARM=('BACKUP,VRSEL,EXPROC,RPTEXT,DATEFORM(J)')
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   DISP=SHR,DSN=RMM.HSKP.MESSAGE
//BACKUP   DD   DISP=SHR,DSN=RMM.HSKP.CDSBKUP
//JRNLBKUP DD   DISP=SHR,DSN=RMM.HSKP.JNLBKUP
//XREPTEXT DD   DISP=SHR,DSN=RMM.HSKP.EXTRACT
//REPORT   DD   DISP=SHR,DSN=RMM.HSKP.REPORT
//ACTIVITY DD   DISP=SHR,DSN=RMM.HSKP.ACTIVITY
//SYSIN    DD *
 RPTEXT RECORDS(V,X)
//* RPTEXT RECORDS(B,D,X,O,P,R,S)
//STEP0002 EXEC PGM=IEBGENER
//SYSPRINT DD   DUMMY
//SYSUT1   DD   DSN=RMM.HSKP.MESSAGE,DISP=SHR
//SYSUT2   DD   SYSOUT=*
//SYSIN    DD   DUMMY
//*
//STEP0003 EXEC PGM=IEBGENER
//SYSPRINT DD   DUMMY
//SYSUT1   DD   DISP=SHR,DSN=RMM.HSKP.EXTRACT
//SYSUT2   DD   DISP=(,PASS),DSN=&REPORT,UNIT=SYSDA,
//         SPACE=(CYL,(5,5),RLSE),DCB=*.SYSUT1
//SYSIN    DD   DUMMY
//*
//STEP0004 EXEC VTFMSRV,SERVICE=UREP     <== PROCEDURE NAME
//HVTUTIL  DD *
 REPORT  REP=REP001,LINECT=60
/*
//STEP0005 EXEC VTFMSRV,SERVICE=USCR,    <== PROCEDURE NAME
//            COND=(0,NE,STEP0001)
//REPIN    DD DISP=SHR,DSN=&REPORT
//HVTUTIL  DD *
 SYNC      TYPE=RMM
/*
//STEP0006 EXEC VTFMSRV,SERVICE=UREP     <== PROCEDURE NAME
//HVTUTIL  DD *
 REPORT  REP=REP001,LINECT=60
/*
```

Figure 10-140 DFSMSrmm daily housekeeping example

Where:

STEP0001 Performs the normal daily EDGHSKP processing. It uses the following functions:

- Vital record specification processing.
- Expiration processing.
- Report extract processing. If the XRPTEXT file is selected, the volume records must be created too.
- Backing up the DFSMSrmm control data set and journal and clearing the journal.

STEP0002 Copies the message file into the joblog.

STEP0003 Copies the report extract file (XRPTEXT) file to a temporary one. This action occurs because the VTFM only allows the use of a temporary file to update the VTFM volume data base.

STEP0004 Creates a VTFM detailed volume list before updating the VTFM volume data base.

STEP0005 Updates the VTFM volume data base based on the DFSMSrmm report extract file.

STEP0006 Creates a VTFM detailed volume list after updating the VTFM volume data base.

Note: STEP0004 and STEP0006 can be removed if the update of the VTFM volume data base is successful.

Additional information

In DFSMSrmm, if you have a primary tape management system and you have set up all the parameters and options correctly, you should add some of the messages (WTORs) VTFM writes to your automated operator system.

For example, VTFM stops the processing for a JOB is DFSMSrmm rejects a volume, as shown in Figure 10-141.

```
EDG4027I VOLUME IVP001 REJECTED. IT IS NOT A SCRATCH VOLUME AND MOUNT
         REQUEST WAS NON-SPECIFIC
EDG4005E VOLUME IVP001 ON 040B REJECTED FOR USE BY RES7GENR, IKJEFT01,  SYSUT2
```

Figure 10-141 DFSMSrmm messages EDG4028I and EDG4005E

In this case, VTFM stopped processing for this JOB and writes a WTOR, as shown Figure 10-142.

```
*619 HVT319W VOLUME=IVP001 REJECTED BY TMS. REPLY R (RETRY), N (FAIL)
```

Figure 10-142 VTFM warning message HVT319W

11

Network File System enhancements

In this chapter, we provide information about the new features of Network File System (NFS) V1.10 and V1.11

We cover the following topics in this chapter:

- NFS overview
- NFS V1.10 enhancements
  - Overview
  - Line items
- NFS V1.11 enhancements,
  - Overview
- NFS V1.11 Server
  - Line items
- NFS V1.11 Client
  - Line items
- Migration/Coexistence
- NFSv4 hints and tips

11.1 Network File System overview

This section briefly describes NFS and describes how data is accessed between the NFS Server and NFS Client.

NFS provides a flexible option for exchanging data between similar and different z/OS and non-z/OS systems that support the NFS protocols. The current version of z/OS NFS supports the industry standard NFS specification for Versions 2, 3, 4, and the TCP/IP protocol versions IPv4 and IPv6.

NFSv2 and V3 are described in the Request For Comments (RFC) 1094, found at the following address:

http://www.faqs.org/rfcs/rfc1094.html

NFSv4 is described in RFC 3530, found at the following address:

http://www.faqs.org/rfcs/rfc3530.hmtl

IPv4 is described in RFC 791, found at the following address:

http://www.faqs.org/rfcs/rfc791.html

IPv6 is described in RFC 2460, found at the following address:

http://www.faqs.org/rfcs/rfc2460.html

Note: The *de facto* standard port to communicate via NFS is port 2049.

NFS basically takes the files or data sets on one machine and makes it possible to access them from another machine. It does so transparently so that the user does not know whether the files are on the machine they are working with or on another machine.

11.1.1 Client/server relationship

A client is a computer or process that requests services in the network. A server is a computer or process that responds to a request for service from a client. A user accesses a service, which obtains data or other resources.

Figure 11-1 shows the z/OS NFS server implementation. z/OS NFS consists of two parts: the NFS client and the NFS server. z/OS data sets and z/OS UNIX files are accessed from remote clients through the z/OS NFS server. Other systems, shown as terminal screens, also support NFS as client or server. The systems are linked through the Transmission Control Protocol/Internet Protocol (TCP/IP) network used to communicate between the clients and servers.

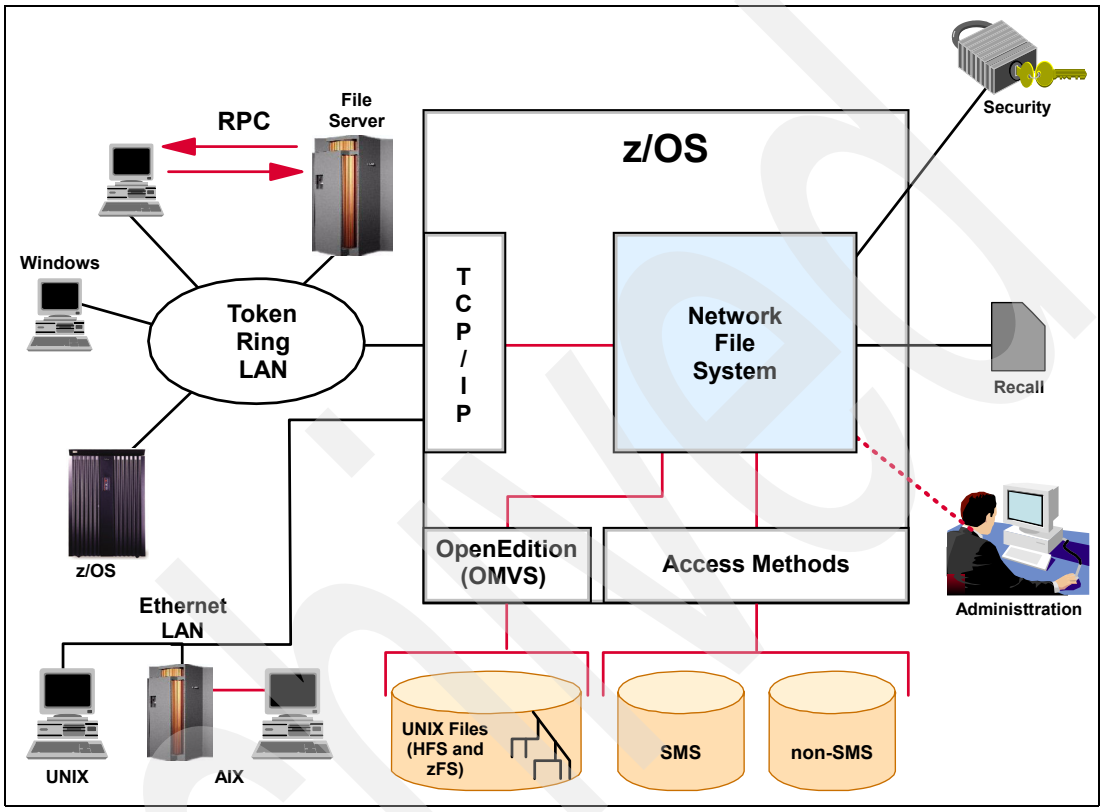


Figure 11-1 z/OS NFS implementation

The NFS uses the communication services provided by TCP/IP, a suite of protocols that includes the remote procedure call (RPC) and External Data Representation (XDR) protocols. RPC allows a program on one machine to start a procedure on another machine, as though the procedure is local. XDR resolves the differences in data representation of different machines. NFS, then, can be used for file sharing between platforms and file serving (as a data repository). If you use NFS as a file server, the z/OS UNIX file system might be a better choice than using conventional MVS data sets, because of its UNIX-based features.

Note: When we refer to HFS files, we mean z/OS UNIX files. These files are supported by NFS and accessed through z/OS Unix System Services (OMVS).

NFS supports UNIX files and MVS data sets. MVS data sets are also known as legacy data sets in this book.

11.2 NFS V1.10 enhancements

This section illustrates the NFS V1.10 enhancements for the z/OS NFS server and client.

Note: Refer to *Network File System Guide and Reference*, SC26-7417 for more information about NFS V1.10 enhancements.

11.2.1 NFS V1.10 overview

This section briefly describes NFS V1.10.

NFS is a distributed file system that provides transparent processing capability for data and information on worldwide and heterogeneous networks. The z/OS NFS provides the implementation that allows the z/OS platform to participate in these networks. A new industry wide version of the communication protocol, Version 4, has been defined and formalized as a standard.

The majority of the NFSv4 protocol was incorporated into the z/OS NFS server in z/OS NFS V1.7 and in the z/OS NFS client in V1.9.

The main focus of NFS V1.10 is intended to extend the z/OS NFS server and client support of the new protocol. In addition, z/OS NFS V1.10 also includes several other line items.

GFSA742W message during z/OS NFS server startup

Since V1.10, the z/OS NFS server shows the GFSA742W message during any startup for each TCP/IP stack defined in the TCP/IP profile that cannot be resolved. GFSA742W is related to the multidomain support of the V1.10 release of the z/OS NFS server, checking the TCP/IP stacks, which are defined in the BPXPRMXX parmlib member about their availability. GFSA742 is a warning message for TCP/IP stacks, also known as Physical File Systems (PFSs), which are defined but not yet started. An example of the startup is shown in Figure 11-2.

```
$HASP373 NFSS     STARTED
IEF403I NFSS - STARTED - TIME=13.26.05 - ASID=006D - SC64
GFSA949I (NFSS) Command DEBUG9 not valid.
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
GFSA742W (NFSS) Could not get hostname for 706
TCPIPOE,
Return Value(FFFFFFFF), Return Code(70), Reason Code(12C400B6).
GFSA742W (NFSS) Could not get hostname for 707
TCPIPMVS,
Return Value(FFFFFFFF), Return Code(70), Reason Code(12C400B6).
GFSA742W (NFSS) Could not get hostname for 708
TCPIPA,
Return Value(FFFFFFFF), Return Code(70), Reason Code(12C400B6).
GFSA742W (NFSS) Could not get hostname for 709
TCPIPB,
Return Value(FFFFFFFF), Return Code(70), Reason Code(12C400B6).
GFSA742W (NFSS) Could not get hostname for 710
TCPIPC,
Return Value(FFFFFFFF), Return Code(70), Reason Code(12C400B6).
GFSA348I (NFSS) z/OS Network File System Server (HDZ1B1N, OA29988)
started.
```

Figure 11-2 Sequence during the startup of z/OS NFS server in syslog

11.2.2 Extended Address Volume support

This section provides information about the Extended Address Volume (EAV) support and its implementation.

z/OS V1.10 DFSMS added support for Extended Address Volume (EAV) data sets (greater than 64 K cylinders). To maintain compatibility with the underlying MVS access methods, z/OS NFS V1.10 was modified to support access to these large volume data sets.

The NFS code is changed to recognize EAV data sets based on the new DSCB flag and to support the new format DSCB records. NFS will support reading, writing, and creating EAV data sets.

NFS support for accessing EAV data sets applies to all data set formats that NFS currently supports and to which the DFSMS EAV data set support applies. Currently, this is only VSAM data sets.

For additional information about EAV, refer to Chapter 4, “Extended Address Volume enhancements” on page 83.

11.2.3 Linux on IBM System z support

This section describes the Linux on System z support for the z/OS NFS server and z/OS NFS client.

z/OS NFS was tested on and is officially supported on the following remote platforms:

- SUN
- AIX®
- Intel®-based Linux
- Windows®

The Linux on System z platform is now supported with V1.10.

Nevertheless, the Client Enabling Utilities are required for remote NFS clients to successfully interact with the z/OS NFS server. Therefore, in order to claim that z/OS NFS supports Linux on System z, these utilities are validated on the Linux on System z platform. If necessary, some code changes might be required to allow these utilities to successfully compile (without errors or warnings), build, and execute successfully on all supported platforms.

z/OS NFS provides the following three Client Enabling Utilities that are now supported on Linux on System z:

- MVSLOGIN
- MVSLOGOUT
- SHOWATTR

Note: You can see the keywords that are added to the NFS client enabling utilities makefiles to add support for the Linux on System z platform in “Client Enabling Utilities makefile changes” on page 449.

11.2.4 Converting NFS Client Enabling Utilities to 64-bit support

This section illustrates converting the Client Enabling Utilities to 64-bit support.

z/OS NFS code only supported 32-bit addressing mode. Many platforms, including the current level of z/OS, provide 64-bit addressing mode as well as 32-bit addressing. Converting from 32-bit to 64-bit addressing mode is a large effort requiring many changes. Therefore, as a precursor to that conversion, this new feature focuses on first converting the Client Enabling Utilities to become 64-bit compatible.

Note: Implementation of this line item assumes the availability of the system runtime libraries on the respective platforms. If the necessary functions are not available, either under the same name, or under a different name for the 64-bit variation, then 64-bit support will not be provided for that platform.

Benefit of converting the Client Enabling Utilities

Converting the Client Enabling Utilities to 64-bit allows users to exploit the benefits of their 64-bit client platforms in conjunction with these utilities. The 64-bit compatibility will be provided for the following platforms:

- AIX
- SUN
- Linux on POWER5™
- Linux on System z

Client Enabling Utilities makefile changes

The following target platform keywords were added to the NFS client enabling utilities makefile to support the various platforms and add support for 64-bit addressing:

- ZLINUX: This keyword will be added to specify that the NFS Client Enabling Utilities are to be built for the Linux on System z platform using 32-bit addressing.
- ZLINUX64: This keyword will be added to specify that the NFS Client Enabling Utilities are to be built for the Linux on System z platform exploiting 64-bit addressing.
- LINUX64: This keyword will be added to specify that the NFS Client Enabling Utilities are to be built for the Linux platform exploiting 64-bit addressing.
- AIX64: This keyword will be added to specify that the NFS Client Enabling Utilities are to be built for the AIX platform exploiting 64-bit addressing.
- SUN64: This keyword will be added to specify that the NFS Client Enabling Utilities are to be built for the SUN platform exploiting 64-bit addressing.

11.2.5 NFS Server message globalization

This section provides you with details about message globalization within the z/OS NFS server.

The z/OS NFS server supported generation of console messages in English. The IBM corporate globalization strategy for the z/OS platform is to provide National Language Support (NLS) for console messages. This line item modifies the z/OS NFS server message support to become compliant with that strategy (only Japanese translation is provided).

Customers interested in the translated messages must run the Message Manufacturing Facility (MMF) against the message skeleton from the FMID for the language desired to build the VSAM MMF data set that contains all the translated messages. This step is typically bypassed by customers using messages only in English.

After the MMF data set is built and installed on a system along with the z/OS components that provide NLS, the system is ready to use the message data set for outputting the desired messages in the target language.

11.2.6 Continuation of NFSv4 server implementation

This section provides information about the new NFSv4 features within the z/OS NFS server.

NFSv4 locking

This feature completes the support for the NFSv4 Locking operations in the z/OS NFS server.

A key aspect of the NFSv4 Protocol is the incorporation of locking within the NFS protocol itself, versus having that setting defined in a separate Network Lock Manager (NLM) and Network Status Monitor (NSM) protocol. For details about the new protocol, refer to RFC3530 NFS Version 4 Protocol Specification, found at the following address:

`http://www.ietf.org/rfc/rfc3530.txt?number=3530`

When discussing file locking, two different types of locking must be addressed:

- Byte range locking is used by UNIX style machines.
- File level share reservations is used by DOS/Windows style machines.

The file level share reservations are used in conjunction with open/close operations. This support was implemented in z/OS NFS V1.7. This line item delivers the byte range locking support, which is provided via the NFSv4 locking operations in the z/OS NFS server.

Note: Refer to *Network File System Guide and Reference*, SC26-7417 for more information about the z/OS NFS server locking.

Byte range locking

Byte range locking is used to synchronize reads and writes to file data. This allows parallel applications to write concurrently to different parts of the same file, while maintaining Portable Operating System Interface (POSIX) read/write atomicity semantics. POSIX is the name of a family of related standards specified by the IEEE to define the application programming interface (API), along with shell and utilities interfaces for software compatible with variants of the UNIX operating system, although the standard can apply to any operating system. Applications that do not require POSIX semantics can utilize data shipping to bypass byte-range locking and avoid token state issues. Acquiring a token for a byte range for the duration of the read/write call and releasing it afterwards can increase the locking processing impact. If other users access other portions of the data, they can still acquire their own byte-range lock.

File level share reservations

File share reservations are an advisory form of access control among cooperating processes, on both local and remote machines. They are most often used by DOS or Windows emulators and DOS-based NFS clients. However, native UNIX versions or Windows applications might also choose to use this form of access control.

NFS locking background

NFSv2 and NFSv2 support provides for an advisory locking model, in that it is designed to work in parallel with the NLM protocol. The NLM protocol provides DOS/Windows style file level share reservations as well as shared and exclusive byte range locking.

NFSv4 incorporates the locking operations into the same protocol as other file access operations, specifically open, close, read, and write. The NFSv4 protocol also varies from NFSv2 and V3 in that it allows for files, file systems, or servers to be defined to use either advisory locking or mandatory locking rules.

Note: The z/OS NFS server only supports advisory locking.

Currently on z/OS, there are two major file system types that are supported by NFS:

- MVS data sets
- UNIX-style file system, which is accessed via z/OS UNIX System Services

The MVS data set file system has no concept of file share reservations or byte range locking, in the context that is described here. MVS applications accessing MVS data sets have never and most likely will never invoke file share or byte range locking interfaces. As such, they can be considered to be non-cooperative applications. Nevertheless, the z/OS NFS server uses services provided by UNIX System Services to implement byte range locks for MVS data sets. However, this lock for the MVS data set is just known by the z/OS NFS server and not by DFSMS. Therefore, a local (non-NFS) user will be able to access such a MVS data set. Byte range lock requests will be handled by the UNIX System Services component residing on each system. There will be no coordination of byte range locks for MVS data sets across systems in a sysplex when multiple NFS servers are defined on various systems in a sysplex.

Also, a NFS server does not know the lock set by another NFS server on the same z/OS system. For these reasons, if byte range locking is to be used by applications accessing MVS data sets, careful consideration should be used if the installation intends to concurrently utilize more than one NFS server within a sysplex.

The z/OS Unix System Services file system allows a UNIX application accessing a UNIX file to do byte range locking. As of z/OS V1.7, the UNIX file system supports both an advisory locking model and file share reservations.

Access control list attribute support

This feature adds support for the NFSv4 access control list (ACL) attributes.

Note: Refer to *Network File System Guide and Reference*, SC26-7417 for more information about the access control list support.

POSIX provides limited file security management granularity. File access security can only be controlled via the permission bits as specified for the user, group, and other classes. Some UNIX platforms have introduced additional security granularity by adding ACL support to provide security specification on an individual user or group basis. An ACL is simply a list that describes which users and groups get access to a file with what type of permission. The precise characteristics of this ACL support are platform specific. There is no industry standard protocol definition for this function.

The NFSv4 protocol has added the ability to remotely manage ACLs by providing the ability to display and modify ACL values via the ACL attribute. The NFSv4 protocol has provided a very rich ACL definition with granularity beyond that provided by many platform ACL implementations. Therefore, it is necessary to map between the NFS ACL definition and the platform definition. The key is to ensure that in this mapping process, the mapping should err in the direction of more restricted access, not less. When the NFS server sets an ACL, it must be set at least as secure as specified by the NFS request. When an NFS server sends an ACL to an NFS client, the client must not perceive the file as more secure than it really is. For details about the NFSv4 ACL definition, see the NFSv4 protocol.

For POSIX permission bits, and some flavors of ACLs, a single entry specifies whether permission is being granted or denied for the target user or group to access the file. By contrast, NFSv4 ACLs have two types of ACLs:

- Allow
- Deny

Allow indicates that the target user or group is being given the specified permission to access the file, while Deny indicates that the target user or group is explicitly being denied the specified permission to access the file.

Furthermore, the ordering of a POSIX ACL entry (ACE) is different from that of an NFSv4 ACE. POSIX has the following defined order:

- Owner
- Supplemental users
- Owning group
- Supplemental groups
- Other

This is a kernel-maintained order and cannot be changed by the user. NFSv4 ACEs do not have a rigid order; it is defined by the order of the entries in the ACL as created by the user.

z/OS UNIX System Services supports the concept of ACLs and provides a set of APIs for communicating ACL information to physical file systems (PFSs), which support this level of security. This line item implemented the capability to display and modify those ACLs via the NFSv4 protocol. The z/OS NFS server maps ACL requests between the z/OS UNIX ACL definition and the NFSv4 protocol definition. If a request cannot be mapped, then an error will be returned, as prescribed by the NFSv4 protocol.

MVS data set security has no notion of ACLs and has no mechanism for specifying, or saving, such values. Therefore, this line item will not implement ACL support for MVS data sets.

z/OS UNIX ACL background

The HFS, z/FS, and TFS z/OS UNIX System Services (OMVS) file systems support ACLs. This section provides a short overview of z/OS UNIX ACL characteristics. For more details, see the section “Using Access Control Lists (ACLs)” in Chapter 4, “Establishing UNIX security”, in *UNIX System Services Planning*, GA22-7800.

z/OS UNIX has three types of ACLs:

- Access ACL: Specifies the protection to be applied to a file system object.
- File Default ACL: Specifies the default access ACL inherited by files created in the parent directory. When sub-directories are created, they also inherit the file default ACL as their file default ACL.
- Directory Default ACL: Specifies the default access ACL inherited by a subdirectory created in the parent directory. It is also inherited as the directory default ACL.

Directories may have all three types of ACLs. All other file system objects only have Access ACLs.

z/OS UNIX ACLs may contain two kinds of ACL entries:

- Base ACL Entries are the same as the permission bits (user, group, and other). They are not physically part of the ACL but can be modified via ACL requests, or via the **chmod** command.
- Extended ACL Entries are ACL entries for individual users or groups. Like permission bits, they are stored with the file, not in RACF profiles. Each ACL type can contain up to 1024 extended ACL entries. Each entry contains the UID or GID to which it pertains, and the permissions being allowed or denied. The acceptable permission values are read, write, and execute.

The z/OS UNIX ACL entries, like POSIX permission bits, only support read, write, and execute permissions. However, the permissions can be specified on an individual user, or group, basis. For the user, or group, in an ACL entry, the permission bits are either turned on or off.

Remote ACL management restrictions

Due to the differences in ACL implementations on the various platforms, the following restrictions must be applied when attempting to remotely manage ACLs to and from z/OS through NFS:

Note: Only the z/OS, AIX. and Sun platforms are supported at this time.

- AIX restrictions
  - AIX supports two kinds of ACLs: AIX ACLs and NFS ACLs. Only the NFS ACLs can be remotely managed from z/OS.
  - The AIX client requires the acl option on the **mount** command to enable ACLs on the mount:

    ```
    mount -o  vers=4,acl maxi:/nfs-authsys/user11_acltest2
    /nfs-authsys/user11_acltest
    ```

    Note: AIX only supports NFS ACLs on JFS2 extended attributes NFSv2 file systems.

- z/OS UNIX does not have a unique Base Mask ACL entry:
  - The z/OS NFS server will use the base group ACL entry permission value for calculating the mask ACL entries sent to NFS clients.
  - The z/OS NFS server will fail set attribute requests containing Mask ACL entries that do not match the base group ACL entry and will return an error.
- z/OS UNIX does not have unique default base ACL entries:
  - The z/OS NFS server will use the access base ACL entries for calculating the default base ACL entries sent to NFS clients (for example, the access and default base ACL entries will always match).
  - The z/OS NFS server will fail set attribute requests containing default base ACL entries that do not match the access base ACL entries and will return an error.
  - Because z/OS UNIX only has default extended ACLs, default base ACLs will be returned by the z/OS NFS server if default extended ACLs are also present.
- Non-z/OS platforms do not have separate file and directory default ACLs:
  - The z/OS NFS server will use the file default ACL entries for the NFS default ACL entries sent to non-z/OS NFS clients. The entries will be used as both file and directory defaults on the non-z/OS platforms.
  - The z/OS NFS server will check all existing z/OS UNIX directory default ACL entries to verify that they match the z/OS UNIX file default ACL entries. If the two lists do not precisely match (that is, the same entries with the same permissions), the get request will fail with an error.
  - The z/OS NFS server will use the NFS default ACL entries sent by a non-z/OS client to set both the file and directory default ACL entries.
  - The z/OS NFS server will support the file and directory default ACL entries as unique entries when communicating with a z/OS NFS client. There are no ACL entry restrictions in this case.

NFS server site attribute

The new processing site attribute znfsclient indicates that client for this mount point is z/OS. This attribute is automatically set by the z/OS NFS client on NFSv2/V3 mount and on final lookup for NFSv4 mount emulation to tell the ACL management whether to transfer separate file and directory default NFS ACLs or check that z/OS UNIX file and directory default ACLs are identical for **getattr** requests.

This attribute is used for compatibility purposes only. For example, z/OS UNIX System Services has separate file and directory default ACLs. Other platforms have a common default ACL which is used for both. This flag allows the z/OS NFS server to customize its response to the NFS client accordingly.

Consideration: This value should never be specified directly. This attribute is automatically appended to the processing attribute string by the z/OS NFS client when it detects that it is sending the mount request (or the last lookup for an NFSv4 mount) to a z/OS NFS server. If this attribute is specified on a non-z/OS client mount request, some requests might not function properly.

Note: OA22759 is the toleration APAR for this function to allow the down-level release z/OS NFS server to accept and ignore the new znfsclient processing attribute.

NFSv4 name mapping support

This feature adds support for mapping between NFSv4 owner/group names and UNIX UID/GID values.

POSIX uses integer values for identifying users (for example, UID) and groups (for example, GID). This works very well when dealing with a closed environment where all the UID values and GID values are globally managed. However, in a large open networked environment, it can become a major maintenance problem to manage them globally, assigning unique values to all users, and ensuring that the same value is used on all systems for a given user or group. Therefore, the NFSv4 protocol moved away from using the UID and GID values to using `owner@dns_domain` and `group@dns_domain` strings for identifying owners and groups. This allows the owner and group names to be managed on a DNS domain basis, not globally. Even if the same owner name is defined in multiple domains but for different owners, this approach still provides name uniqueness.

Use of this domain-based name string construct frees the local systems to use their own local owner and group identification. This does mean that any given server might not understand all possible `owner@dsn_domain` values. A NFSERR_BADOWNER error code is defined in the NFSv4 protocol for indicating that a specified value is not understood by the receiving system. The only requirement on servers is that they support at least one dns_domain.

In prior releases, the z/OS NFS server did not support user/group names in the `owner@dns_domain` string. Instead, it only accepted UID/GID values converted to numerical strings. Therefore, this line item was created to resolve this restriction.

This line item provides z/OS NFS server support for mapping between owner and group names and the z/OS UNIX UID/GID values for a single dns_domain. These values are used for file ownership attribute specification and for remote security ACL management. The default dns_domain value is the local system domain. However, if DNS is not active, then the domain must be specified by the new site attribute NFSv4DOMAIN(domain). The value can also be overridden with a new NFS server site attribute value. When such a value is specified, it will be used for all owner/group name translations.

Also, an NFSv4DOMAIN(domain) attribute is added to the z/OS NFS client to allow the default local domain to be overridden in a similar manner, if necessary. If DNS is not active, then the domain must be specified via this site attribute.

Consideration: This line item does not support owner and group name resolution via a global name server, such as LDAP. It will only resolve names defined in RACF on the local system.

Consideration: In order to map between owner and group and UID and GID values, the owner and group name must be defined to RACF with the appropriate UID or GID value. This also restricts the support to only names that are supported by RACF. Thus, the names cannot be longer than eight characters and the names are case insensitive. Any mixed case names will be converted to upper case.

11.2.7 Continuation of NFSv4 client implementation

This section provides information about the new NFSv4 features within the z/OS NFS client.

Note: Refer to *Network File System Guide and Reference*, SC26-7417 for more information about the new z/OS NFS client features.

The majority of the NFS Client V4 protocol support was implemented in z/OS NFS V1.9. The following is a summary of additional items that were implemented in z/OS NFS V1.10:

- NFSv4 security
- NFSv4 locking
- Access control list (ACL) attributes

RPCSEC_GSS security

Note: RFC3010 NFSv4 protocol defines various types of authentication, which it calls “flavors”.

For more information about RPCSEC_GSS security, see the RFC 2203 RPCSEC_GSS protocol specification, found at the following address:

`http://www.faqs.org/rfcs/rfc2203.html`

The RPC level authentication that the z/OS NFSv2 and NFSv2 client supported was limited to the AUTH_SYS flavor. NFS data was therefore not protected across network transmissions.

The NFSv4 protocol provides stronger authentication and network transmission protection for NFS data by providing integrity with cryptographically strong checksums and also by providing for data privacy using encryption algorithms. Also, NFSv4 allows for multiple security mechanisms per file and provides for a means to negotiate security as the file system is being explored by the NFS clients. To facilitate this stronger security, NFSv4 mandates the use of the RPCSEC_GSS (Generic Security Service) security authentication flavor for RPC implementations and introduces the SECINFO operation for automated security negotiation. The z/OS NFSv4 client will support the security framework outlined in RFC3530 NFSv4 protocol, with the exceptions noted in this book.

In addition to supporting the new security flavor of RPCSEC_GSS, the z/OS NFS client will continue to support the existing RPC security authentication flavor of AUTH_SYS for NFSv4 that is currently supported for NFSv2 and NFSv2.

Limitations of RPCSEC_GSS security authentication flavor support

The z/OS NFSv4 client will support Version 1 of the RPCSEC_GSS security authentication flavor defined in RFC 2203 as required by RFC3530. However, the z/OS NFS client's support of this flavor will be limited to only supporting RFC1964 Kerberos V5 Security Mechanism with all three pseudo-flavors:

- krb5: This flavor provides Kerberos V5 based integrity for the RPC credentials (but not data) using the DES_MAC_MD5 integrity algorithm and uses the RPCSEC_GSS service of rpc_gss_svc_none.
- krb5i: This flavor provides Kerberos V5 based integrity for both the RPC credentials and data using the DES_MAC_MD5 integrity algorithm and uses the RPCSEC_GSS service of rpc_gss_svc_integrity.
- krb5p: This flavor provides Kerberos V5 based integrity and privacy on both the RPC credentials and data using the DES_MAC_MD5 algorithm for integrity and 56-bit DES for privacy. It uses the RPCSEC_GSS service of rpc_gss_svc_privacy.

Note: The z/OS NFS client will not support any pseudo-flavors of the LIPKEY and SPKM-3 mechanisms in this release.

Mount attribute secure()

This section illustrates the new z/OS NFS client mount attribute secure().

The secure(flavor) keyword on the client **mount** command will be expanded so that the user can establish a secure mount point with the specified security flavor. The valid flavor values are udp, krb5, krb5i, or krb5p. If this keyword is not specified, the security flavor of AUTH_SYS will be used. This keyword is similar to the -sec=flavor mount option on other UNIX platforms.

More information about the secure(flavor) mount attribute is shown in Example 11-1.

Example 11-1 secure(flavor) mount attribute

```
secure(sys | krb5 | krb5i | krb5p | udp)
```

The secure(flavor) mount attribute specifies the transport protocol for the NFS client to use to bind reserved (privileged) ports when communicating to the NFS server. Note that:

- secure(sys) uses the system authentication.
- secure(krb5) provides Kerberos V5 based integrity on the RPC credentials (but not data) using the DES_MAC_MD5 integrity algorithm and uses the RPCSEC_GSS service of rpc_gss_svc_none. secure(krb5) is valid only for the NFS Version 4 protocol.
- secure(krb5i) provides Kerberos V5 based integrity on both the RPC credentials and data using the DES_MAC_MD5 integrity algorithm and uses the RPCSEC_GSS service of rpc_gss_svc_integrity. secure(krb5i) is valid only for the NFSv4 protocol.
- secure(krb5p) provides Kerberos V5 based integrity and privacy on both the RPC credentials and data using the DES_MAC_MD5 algorithm for integrity and 56-bit DES for privacy. It uses the RPCSEC_GSS service of rpc_gss_svc_privacy. secure(krb5p) is valid only for the NFSv4 protocol.
- secure(udp) is functionally equivalent to proto(udp).

- If secure(udp) is specified, proto(tcp) is ignored and the NFS client uses udp as the transport protocol to communicate with the NFS server.
- secure(udp) is mutually exclusive with the vers(4) parameter. secure(udp) is valid only for the NFS Version 2 and Version 3 protocols.

During mount, when sys,krb5,krb5i or krb5p is specified in the secure keyword, the client does not attempt a security negotiation.

Note: The new security flavors only apply to NFSv4 mount points. For NFSv2 or NFSv2 mount points, only a flavor of secure(*udp)* is allowed. This information is just about the using the Kerberos GSS security; NFSv2 and NFSv2 still supports UDP and TCP as the TCP/IP protocol.

For more information about the secure() security keyword, see *NFS Guide and Reference*, SC26-7417 and “Using the secure() sec keyword” on page 499.

z/OS NFS client attribute llock()

This section illustrates the new z/OS NFS client attribute llock().

The new llock(y|n) NFS client attribute keyword is added to indicate whether NFSv4 file lock requests should be managed locally (for example, llock(y)) by z/OS UNIX System Services on the client system or sent to the NFS server to be managed remotely (for example, lock(n)). The purpose of this keyword is to provide a means for deactivating the NFSv4 client locking support if problems are encountered with it. The default value is llock(n).

The new attribute can be specified as an installation parameter in the NFS client definition statement in the BPXPRMxx PARMLIB member, or as a client **mount** command keyword.

This new attribute only applies to NFSv4 mounts. For NFSv2 and NFSv2 mounts, file locks are always managed locally by z/OS UNIX System Services because the z/OS NFS client does not support NFSv2 and NFSv2 lock requests.

An example of the llock() mount attribute is shown in Example 11-2.

Example 11-2 Example of the llock() mount attribute

```
llock(Y|N)
```

The llock() mount attribute specifies whether file locking requests are managed on your local z/OS UNIX file system or remotely on the NFS server:

- Y: The local Byte Range Lock Manager (BRLM) manages the lock.
- N: The z/OS NFS client sends various NFSv4 locking operations to the remote NFS server to manage the file lock requests. The z/OS NFS client and other participating NFS clients can perform Byte Range Locking on the remote files at the supported NFS server.

The llock() attribute default value is N.

Note: If the keyword is specified on an NFSv2 or NFSv2 **mount** command, the keyword will be ignored and an informational message will be generated.

This attribute is valid only for the NFSv4 protocol. If the NFS mounted file system is NFSv2 or NFSv2 or if the NFS server does not support the NFSv4 locking, then `/usr/lpp/NFS/nfsstat` will report llock(y).

As of the writing of this book, a NFSv2 mount with the llock() keyword failed, but the keyword is just ignored. The attempt is shown in Example 11-3.

Example 11-3 NFSv2 mount with llock()

```
mount filesystem(mary) type(nfs) mountpoint('/user/mount')
parm('WTSC64:/hfs/user/hoerner,vers(3),llock(y)')
 ASYNCHRONOUS MOUNT FAILED FOR FILE SYSTEM MARY.
BPXFO28I FILE SYSTEM MARY WAS 558
NOT MOUNTED.  RETURN CODE = 0000009D, REASON CODE = 6E2E8D6E
```

11.2.8 Serviceability improvements

This section provides information about the serviceability improvements of NFS V1.10.

In a continuing effort to improve the quality and usability of z/OS NFS, the following serviceability improvements have been implemented. These improvements are grouped under this line item.

Note: Refer to *Network File System Guide and Reference*, SC26-7417 for more information about these serviceability improvements.

Enhanced Ctrace filtering

The NFS Ctrace diagnostic information often is rather overwhelming and can make it difficult to identify the specific records of interest when attempting to analyze a problem. To aid in the analysis of the immense data volume, the ability to filter the trace records based on the record type was introduced in a previous release. This line item further enhances this data reduction capability by adding the ability to filter the records based on additional parameters. Furthermore, to minimize typing errors when entering the filtering criteria, a set of panels will be added on which the desired values can be specified.

This line item enhances the filtering capability for both the NFS server and client. Some of the filter values are different for the server versus the client, because the diagnostic requirements are somewhat different. To start the Ctrace record filtering, use the *PANEL* option on the IPCS Ctrace panel, as shown in Figure 11-3.

```
------------------------------------------- CTRACE DISPLAY PARAMETERS

  System      ===>           (System name or blank)
  Component   ===> DFNFS     (Component name (required))
  Subnames    ===>

  GMT/LOCAL   ===> L                        (G or L, GMT is default)
  Start time  ===>                          (mm/dd/yy,hh:mm:ss.dddddd or
  Stop time   ===>                           mm/dd/yy,hh.mm.ss.dddddd)
  Limit       ===> 0         Exception ===>
  Report type ===> FULL      (SHort, SUmmary, Full, Tally)
  User exit   ===>           (Exit program name)
  Override source ===>
  Options         ===> PANEL

  To enter/verify required values, type any character
  Entry IDs ===>   Jobnames ===>   ASIDs ===>   OPTIONS ===>   SUBS ===>

  CTRACE COMP(DFNFS) LOCAL FULL OPTIONS((PANEL))


  ENTER = update CTRACE definition.  END/PF3 = return to previous panel.
  S = start CTRACE.  R = reset all fields.


COMMAND ===> s
```

Figure 11-3 Start NFS Ctrace filtering

z/OS NFS server Ctrace filtering

The existing z/OS NFS server Ctrace function supports record filtering based on the NFS Server Ctrace record types. These values must be specified in a comma separated list format in the Ctrace Options menu under the Options parameter. A simple typo in the specification of the values can invalidate the filtering process. To avoid this problem in the future, a z/OS NFS server filtering panel is added to select the desired filtering criteria, as shown in Figure 11-4.

```
----------------------   NFS Server Filtering Criteria   ----------------------
------------------------------------------------------------------------------
            O P T I O N S                |      S U P E R O P T I O N S
------------------------------------------------------------------------------
_ FFDC        | _ DFP_RETURN   | _ INFO    | _ CALL (ENTRY, EXIT)
_ ENTRY       | _ CB_MGMT      | _ WARNING | _ TASK_FLOW (SUSPEND, RESUME,
_ EXIT        | _ NETWORK      | _ ERROR   |              SCHEDULE, DISPATCH)
_ SUSPEND     | _ GENERAL      | _ DEBUG1  | _ USS (USS_REQUEST, USS_RETURN)
_ RESUME      | _ DETAIL       | _ DEBUG2  | _ MVS (DFP_REQUEST, DFP_RETURN)
_ SCHEDULE    | _ TRAP         | _ DEBUG3  | _ LOCK (LOCK_REQUEST, LOCK_RESUME,
_ DISPATCH    | _ BUFFER       | _ DEBUG4  |         LOCK_RELEASE)
_ USS_REQUEST | _ LOCK_REQUEST | _ DEBUG5  | _ DEBUG7 (DEBUG 1, 5, 6)
_ USS_RETURN  | _ LOCK_RESUME  | _ DEBUG6  | _ DEBUG9 (ALL OF DEBUG RECORDS)
_ DFP_REQUEST | _ LOCK_RELEASE | _ DEBUG8  |
------------------------------------------------------------------------------
MODNAME  => ________    TASKNAME  => ________ FUNCNAME   => ________________
REQBLK   => ________    REQID     => ________    OPERATION  => ________
------------------------------------------------------------------------------
Batch processing  => N (Y/N)
------------------------------------------------------------------------------
  ENTER = update Filtering Criteria definition.
  END/PF3 = return to previous panel.  S = start.   R = reset all fields.


COMMAND ===>
```

Figure 11-4 NFS server Ctrace filtering

Any combination of record types can be selected from the OPTIONS and SUPEROPTIONS sections of the menu. Within those record types, further selection filtering can be done based on the following criteria. If multiple criteria are specified, records matching any one of the specified criteria are considered to be a match and will be selected:

- *MODNAME* is the one to eight character name of one of the NFS load modules.
- *TASKNAME* is the one to eight character name of one of the NFS tasks.
- *FUNCNAME* is the 1 to 16 character name of one of the NFS functions. A name followed by an asterisk at the end matches all names that match the preceding string.
- *REQBLK* is the hexadecimal address of a specific request block.

  Note: Request blocks are reused.

- *REQID* is the hexadecimal request ID associated with a specific request received from an NFS client.
- *OPERATION* is the 1 to 22 character name of a specific NFS request/operation type.

z/OS NFS client filtering

The existing z/OS NFS client Ctrace function supports record filtering based on the NFS Client Ctrace record types. These values must be specified in a comma separated list format on the Ctrace Options menu under the Options parameter. A simple typo in the specification of the values can invalidate the filtering process. To avoid this problem in the future, an appropriate panel is added to select the desired filtering criteria. See Figure 11-5 for more information about this z/OS NFS client filtering panel.

```
 ---------------------   NFS Client Filtering Criteria
------------------------------------------------------------------------------
                O P T I O N S              |    S U P E R O P T I O N S

------------------------------------------------------------------------------
   FFDC       | _ NFS_REQUEST    | _ CB_MGMT   | _ CALL (ENTRY, EXIT)
 _ ENTRY      | _ NFS_RETURN     | _ BUFFER    | _ TASK_FLOW (SUSPEND, RESUME,
 _ EXIT       | _ USS_REQUEST    | _ NETWORK   |          SCHEDULE, DISPATCH)
 _ SUSPEND    | _ USS_RETURN     | _ GENERAL   | _ USS (USS_REQUEST, USS_RETURN)
 _ RESUME     | _ LOCK_REQUEST   | _ DETAIL    | _ NFS (NFS_REQUEST, NFS_RETURN)
 _ SCHEDULE   | _ LOCK_RESUME    | _ TRAP      | _ LOCK (LOCK_REQUEST,
 _ DISPATCH   | _ LOCK_RELEASE   | _ MSG       |    LOCK_RESUME, LOCK_RELEASE)

------------------------------------------------------------------------------
MODNAME  => ________     PID      => ________     FUNCNAME  => ________________
REQBLK   => ________     REQID    => ________     VNOP      => ________
RNODE    => ________     DIRRNODE => ________     MNTINFO   => ________

------------------------------------------------------------------------------
                       Buffer Filtering Criteria
 BNPTR    => ________     BNFLAG    => ___         BNNUM     => ________________

------------------------------------------------------------------------------
 Batch processing  => N (Y/N)

------------------------------------------------------------------------------
   ENTER = update Filtering Criteria definition.
   END/PF3 = return to previous panel.  S = start.   R = reset all fields.

 COMMAND ===>
```

Figure 11-5 z/OS NFS client filtering

Any combination of record types can be selected from the Options and Superoptions sections of the menu. Within those record types, further selection filtering can be done based on the following criteria. If multiple criteria are specified, records matching any one of the specified criteria are considered to be a match and will be selected:

- *MODNAME* is the one to eight character name of one of the NFS load modules.
- *FUNCNAME* is the 1 to 16 character name of one of the NFS functions. A name followed by an asterisk at the end matches all names that match the preceding string.
- *PID* is the one to eight character process name.
- *REQBLK* is the hexadecimal address of a specific request block.
- *REQID* is the hexadecimal request ID associated with a specific request.
- *VNOP* is the one to eight character vNode request name.

- *RNODE* is the hexadecimal address of a specific object rnode.
- *DIRRNOD* is the hexadecimal address of a specific parent directory rnode.
- *MNTINFO* is the hexadecimal address of a specific mount information block.

Note: For BUFFER record types, several additional selection criteria are available. These criteria are ignored if no buffer records are selected.

- *BNNUM* is the hexadecimal number of a specific buffer.
- *BNFLAG* is a specific hexadecimal flag field value for selecting the buffers. This value is treated as a mask for selecting buffers whose selected bits are on.
- *BNPCH* is the hexadecimal address of a specific 8 K buffer.

NFS client hang problem analysis

NFS client hang situations can arise from many different causes. Detailed analysis of the situation is required to determine the source. Some of the possible causes could be:

- Slow server or underlying file system response
- Server failure
- Network failure
- Socket hang

When a client hang occurs, the general z/OS UNIX **netstat** command, TSO **netstat** command, or z/OS shell **onetstat** command can be used to determine whether this could be a socket hang problem. If the **netstat** command is issued on the failing client system and it shows that a very large number of TCP sockets are in CLOSEWAIT state with the destination IP address of the z/OS NFS server, it indicates that sockets are probably hung (z/OS NFS server may not accept new TCP/IP connections). In this case, it is very probable that the diagnostic trace data recorded by the z/OS NFS server is already lost by the time that the hang is recognized at the client. The only thing that can be done to resolve the immediate situation is to restart the z/OS NFS server.

After the z/OS NFS server is restarted, use the sockhang operand to help capture the necessary diagnostic data before it is lost, should this situation occur again. This command tells the server to monitor the server's sockets for potential hang conditions and to create a dump when a hang is suspected so that the diagnostic trace data can be captured before it is lost.

For details about the sockhang operand, see "Sockhang operand" on page 462. For other possible causes of a client hang, use standard problem analysis techniques.

Sockhang operand

With z/OS V1.10, when the z/OS NFS server starts up, a new operand, sockhang, has an initial setting of off. Set sockhang to on if you want the server to create a dump if it detects a potential socket hang condition. This optimizes the ability to detect the situation and capture the necessary diagnostic trace data before the trace data is lost. Use this operand in the following manner:

- When sockhang is set to on, tracing levels Network, MVS (DFP_request and DFP_return), and Task_Flow (Dispatch, Resume, Schedule, and Suspend) are set on to ensure that the necessary diagnostic data is collected when a socket hang is first detected.
- When sockhang is subsequently set to off, tracing levels Network, MVS, and Task_Flow are reset to off; any previous settings of Network, MVS, and Task_Flow before sockhang was turned on must be reset as needed.

Detail level tracing provides the ideal setting to further isolate the source of a socket hang condition, but with a greater effect on system performance. The sockhang operand can be changed via the following modify commands:

```
/f mvsnsf, sockhang= on
/f mvsnsf, sockhang= off
```

Where:

- off: Will not create a dump if a socket hang is detected.
- on: Will create a dump if a socket hang is detected.

Reserving privileged ports

With z/OS V1.10, the z/OS NFS client uses a reserved (privileged) port to prevent the NFS server from rejecting a client request. The z/OS NFS client attempts to use reserved port 1023 and if that port is not available, the z/OS NFS client will subtract one from 1023 until a reserved port is available. If no reserve ports are available, the z/OS NFS client will issue the GFSC724E error message.

The number of reserved ports the z/OS NFS client can use is based on the client attribute biod. The number of reserved ports can be calculated by using the following formula:

Reserved ports = 8 + (#biod * 4)

The privileged ports should be reserved in the `tcpip.profile` file using the PORTRANGE statement. An example of a `tcpip.profile` file is shown in Figure 11-6 on page 464. The default biod(6) and eight additional ports correspond to 32 privileged ports that can be used by the z/OS NFS client. For biod(6), the `tcpip.profile` file should include the following PORTRANGE statement:

```
PORTRANGE 991 32 UDP MVSNFSC
PORTRANGE 991 32 TCP MVSNFSC
```

This allows ports 991 through 1023 to be used by the z/OS NFS client.

Note: MVSNFSC is the default z/OS NFS client startup procedure. Specify the z/OS NFS client startup procedure if it is different than the default.

```
ARPAGE 20

DATASETPREFIX TCPIPMVS

TCPCONFIG        RESTRICTLOWPORTS
UDPCONFIG        RESTRICTLOWPORTS
IPCONFIG
 SOURCEVIPA
 VARSUBNETTING
 SYSPLEXRouting
VIPADYNAMIC
  viparange define 255.255.255.248
ENDVIPADYNAMIC

; ---------------------------------------------------------------------
; AUTOLOG the following servers
;
PORT
    20 TCP OMVS               ; OE FTP Server
           DELAYACKS          ; Delay transmission acknowledgements
    21 TCP OMVS               ; OE FTPD control port
    23 TCP INTCLIEN           ; MVS Telnet Server
    53 UDP NAMED
    53 TCP NAMED
    80 TCP OMVS               ; OE Web Server
   111 TCP PORTMAP            ; Portmap Server
   111 UDP PORTMAP            ; Portmap Server
; -------------------------------------------------------
; PORTRANGE: Reserves a range of ports for specified jobnames.
;   In a common INET (CINET) environment, the port range indicated by
;   the INADDRANYPORT and INADDRANYCOUNT in your BPXPRMxx parmlib member
;   should be reserved for OMVS.
;   The special jobname of OMVS indicates that the PORTRANGE is reserved
;   for ANY OE socket application.
;   The special jobname of * indicates that the PORTRANGE is reserved
;   for any socket application, including Pascal API socket
;   applications.
;   The special jobname of RESERVED indicates that the PORTRANGE is
;   blocked. It will not be available to any application.
PORTRANGE 10000 2000 TCP OMVS ; TCP 10000 - 11999
PORTRANGE 10000 2000 UDP OMVS ; UDP 10000 - 11999
;DEVICE OSA22E0  MPCIPA
;LINK   OSA22E0LNK  IPAQGNET      OSA22E0
 DEVICE OSA2CC0  MPCIPA
 LINK   OSA2CC0LNK  IPAQENET      OSA2CC0

HOME
;  9.12.6.10     OSA22E0LNK
   9.12.6.10     OSA2CC0LNK
;  9.12.14.250   OSAL2100
```

Figure 11-6 TCP/IP profile

Privileged UDP ports

When specifying secure(udp) or proto(udp), the z/OS NFS client uses the privileged UDP ports to communicate with the NFS servers. When specifying proto(tcp), the z/OS NFS client uses the privileged TCP ports to communicate the MOUNT RPC or UNMOUNT RPC with the NFS server. However, the z/OS NFS client uses the ephemeral TCP ports to communicate NFS RPC with the NFS server. As a result, the z/OS NFS client does not work with NFS servers that require all source TCP ports to be privileged.

Note: An ephemeral (short-lived) port is a transport protocol port for Internet Protocol (IP) communications allocated automatically from a predefined range by the TCP/IP stack software. It is typically used by the Transmission Control Protocol (TCP), User Datagram Protocol (UDP), or the Stream Control Transmission Protocol (SCTP) as a port for the client end of a client-server communication when the application does not bind the socket to a specific port number, or by a server application to free up a service's well-known listening port and establish a service connection to the client host. The allocations are temporary and only valid for the duration of the connection. After the communication session completes, the ports become available for reuse. Most implementations simply increment the last used port number until the ephemeral port range is exhausted.

11.2.9 Summary of migration and coexistence characteristics

Installation of z/OS NFS V1.10 will automatically replace any older release of z/OS NFS.

If an installation contains a multi-system z/OS sysplex, and has NFS servers and clients activated on multiple systems of the sysplex, there is no requirement for all of the systems in the sysplex to be at the same NFS release level. However, sysplex wide file share and byte range locking is only supported if all of the systems are using the z/OS NFS V1.10 server.

With the implementation of NLS in this release, the mixcase/upcase site attributes have been removed. All messages will be generated in mixed case. For migration purposes, the presence of these keywords will be tolerated in the site attribute file, but they will be ignored.

A new z/OS NFS server processing attribute, znfsclient, is added so that the z/OS NFS client can tell the z/OS NFS server that it is communicating with a z/OS NFS client. This value will facilitate better z/OS to z/OS ACL handling because the z/OS platform has unique file default and directory default ACLs while other platforms use common default ACLs. The attribute will be automatically appended by the z/OS NFS client to a mount request when sent to a z/OS server. Therefore, co-existence APAR OA22759 is required for down-level z/OS NFS servers so that they will accept (and ignore) this new attribute. Means, this attribute should not be specified by the user.

11.3 NFS V1.11 enhancements

This section gives an overview of the z/OS NFS server and z/OS NFS client enhancements in V1.11.

NFS is a distributed file system that provides transparent processing capability for data and information on worldwide and heterogeneous networks. z/OS NFS provides the implementation that allows the z/OS platform to participate in these networks.

The majority of the NFSv4 protocol was incorporated into the z/OS NFS server in z/OS NFS V1.7 and in the z/OS NFS client in V1.9. Several additional functions were completed for the server and client in V1.10.

These V1.11 enhancements focus on delivering several line items for adding remaining NFSv4 functionality, performance improvements in the NFS server and client and some reliability, availability, and serviceability (RAS) cleanup in the NFS client, and continue to support the DFSMS enhancements.

11.3.1 New z/OS NFS server version parameter

With V1.11, a new command line parameter named VERSION is added in the NFS server's parm field in order to print the current versions of z/OS NFS server program modules in LOG data sets on startup. The modify command **/f nfss,flushlog** forces the trace data to be written to the log data sets.

An example of the version parameter in the parm field is shown in Figure 11-7.

```
//NFSMVS  PROC MODULE=GFSAMAIN,PARMS='DEBUG9,DSPS=600,VERSION'
```

Figure 11-7 Example of the parms field section

An example of the current versions of the z/OS NFS modules printed to the log data sets is shown in Figure 11-8.

```
GFSA947I  (I) GFSAMAIN ANMAI main_event       2467-00000039: GFSAAACC ! OA29987
! Aug 21 2009 10:07:17
GFSA947I  (I) GFSAMAIN ANMAI main_event       2467-00000039: GFSAAARN ! HDZ1B1N
! Apr 20 2009 14:20:00
GFSA947I  (I) GFSAMAIN ANMAI main_event       2467-00000039: GFSAAATT ! OA29987
! Aug 21 2009 10:07:28
GFSA947I  (I) GFSAMAIN ANMAI main_event       2467-00000039: GFSAACRE ! HDZ1B1N
! Apr 20 2009 14:20:34
GFSA947I  (I) GFSAMAIN ANMAI main_event       2467-00000039: GFSAAFAM ! HDZ1B1N
! Apr 20 2009 14:20:28
GFSA947I  (I) GFSAMAIN ANMAI main_event       2467-00000039: GFSAAIDC ! HDZ1B1N
! Apr 20 2009 14:20:00
GFSA947I  (I) GFSAMAIN ANMAI main_event       2467-00000039: GFSAALIN ! HDZ1B1N
! Apr 20 2009 14:20:35
GFSA947I  (I) GFSAMAIN ANMAI main_event       2467-00000039: GFSAALOC ! HDZ1B1N
! Apr 20 2009 14:20:00
```

Figure 11-8 Output of the version command

The operator modify command **/f nfss,version=all** will print the current versions of z/OS NFS server program modules into the syslog.

11.3.2 z/OS NFS server startfail parameter

The new startfail parameter has been added to the **start** command for the z/OS NFS server.

The **startfail** command specifies the action the z/OS NFS server should take if it encounters a terminating error during startup, or later during execution, when it did not otherwise create a dump.

The actions are:

- Ignore: No special action is taken. The z/OS NFS server proceeds with termination as it has done prior to z/OS NFS V1.11. This is the default setting if the parameter is not specified.
- Dump: For terminating errors, an SVC dump is created before terminating, if no dump was otherwise created.

An example of the **startfail** command is shown in Figure 11-9.

```
S NFSS,PARMS='STARTFAIL=DUMP'
$HASP100 NFSS     ON STCINRDR
IEF695I START NFSS     WITH JOBNAME NFSS     IS ASSIGNED TO USER STC
   , GROUP SYS1
$HASP373 NFSS     STARTED
IEF403I NFSS - STARTED - TIME=19.47.16 - ASID=0086 - SC64
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
GFSA403I (NFSS) Parse failed in line 2 830
"dlydtimeout(61)".
GFSA402I (NFSS) Read failed for the attribute data set.
IEA045I AN SVC DUMP HAS STARTED AT TIME=19.47.17 DATE=09/28/2009 832
FOR ASID (0086)
QUIESCE = YES
IEA794I SVC DUMP HAS CAPTURED: 833
DUMPID=001 REQUESTED BY JOB (NFSS    )
DUMP TITLE=z/OS NFS SERVER DUMP
$HASP395 NFSS     ENDED
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0086.
IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00010)
IEF196I         DSN (DUMP.D090928.H23.SC64.NFSS.S00001           )
IEF196I         STORCLAS (SCDUMP) MGMTCLAS (MCDB22) DATACLAS (
)
IEF196I         VOL SER NOS= SBOXZ3
IEF196I IGD104I DUMP.D090928.H23.SC64.NFSS.S00001            RETAINED,
IEF196I DDNAME=SYS00010
IEA611I COMPLETE DUMP ON DUMP.D090928.H23.SC64.NFSS.S00001 847
DUMPID=001 REQUESTED BY JOB (NFSS    )
FOR ASID (0086)
```

Figure 11-9 Example of the startfail command

Note: No dump is produced if the z/OS NFS server is stopped by the operator stop request. A dump is generated if the z/OS Network File System server shuts down because of a TCP/IP termination.

The available start parms commands are shown in Figure 11-10.

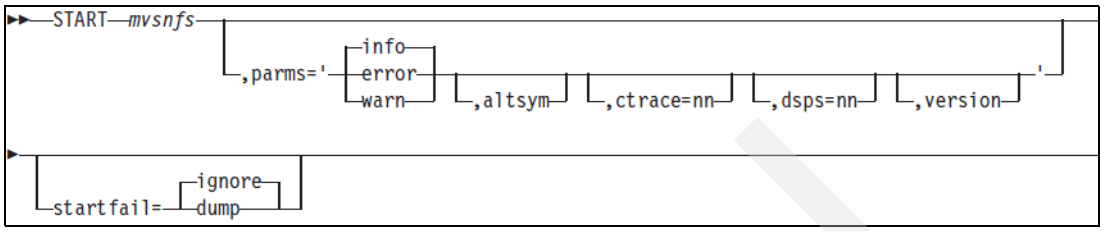


Figure 11-10 z/OS NFS server START parms

Note: These start parameters override the parameter settings in the server startup procedure.

11.3.3 z/OS NFS server restrictions

As of the writing of this book, the use of the following functions is restricted. The following information is not an complete list of remote platform restrictions.

AIX 5L V5.3 client

The following items are true:

- AIX 5L™ V5.3 cannot mount NFSv4 file systems that include symbolic links in the mount path name.
- The AIX 5L V5.3 NFS server will crash after the NFSv4 client attempts to mount a symlink without fix IZ28515 for AIX 5L V5.3.
- AIX 5L V5.3 cannot recover from the NFS4ERR_FHEXPIRED error code from the z/OS NFS server when locks are present:
  - There is an AIX interim fix available when it is a z/OS UNIX file.
  - APAR OA29666 fixes a potential NFS4ERR_FHEXPIRED problem in the z/OS NFS server.

Linux client

The following items are true:

- Red Hat EL 5.2 cannot mount NFSv4 shares that include symbolic links.
- SLES 10.2:
  - It cannot mount NFSv4 shares that include symbolic links.
  - The Kerberos security KRB5p is not supported.

Windows client

The following items are true:

- The Maestro 2008 client cannot mount to the z/OS NFS server with z/OS NFS server attributes specified on the command line.
- If security(export) mode is used on the z/OS NFS server, the mount path must exactly match the export name with the Maestro 2008 client. For example, you can export `/u/user*` in the export file, but a mount to `/u/user4` will fail.

11.4 NFS V1.11 server

This section provides information about the line items and enhancements of the z/OS NFS server in V1.11.

11.4.1 Delay Detection in the NFS server

The Delay Detection of the z/OS NFS server in V1.11 was requested by a customer via the FITS requirement MR1112064932. This function detects “excessive” delays in UNIX or MVS calls, and generates an operator message when detected.

Prior to z/OS NFS V1.11, UNIX object access is done with before-and-after time stamp calls, and if the time difference upon the return exceeds an internal value, an INFO trace is generated. Unfortunately, this informational entry was not generated until the delayed call had completed, and there was no external indication that such a call is delayed while it is still in progress. This line item creates a mechanism to detect that an object access call has been outstanding for an amount of time that exceeds the z/OS NFS server attribute DLYDTIMEOUT() value, and places an operator message on the console. When the call has completed, the operator message will be deleted from the console. Such a detection mechanism will be applied to both UNIX and MVS data access calls. The message contains both the operation and the path name for the object being delayed.

z/OS NFS server dlydtimeout operator command

An NFS server modify operator command is added to change the NFSv4 server delayed timeout value in seconds:

```
/F mvsnfs, DLYDTIMEOUT=n
```

mvsnfs is the NFS server job name, and *n* specifies the minimum Delay Detection time value in seconds before the Delay Detection mechanism observes a delay in an external call/function and prints message GFSA1030W on the console. The default value is 10 seconds. Its value (for example, n) can range from 5 to 60. If DLYDTIMEOUT is set to 0, the Delay Detection mechanism is turned off.

An output example of the operator command **/f mvsnfs,DLYDTIMEOUT** is shown in Figure 11-11.

```
F NFSS,DLYDTIMEOUT=70
GFSA1032E (NFSS) The Delay Detection timeout value must be
from 5 to 60 seconds.
```

Figure 11-11 DLYDTIMEOUT operator command

z/OS NFS server site attribute dlydtimeout

This section illustrates the new Delay Detection z/OS NFS server site attribute specified in the z/OS NFS server attribute file.

DLYDTIMEOUT(n) specifies the minimum Delay Detection time value in seconds before the Delay Detection mechanism observes a delay in an external call/function and prints message GFSA1030W on the console. The default value is 10 seconds. The value (n) can range from 5 to 60. If DLYDTIMEOUT is set to 0, the Delay Detection mechanism is turned off. Any value of DlYDTIMEOUT from 1 to 4 seconds is converted to 5 seconds.

An example of the new DLYDTIMEOUT() attribute is shown in Example 11-8 on page 501.

z/OS NFS server messages

This section illustrates the new z/OS NFS server Delay Detection messages:

- `GFSA1030W (mvsnfs) Network File System Server subtask is waiting for a reply from the resource (FileSystemTypeName).`

  Explanation: The GFSA1030W message is printed if one of the NFS worker tasks has been waiting for a reply longer than specified by the DelayDetectionTimeout parameter. In the message text, mvsnfs is the Network File System Server start up procedure name, resource is the name of file system for HFS or z/FS or DATASET name for the MVS file system, and FileSystemTypeName is the name of the type of the file system.

  System Programmer Response: If needed, collect the dump/trace regarding the type of file system being waited on.

  Note: The GFSA1030W message will be issued just once when DLYDTIMEOUT() is reached. It should stay on the console until the GFSA1031I completion message is issued.

- `GFSA1031I (mvsnfs) Network File System Server subtask wait for a reply from the resource (FileSystemType name) was completed after n seconds.`

  Explanation: The GFSA1031I message is printed after the operation completes if the GFSA1030W message has been previously printed. In the message text, mvsnfs is the Network File System Server start up procedure name. resource is the name of file system or DATASET name for the MVS file system, FileSystemTypeName is the name of the type of the file system, and n is the amount of time Network File System Server waited for reply (in seconds).

- `GFSA1032E (mvsnfs) The Delay Detection timeout value must be between 5 and 60 seconds.`

  Explanation: The GFSA1032E message is printed if an incorrect value for the Delay Detection timeout was specified in the MODIFY mvsnfs,DDTIMEOUT=x operator command. In the message text, mvsnfs is the name of Network File System Server start up procedure.

  Operator Response: Specify the correct value for the Delay Detection timeout.

- `GFSA1033E (mvsnfs) There are many delays detected. There is more information in Network File System Log.`

  Explanation: The GFSA1033E message is printed if there are more than five delays detected. The information about the new delays is printed in the Network File System Log. In the message text, mvsnfs is the Network File System Server start up procedure name.

  System Programmer Response: If needed, collect the dump/trace regarding the type of file system being waited on.

  Note: At the time of the writing of this book, the GFSA1033 message is issued as a Error message. This message will be changed to an Warning message and you might need to update your system automation accordingly.

  The five detected delays are in total. This is regardless of whether the detected delays are consecutive from the same or a separate object.

DLYDTIMEOUT documentation

At the time of the writing of this book, our tests show several inconsistencies compared to the z/OS NFS documentation:

- When you use a DLYDTIMEOUT(4), no GFSA1032E message was issued, but the z/OS server starts without a warning about the mismatch. This behavior is documented in *Network File System Guide and Reference*, SC26-7417 and is mentioned in “z/OS NFS server site attribute dlydtimeout” on page 469.
- We you use a higher value than 60 for DLYDTIMEOUT, no GFSA1032E message is issued, but the z/OS does not start. The appropriate syslog entry is shown in Figure 11-12.

```
$HASP373 NFSS     STARTED
IEF403I NFSS - STARTED - TIME=16.31.28 - ASID=006D - SC64
GFSA949I (NFSS) Command DEBUG9 not valid.
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
GFSA403I (NFSS) Parse failed in line 1
"dlydtimeout(61)".
GFSA402I (NFSS) Read failed for the attribute data set.

IEF404I NFSS - ENDED - TIME=16.31.29 - ASID=006D - SC64
-NFSS     ENDED.  NAME-
```

Figure 11-12 DLYDTIMEOUT() issue

Note: At the time of the writing this book, the GFSA403 message is issued as a Informational message. This message will be changed to an Error message and you might need to update your system automation accordingly.

11.4.2 Support for a non-VSAM EAV

This section provides information about support for a non-VSAM Extended Address Volume (EAV) of the z/OS NFS server in V1.11.

In z/OS DFSMS V1.11, EAV support is provided for non-VSAM extended-format sequential data sets. For this release, the NFS server ensures that it is compatible with this newly provided function.

There is nothing that the user needs to do to take advantage of the EAV data sets. For more information about EAV, refer to Chapter 4, “Extended Address Volume enhancements” on page 83.

11.4.3 Delegation

NFSv4 File Management Delegation support has been added to the NFS Server in V1.11. This line item will only provide this support for MVS data sets.

The NFSv4 protocol provides the ability for an NFS server to temporarily delegate management of a file's resources to an NFS client. The key purpose of the delegation is to provide improved performance by eliminating communications with the NFS server.

When a file's management is delegated to an NFS client, other than the initial data buffer and attribute information, which must be sent from the NFS server to the client, all other application file access requests can be managed locally by the NFS client while the file is delegated. The actual performance improvement obtained will vary greatly depending on the NFS requests issued and the data volumes being transferred.

NFS file management delegation background

In this section, we discuss the background of delegation support.

The NFSv4 protocol defines two types of file delegation:

- Read delegation: A Read delegation means that the file contents may only be read by applications on the NFS client because a file might be Read delegated to multiple NFS clients concurrently. This is the equivalent of shared read-only access.
- Write delegation: When a file is Write delegated, the NFS client is given exclusive access to the file and thus may modify the file contents without immediately sending the updates back to the server.

File delegation is totally at the NFS server's discretion. When an NFS client sends a file open request, it might indicate a desire to have the file delegated. The server then decides whether to delegate the file or not. The NFS client cannot assume that the file will be delegated.

The NFS server may recall the delegation at any time. The NFS client must then send any modified data buffers and attributes back to the server. It should also send any locally established file locks to the server so that those locks can be established on the NFS server to maintain the client's lock status over those files.

In conjunction with delegation, remote servers might indicate, via NFSv4 OPEN operation rflags, the server's ability to support the full POSIX byte range locking model. When the POSIX flag is off, the remote server may not support byte range lock merge and split semantics. The NFS client must then indicate to the Byte Range Lock Manager (BRLM) not to merge adjacent byte ranges. Merging lock ranges might satisfy the client requests until the delegation is recalled. However, if a merged range is sent to the remote server, subsequent proper unlock range requests might fail if the server does not support subrange locking semantics.

Note: When this partial POSIX indicator is set for an object, BRLM must fail partial range requests. In essence, the failure is indicated at the earliest detection.

An application termination indicates a condition where all locks are released for the lockowner. The NFS client should perform the same action as for delegation recalls for that lockowner's files.

z/OS NFSv4 server implementation

NFSv4 File Delegation is part of the NFSv4 protocol. Therefore, implementation of the NFSv4 File Delegation support will improve the z/OS NFS server's interoperability with NFSv4 clients.

Because the NFS server is only one of many applications running on the z/OS system and accessing its files, the ability to support file delegation is dependent on cooperative interaction between the z/OS NFS server and the underlying file system access methods, that is, UNIX LFS for z/OS UNIX file systems and the MVS access methods for MVS data sets.

The cooperative interaction is divided into three basic parts:

- Registration: When the NFS server opens a file, it must indicate its desire to receive delegation authority over that file. The underlying file system will indicate whether the delegation is granted in the open response.
- Delegation: Delegation of the file is given to the interested NFS clients.
- Recall: When some other application on the z/OS system requires access to the delegated file, the system must notify the NFS server that the delegation is being recalled. The server must then, in turn, recall the file from the delegated NFS client(s). Any modified data buffers and file attributes must be updated and any necessary file locks must be established before the recall can be completed.

The detailed interfaces for the above interactions will be different for z/OS UNIX and MVS file systems, but conceptually they will be the same.

When the z/OS NFS server receives an NFSv4 Open request that contains a delegation request, the server will first determine whether it has an operational callback registered for this client. If not, then the delegation aspect of the request will be ignored and normal open processing will proceed.

If the server does have a callback registered, it then determines whether it already has delegation authority over that file. If the server already has the authority, and this request is compatible with any of the existing client delegations for the file, then the delegation will also be granted to this client. If this request is not compatible with existing client delegations for the file, then the existing delegations will be recalled and, to avoid delegation thrashing, access will be granted to the interested clients in a non-delegated mode, in a normal file open priority order.

When the z/OS NFS server receives a delegation request and does not yet have delegation authority, it will request that authority from the underlying file system. If it is granted, the delegation will be forwarded to the NFS client. If the delegation is not granted, a normal Open response will be returned without delegation.

If an NFS client does not respond to a recall request within a reasonable amount of time based on the z/OS NFS server attribute leasetime(n), the delegation will be treated as having been successfully recalled from the standpoint of any other delegation requests. At that point, any subsequent file access operation requests from this client will fail, until the file is closed by the client.

In order to avoid excessive delegation/recall processing, a threshold time value will be defined such that a file will not be re-delegated within that time limit after the last recall for that file.

When an NFS client's lease expires, the z/OS NFS server will recall any outstanding delegations as part of the lease expiration process.

During the z/OS NFS server's startup processing grace period, it will allow NFS clients to attempt to reestablish delegations, which they had from the prior server instance. However, the granting of these delegations is not guaranteed.

Consideration: In this NFS V1.11 release, only delegation support for MVS data sets is implemented. z/OS UNIX file delegation is not provided. Also, only the AUTH_UNIX RPC flavor is supported for delegation. NFSv4 delegation reclamation after an NFS client reboots (that is, Open claim type, CLAIM_DELEGATE_PREV) is not supported due to potential issues if the client reboots in close succession.

z/OS NFS server site attributes

In this section, we talk about the new delegation and nodelegation z/OS NFS server site attributes defined in the z/OS NFS server attribute file.

Using the site attribute *delegation,* the z/OS NFS server starts with the delegation support. With *nodelegation*, the delegation support is disabled.

An example of this new attribute is shown in Example 11-8 on page 501.

NFSv4 delegation and the related operator command

In this section, we discuss the changed NFSv4 delegation operator command.

The NFS server modify operator command is changed as follows:

- The NFS Server job respectively started task name is in the **nfss** command.
- **/F nfss,v4delg=OFF** stops the delegation support. No new delegation is allowed. Existing delegations will not be recalled immediately. The command's output is shown in Figure 11-13.

```
F NFSS,V4DELG=OFF
GFSA464I (NFSS) MVS delegation was disabled.
```

Figure 11-13 Output of the v4delg command to turn delegation off

- **/F nfss,v4delg=on** starts the delegation support. This command should only be used when directed to do so by IBM Support. The command's output is shown in Figure 11-14.

```
F NFSS,V4DELG=ON
GFSA465I (NFSS) MVS delegation was enabled.
```

Figure 11-14 Output of the v4delg command to start the delegation

- **/F nfss,debug=deleg** starts the debugging delegation with a console message. This command should only be used when directed to do so by IBM support. The output of the delegation debug command is shown in Figure 11-15.

```
F NFSS,DEBUG=DELEG
GFSA796I (NFSS) DEBUG=DELEG: completed successfully.
```

Figure 11-15 Output of the delegation debug command

- **/F nfss,debug=delegoff** stops the debugging delegation with a console message. Its output is shown in Figure 11-16.

```
F NFSS,DEBUG=DELEGOFF
GFSA796I (NFSS) DEBUG=DELEGOFF: completed successfully.
```

Figure 11-16 Output of the command to stop the delegation debugging

- **/F nfss,listlock=file_name** displays share/lock/delegation information.

  file_name Is the name of an MVS data set, z/OS UNIX file, PDS member, or PDSE member. File names specified with a prefix must be specified in single quotes with the prefix (/hfs or /mvs) followed by the z/OS UNIX path name or the MVS data set name, respectively. For example, if the HFS(/hfs) site attribute was specified, LISTLOCK='/hfs/u/user' will indicate the z/OS UNIX file /u/user.

  For an MVS data set, PDS member, or PDSE member:

  - If *file_name* references a z/OS conventional MVS data set for the first time, the GFSA921I message is issued.
  - If *file_name* references a PDS or a PDSE data set (that does not include the member name), the GFSA918I message is issued.
  - If *file_name* references an invalid member name of a PDS or a PDSE data set, the GFSA919I message is issued.
  - If *file_name* is found and there are no locks, share reservations, or delegations to list, the GFSA793I message is issued.
  - When an NFS client closes a file, delegation is returned to the NFS server. Assuming that there is no local user access request, the delegation is kept by the NFS server, and not returned to MVS. In this case, the GFSA790I message is issued.

- For a z/OS UNIX file, if there are no locks to list, the GFSA793I message is issued.

  If the specified file does not have any locks, the GFSA793I message is issued to report that no locks exist.

  Because the lock information may only be reported to the server log data set or component trace buffer, and not back to the console, the GFSA794I message is issued to indicate that the listlock function completed successfully.

Delegation support messages

This section illustrates the new file delegation messages.

GFSA462E

The syntax of this message is:

```
GFSA462E   MVS delegation disabled.  Return code=0xh_digit1.
```

Explanation: An error occurred when the NFS Server was registering ENF 51 to listen for system events. In the message text, Return code, h_digit is documented for the ENFREQ in z/OS MVS Authorized Assembler Services Reference. The capability of delegation for MVS data set is disabled.

System Action: The Network File System server continues processing.

GFSA463E

The syntax of this message is:

```
GFSA463E   MVS delegation disabled due to short of storage.
```

Explanation: There is not enough memory to support NFSv4 file management delegation. The capability of delegation for MVS data set is disabled.

System Action: The Network File System server continues processing.

GFSA464I

The syntax of this message is:

```
GFSA464I   MVS delegation was disabled.
```

Explanation: NFSv4 file management delegation for MVS data sets is disabled by the V4DELG=OFF modify operator subcommand.

System Action: The Network File System server continues processing.

GFSA790I

The syntax of this message is:

```
GFSA790I   NETWORK FILE SYSTEM SERVER holds delegation= access
```

Explanation: This message displays the fact that the NFS server holds access type delegation for the file requested in the LISTLOCKS operator command. This does not necessarily mean that the NFSv4 delegation is currently granted to a remote NFS Client. In the message text, access indicates the delegation access type:

- R: Read
- RW: Read/Write

System Action: The Network File System server continues processing.

GFSA792I

The syntax of this message is:

```
GFSA792I   Owner(serverid clientid userid processid)  access=access  deny=deny
delegation= delg_access
```

Explanation: This message displays a lock share holder for the file requested in the LISTLOCKS operator command. The GFSA792I message will be returned for each share holder. The share owner is identified by serverid, clientid, userid, and processid. The lock access mode access and deny mode deny are displayed.

The delegation access information was added to this message. In the message text, Delg_access indicates the delegation access type:

- R: Read
- RW: Read/Write

System Action: The Network File System server continues processing.

GFSA793I

The syntax of this message is:

```
GFSA793I (procname) text does not have Locks.
```

Explanation: This message is returned from the LISTLOCKS operator command for the specified file text if the file does not have any locks or shares.

System Action: The Network File System server continues.

GFSA794I

The syntax of this message is:

```
GFSA794I (procname) LISTLOCK for text found count shares/locks. Command was
completed successfully.
```

Explanation: This message indicates that processing of the LISTLOCKS operator command for the specified file text has been completed. count is the number of shares or locks that were found and were listed with he GFSA791I (for byte range locks) or GFSA792I (for shares) messages.

System Action: The NFS server continues.

GFSA918I

The syntax of this message is:

```
GFSA918I (procname) dsname text1 IS NOT A VALID NAME.
```

Explanation: The text1 specified in the release, the listlock, or the unmount operand of the modify command is not a valid name:

- For the release operand, only a valid MVS data set name is allowed.
- For the listlock operand, only a valid MVS data set name or a UNIX file name is allowed.
- For the unmount operand, only a valid MVS data set name or a UNIX directory is allowed.

In the message text, *procname* is the name of the start procedure.

System Action: NFS processing continues.

Operator response: Specify the command again with a valid data set name.

GFSA919I

The syntax of this message is:

```
GFSA919I (procname) text IS NOT A VALID MEMBER NAME.
```

Explanation: The member name text specified in the release, the listlock, or the unmount operand of the modify command is not a valid MVS member name.

In the message text, *procname* is the name of the start procedure.

System action: NFS processing continues.

Operator response: Specify the command again with a valid member name.

GFSA921I

The syntax of this message is:

```
GFSA921I (procname) NO ACTIVE DATA SETS.
```

Explanation: This message is in response to the list=dsnames or the listlock operand of the modify command and shows that there are no clients actively accessing data sets.

In the message text, *procname* is the name of the start procedure.

System Action: NFS processing continues.

11.4.4 Ctrace performance improvement

This section provides information about the Ctrace performance improvement of the z/OS NFS server in V1.11. Basically, it is not a Ctrace improvement, but rather a common z/OS NFS server performance improvement, so that you do not write DEBUGx records to the LOG data sets.

Note: All DEBUGx levels are still recorded when using the Ctrace with options(all).

The server startup procedure parameter 'DEBUGn' (where n is a number from 1 to 9) is no longer supported as of this release. Anyone using DEBUGx for this parameter should change it to either 'ERROR', 'WARN', or 'INFO', or the GFSA949I message will occur and the server logging will be set to INFO, which is the default. An example of the syntax specifying the 'DEBUGx' parameter is:

```
//MVSNFS PROC MODULE=GFSAMAIN,PARMS='DEBUG9'
```

Note: DEBUGx (old Errlog) functionality has been identified as a performance issue.

The operator command to switch the tracing from INFO to DEBUGx is also not supported as of V1.11. The output of the modify command **`/f nfs_proc,log= debug9`** is shown in Figure 11-17.

```
F NFSS,LOG=DEBUG9
GFSA797I (NFSS) Command LOG=DEBUG9 is obsolete. Instead,
use the TRACE CT operator command to capture the desired diagnostics.
```

Figure 11-17 Output of log=debug command

11.4.5 Elimination of MVSLOGIN and MVSLOGOUT for NFSv4 Kerberos

This section provides information about the elimination of MVSLOGIN and MVSLOGOUT from the z/OS NFS server in V1.11.

Consideration: This line item eliminates the need to issue the mvslogin and mvslogout NFS client utility commands just for RPCSEC_GSS requests directed to the z/OS NFS Server. For non-Kerberos supported system authentication (sys), mvslogin and mvslogout are still required.

Background: Elimination of MVSLOGIN and MVSLOGOUT for Kerberos

When the security site attribute of the z/OS NFS server is set to saf or safexp, the server provides strong system access security by performing SAF checking based on the user's privileges. Prior to the support provided by the RPCSEC_GSS authentication flavor, the z/OS NFS server only supported system authentication. The system authentication framework is based on the Trusted Host model, where a client user is identified by its User Identifier on its local host system that is sent in the RPC packet, with nothing in the verifier to enable the server to authenticate the identity being claimed by the client user. To overcome the weakness of the Trusted Host model of system authentication, the z/OS NFS server in its SAF mode of operation required the NFS client users to provide their RACF user identifier and the password via the MVSLOGIN client utility. The z/OS NFS server would then use RACF services to verify the password for the RACF User Identifier and hence validate the identity being claimed by the client user. After a client user was done with its session, it would typically issue the client utility MVSLOGOUT to enable the server to log it out.

For more information about Kerberos security see “RPCSEC_GSS security” on page 455, *NFS Guide and Reference*, SC26-7417, and RFC 2203 RPCSEC_GSS protocol specification, found at the following address:

http://www.faqs.org/rfcs/rfc2203.html

In order to start a z/OS NFS server with Kerberos authentication features, Kerberos must be active and a Kerberos Key Distribution Center must be set up. For more information about how to set up a Kerberos Key Distribution Center, see “Setting up a Kerberos Key Distribution Center”, “Configuring a secure z/OS NFS server”, and “Configuring a secure z/OS NFS client” in *NFS Guide and Reference*, SC26-7417 and 11.4.8, “Setting up a secure NFS environment in our test environment” on page 486.

In the Security Access Facility (SAF) modes of operation of the z/OS NFS server, the server required that the NFS client users issue an MVSLOGIN to enable the z/OS NFS server to verify their identities and enforce system access checks based on the privileges that these users were defined with, to the RACF on the z/OS Sysplex in which the NFS server was running. The NFS client users were then expected to issue an MVSLOGOUT after they were done.

However, for the RPCSEC_GSS framework, using the Kerberos security mechanism provides strong authentication. The NFS client users have an associated Kerberos principal that is also defined to RACF in the z/OS sysplex in which the NFS server is running. The NFS server validates these users by querying the mapping of the associated principal to a valid RACF user identifier and enforces system access checks based on the privileges that these users were defined to RACF. A cryptographic validation of RPC credentials of the incoming RPCSEC_GSS requests performed by the NFS server helps validate the identity of the NFS client user being claimed in the incoming RPC request. Therefore, an MVSLOGIN and a subsequent MVSLOGOUT do not need to be issued for RPCSEC_GSS requests directed to the z/OS NFS server.

With the support of this line item, it is no longer necessary to issue an MVSLOGIN and MVSLOGOUT for RPCSEC_GSS requests to a SAF z/OS NFS server. Users or applications that initiate RPCSEC_GSS workloads to a SAF z/OS NFS server should be made aware of this support to be able to exploit it. However, there is no harm in issuing MVSLOGIN and MVSLOGOUT, even for RPCSEC_GSS, and it is therefore not necessary to remove them from existing applications.

Note: Prior to z/OS NFS V1.11, both MVSLOGIN and **`kinit`** need to issued using the Kerberos flavors of the RPSEC_GSS requests. With the support of this line item, MVSLOGIN and MVSLOGOUT are eliminated, but the **`kinit`** command is still necessary to obtain or renew the Kerberos ticket-granting ticket. For more information about the **kinit** command, see *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926.

Setting up a KDC, a secure zNFS server, and a secure zNFS client

To use the Kerberos flavors, a Kerberos Key Data Center (KDC) must be available and both the z/OS NFS server and the z/OS NFS client must be secure in order to work with this KDC. All three steps are shown in “Setting up a Kerberos Key Distribution Center”, “Configuring a secure z/OS NFS server”, and “Configuring a secure z/OS NFS client” in *NFS Guide and Reference*, SC26-7417. A successful setup is shown in “Setting up a secure z/OS NFS client” on page 486.

A non-secure z/OS NFS client will start, but with the GFSC284I message will be issued, followed by the GFSA700I message. A example of the startup is shown in Example 11-4.

Example 11-4 Insecure z/OS NFS client

```
$HASP373 MVSNFSC7 STARTED
IEF403I MVSNFSC7 - STARTED - TIME=19.58.08 - ASID=008B - SC64
IEE252I MEMBER CTINFC00 FOUND IN SYS1.IBM.PARMLIB
GFSC284I NETWORK FILE SYSTEM CLIENT COULD NOT GET GSS CREDENTIALS 431
```

```
FOR THE NFS CLIENT : GSS API krb5_get_in_tkt_with_keytab() FAILED
WITH GSS MAJOR STATUS 96C73A06 GSS MINOR STATUS 00000000
GFSC700I z/OS NETWORK FILE SYSTEM CLIENT (HDZ1B1N) started. 0A29432,
```

Note: For more information about the Kerberos status codes, see Chapter 6, “Status Codes”, in *z/OS Integrated Security Services Network Authentication Service Administration,* SC24-5926.

If the z/OS NFS client is secure, the GFSC284I message will not appear during startup.

The GFSA730I message shown during a secure z/OS NFS server startup means that the server will work properly with the KDC. An example is shown in Example 11-5.

Example 11-5 Secure z/OS NFS server

```
$HASP373 NFSS     STARTED
IEF403I NFSS - STARTED - TIME=17.35.07 - ASID=008A - SC64
GFSA949I (NFSS) Command DEBUG9 not valid.
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
GFSA730I (NFSS) NETWORK FILE SYSTEM SERVER KERBEROS INITIALIZATION
SUCCESSFUL
GFSA348I (NFSS) z/OS Network File System Server (HDZ1B1N, 0A29988)
started.
```

When the GFSA730I message is missing during startup, this means that you have an insecure z/OS NFS server. The *GFSA737I* message appears in the log of the z/OS NFS server after running **flushlog**. Example 11-6 shows the appropriate extract of the z/OS NFS server log.

Example 11-6 Insecure z/OS NFS server

```
07:14:18 GFSA737I  (W) GFSAMAIN AGSS  krb5_tkt 911-00000000:
(MVSNFSS) NETWORK FILE SYSTEM SERVER COULD  NOT GET KERBEROS TICKET IN ROUTINE
krb5_get_default_realm(), KERBEROS RETURN CODE(96C73ADF)
```

A domain name server (DNS) resolver should also be available to the z/OS system in order to enable the security feature. Otherwise, the GFSA735I message appear during the startup of the secure z/OS NFS server. The Kerberos KDC must be running, and must contain the z/OS NFS server’s principal before the secure z/OS NFS server starts. If the KDC is not set up correctly, whether the z/OS NFS server can start depends on the hfssec, mvssec, and pubsec attribute settings. If any of these three z/OS NFS server attributes also contains the sys security flavor in addition to any of the Kerberos flavors, the z/OS NFS server starts and the GFSA737I message appears, and only the functions with the sys security flavor are present. On the other hand, if none of the hfssec, mvssec, or pubsec attributes contain the sys security flavor and the KDC is not available, the GFSA736E message appears and the z/OS NFS server does not start. The KDC can run on z/OS, either on the same host as the z/OS NFS server itself or remotely from the z/OS NFS server. It can also be a KDC running on other platforms, for example, a SUN Solaris system or any other platform.

A summary of how to set up a secure environment as a prerequisite to testing the elimination of the **mvslogin** command is described in 11.4.6, “Configuring a secure z/OS NFS client ” on page 481.

11.4.6 Configuring a secure z/OS NFS client

In order for the z/OS NFS client to support the RPCSEC_GSS authentication flavor using the Kerberos V5 Security Mechanism, you must perform the following steps. We assume that a properly configured KDC is already set up in your environment and that you are using a KDC provided by a security server and Integrated Security Services (RACF) provided by IBM. The following steps need to be performed on the system that contains the KDC, unless otherwise noted.:

1. Add the client principal mvsnfsc to the Kerberos database on the KDC, with a defined password. This principal should not use randkey as the password. For example, for the z/OS NDBM type or SUN KDC, issue the **addprinc mvsnfsc** command in the kadmin interface, then enter the desired password in the prompt. For the z/OS SAF type KDC, the password can be defined in the PASSWORD field of the RACF **adduser** or **altuser** commands.
2. Map the principal mvsnfsc in lower case to a RACF user. For example:

   ```
   ADDUSER mvsnfsc OWNER(ibmuser) OMVS(UID(100)) ALTUSER mvsnfsc
   PASSWORD(password) NOEXPIRED KERB(KERBNAME('mvsnfsc')) PASSWORD USER(mvsnfsc)
   NOINTERVAL
   ```

   – Change mvsnfsc to refer to the z/OS NFS client in order to meet your installation requirements.

   – The RACF user ID mvsnfsc that the z/OS NFS client uses requires a unique UID in the OMVS segment.

   – Including the NOINTERVAL option will prevent the password from expiring.
3. While in the OMVS shell, the system administrator must add the principal mvsnfsc to the keytab. If /etc/skrb/krb5.keytab does not exist, create a new one. For example:

   ```
   IBMUSER:/ :> keytab add mvsnfsc -p password -k /etc/skrb/krb5.keytab  -v 1
   ```

   – The password must match the password entered in Step 2, when the principal was added to the SAF KDC Kerberos database. This principal mvsnfsc is used to perform mounts for all users. To gain access to a secure mount point, all users should run the **kinit** command to acquire their Kerberos credentials.

   – The key version used to create the keytab must be the same key version in the RACF database. The -v option of the **keytab** command is used to specify the key version when adding a principal to a keytab. Issue the following RACF command to see the current key version:

   ```
   LU mvsnfsc NORACF KERB
   ```

   – The password used with the **keytab** command is case sensitive. If mixed case password support is not in effect, you must enter the password in uppercase.

   – Secure copy the new keytab to the z/OS NFS Client and place it in /etc/skrb/krb5.keytab.
4. The z/OS NFS client requires that the Kerberos configuration file krb5.conf be configured to match your site's Kerberos environment:

   ```
   Sample /etc/skrb/krb5.conf file to be put on the z/OS NFS client system:
   [libdefaults]
   default_realm = KRB390.IBM.COM
   kdc_default_options = 0x40000010
   use_dns_lookup = 0
   default_tkt_enctypes = des-cbc-crc
   default_tgs_enctypes = des-cbc-crc
   [realms]
   ```

```
KRB390.IBM.COM = {
kdc = dcesec4.krb390.ibm.com:88
kpasswd_server = dcesec4.krb390.ibm.com:464
admin_server = dcesec4.krb390.ibm.com:749
}
KRB2000.IBM.COM = {
kdc = sstone1.krb2000.ibm.com:88
admin_server = sstone1.krb2000.ibm.com:749
}
[domain_realm] .
krb390.ibm.com = KRB390.IBM.COM .
krb2000.ibm.com = KRB2000.IBM.COM
[capaths]
KRB390.IBM.COM = {
KRB2000.IBM.COM = .
}
```

Note: If the z/OS NFS server will be supporting multiple platforms of NFS clients, use DES-CBC-CRC encryption types only for compatibility.

11.4.7 Configuring a secure z/OS NFS server

In order for the z/OS NFSv4 server to be able to provide RPCSEC_GSS security authentication flavors such as krb5, krb5i and krb5p, the z/OS NFS server must be configured to communicate with the Kerberos facilities. To accomplish this task, you must perform the following steps.

We assume that a properly configured KDC is already set up in your environment and that you are using a KDC provided by “Security Server and Integrated Security Services” (RACF) provided by IBM. These steps assume that Resource Access Control Facility (RACF) is available in the system. If you have a different but equivalent external security manager, refer to its product documentation for instructions. A domain name server (DNS) resolver should also be available to the z/OS system in order to enable the security feature. Otherwise, the GFSA735I message appears during startup of the secure z/OS NFS server. Because there are many options to set up a DNS resolver, such as by setting `/etc/resolv.conf` or GLOBAL TCPIPDATA, specific examples are not given here. For more information about setting up a DNS resolver, see *z/OS Communications Server: IP Configuration Guide*, SC31-8775.

Perform the following steps:

1. The Kerberos key distribution center (KDC) must be running, and must contain the z/OS NFS server’s principal before the secure z/OS NFS server starts. If the KDC is not set up correctly, whether the z/OS NFS server can start depends on the hfssec, mvssec, and pubsec attribute settings. If any of these three attributes also contains the sys security flavor in addition to any of the Kerberos flavors, the z/OS NFS server is started, shows the GFSA737I message, and only uses the sys security flavor. On the other hand, if none of the hfssec, mvssec, or pubsec attributes contain the sys security flavor and the KDC is not available, the GFSA736E message is shown and the z/OS NFS server does not start. The KDC can be running on z/OS, either on the same host as the z/OS NFS server itself or remotely from the z/OS NFS server. It can also be a KDC running on other platforms, for example, a SUN Solaris system or any other platform. To set up other platforms' KDCs, refer to the specific platform’s documentation.

2. Define the local realm and default policy.

   Run the following TSO command:

   ```
   RDEFINE REALM KERBDFLT KERB(KERBNAME(KRB390.IBM.COM) PASSWORD(password)
   ```

   KRB390.IBM.COM"should match the Kerberos REALM of the KDC.

3. Define IRR.RUSERMAP and grant READ authority to all system users by running the following TSO commands:

   ```
   RDEFINE FACILITY IRR.RUSERMAP UACC(READ) SETROPTS RACLIST (FACILITY)  REFRESH
   PERMIT IRR.RUSERMAP CLASS(FACILITY) ID(mvsnfs) ACCESS(READ) SETROPTS CLASSACT
   (FACILITY)
   ```

   ID mvsnfs is the RACFID of the z/OS NFS server.

4. Add this path in the z/OS UNIX /.profile `PATH=/usr/lpp/skrb/bin:$PATH` and export the PATH.

5. Create RACF user IDs with Kerberos segments for the z/OS NFS server.

   For example, to add a RACF ID for the z/OS NFS server (if one does not already exist), run the following TSO command:

   ```
   AU (mvsnfs) OWNER(owner) OMVS(UID(101))
   ```

   To add the needed Kerberos segment to this user definition, run the following TSO commands:

   ```
   ALTUSER mvsnfs PASSWORD(password) NOEXPIRED
   KERB(KERBNAME(nfs/hostname.domain))
   PASSWORD USER(mvsnfs) NOINTERVAL
   ```

   If the Kerberos segment is not defined correctly to RACF, the following error message appears on the server when a NFS client tries to mount to z/OS NFS server with Kerberos:

   ```
   GFSA728E SAF APPLICATION USER MAPPING FAILED WITH SAF RETURN CODE 8,
   RACF RETURN CODE 8, AND RACF REASON CODE 16.
   ```

   Note:

   - The ALTUSER command converts the password to upper case if the MIXEDCASE SETROPTS option is not set. If MIXEDCASE is not set, you must ensure that the upper case value is used when you request an initial ticket. The principal name is not converted to upper case and the realm name is not included. You must change the password for the user in order to create the Kerberos secret key.
   - The Kerbname must be the fully qualified host name and domain. For example hostname.domain could be host1.ibm.com.
   - Including the NOINTERVAL option will prevent the password from expiring.

6. The z/OS NFS server requires that Kerberos configuration file krb5.conf be configured to match your site's Kerberos environment.:

   ```
   Sample /etc/skrb/krb5.conf file to be put on the z/OS NFS server host:
   [libdefaults]
   default_realm = KRB390.IBM.COM
   kdc_default_options = 0x40000010
   use_dns_lookup = 0
   default_tkt_enctypes = des-cbc-crc
   default_tgs_enctypes = des-cbc-crc
   [realms]
   KRB390.IBM.COM = {
   ```

```
kdc = dcesec4.krb390.ibm.com:88
kpasswd_server = dcesec4.krb390.ibm.com:464
admin_server = dcesec4.krb390.ibm.com:749
}
KRB2000.IBM.COM = {
kdc = sstone1.krb2000.ibm.com:88
admin_server = sstone1.krb2000.ibm.com:749
}
[domain_realm] .
krb390.ibm.com = KRB390.IBM.COM .
krb2000.ibm.com = KRB2000.IBM.COM
[capaths]
KRB390.IBM.COM = {
KRB2000.IBM.COM = .
}
```

Note: If the z/OS NFS server will be supporting multiple platforms of an NFS client, use DES-CBC-CRC encryption types only for compatibility.

7. Generate the keytab from the KDC and put it in /etc/skrb of the z/OS NFS server unless otherwise defined.

   From the OMVS shell, the system administrator must add the principal nfs/hostname.domain into the keytab. If /etc/skrb/krb5.keytab does not exist, create a new one. For example:

   ```
   IBMUSER:/ :> keytab add nfs/hostname.domain -p password -k
   /etc/skrb/krb5.keytab  -v 1
   ```

   Where:

   – The password must match the password entered in Step 5, when the principal was added to the SAF KDC Kerberos database. This principal is used to authenticate the z/OS NFS server to the KDC.

   – The key version used to create the keytab must be the same key version as in the RACF database. The -v option of the **keytab** command is used to specify the key version when adding a principal to a keytab. Issue the following RACF command to see the current key version:

   ```
   LU mvsnfs NORACF KERB
   ```

   – The password used with the **keytab** command is case sensitive. If mixed case password support is not in effect, you must enter the password in uppercase.

8. For systems with multiple TCP/IP stacks, you must create the keytab with principals for each stack. If a stack is part of a different REALM, then keys will need to be added to the keytab from each KDC. Cross REALM trust must also be created. Use the following command:

   ```
   IBMUSER:/:>klist -k
   Key table: /etc/skrb/krb5.keytab
   Principal: nfs/host1.domain.com@ KRB390.IBM.COM
   Key version: 4
   Principal: nfs/host2.domain.com@ KRB390.IBM.COM
   Key version: 4
   Principal: nfs/host3.domain.com@ KRB2000.IBM.COM
   Key version: 2
   ```

9. If there is any multi-realm setup in the environment, the z/OS NFS server needs to have the foreign principals mapped to a RACF ID. For example, to map a foreign principal fprinc in KRB2000.IBM.COM to RACF ID fprealm2, issue the following TSO commands:

   ```
   ADDUSER (fprealm2) OWNER(owner) OMVS(UID(102))
   ALTUSER prealm2 PASSWORD(password) NOEXPIRED
   PASSWORD USER(prealm2) NOINTERVAL
   RDEFINE KERBLINK /.../KRB2000.IBM.COM/fprealm2 APPLDATA(''fprealm2')
   ```

   To map the entire foreign realm (every principal in the trusted foreign realm) to a RACF user, issue the following TSO command:

   ```
   RDEFINE KERBLINK /.../KRB2000.IBM.COM/ APPLDATA('fprealm2')
   ```

   Where:

   – The /.../ and trailing slash are required.

   – KRB2000.IBM.COM is the foreign realm.

10. Start the z/OS NFS server. If its setup is correct, the following message should appear:

    ```
    GFSA730I NETWORK FILE SYSTEM SERVER KERBEROS INITIALIZATION SUCCESSFUL
    ```

    – These are the minimal requirements to set up a secure z/OS NFS server in order for it to communicate with Kerberos facilities. For more advanced configurations, see *z/OS Integrated Security Services Network Authentication Service Administration*, SC24-5926.

    – If a z/OS NFS server is configured to use a KDC that resides on a remote host, the local KDC procedure (for example, skrbkdc) on the same host as the z/OS NFS server should not be started.

    – For Linux (Enterprise Linux 5) users, as of July 2009, Linux NFSv4 is still considered experimental. Although base NFSv4 function is considered complete, RPCSEC_GSS is still under development. Linux Kerberos works differently from other UNIX platforms:

      - In order to perform a secure Kerberos mount, Linux requires a keytab generated by the Kerberos KDC to be put in the Linux machine's `/etc` directory. This behavior will change after kernel keyring support is completed.
      - Because Linux has the keytab, the system is able to perform secure NFS mount without having the credentials acquired by the **`kinit`** command. This behavior will change after kernel keyring support is completed.
      - The root user (UID = 0) uses Linux machine credentials, but not the regular user credentials obtained by **`kinit`**. Thus, the root user will be able to browse the NFS mount point without running **`kinit`**. Regular users will need **`kinit`** to access the mount points. This behavior will change after kernel keyring support is completed.
      - The **`kdestroy`** command will not destroy the context in the Linux kernel. This behavior will change after kernel keyring support is completed.
      - Extra configurations are needed for Linux remote realm setup, because Linux sends `nfs/host.domain` instead of user principal during mount time. If the Linux's NFS principal is not defined to RACF, z/OS NFS server will reject mount requests. A simple way to solve this problem is to map the entire remote realm to RACF. Another more secure way to work around this limitation is to map individual Linux machines to a special realm in the [domain_realm] section in `/etc/skrb/krb5.conf`, and map that realm to a special RACF on the z/OS NFS server, thus leaving all other machines in the remote realm intact.

– For Windows users, use the MIT Leash Kerberos Ticket Manager with Hummingbird Maestro NFS client, especially when using Windows XP. However, to use Hummingbird NFS Maestro Version 9 or below with Windows XP, a user might need to use Microsoft® SSPI instead of the MIT Leash Kerberos Ticket Manager. For more information about Hummingbird NFS Maestro client and the configurations of the Kerberos on Windows platform, consult *Hummingbird NFS Maestro User's Guide*. If the z/OS NFS server security site attribute is set to SAF or SAFEXP, the only supported authentication level on Windows platform is authsys. RPCSEC_GSS authentication levels such as krb5, krb5i, and krb5p are not supported, because Hummingbird Maestro 10 and higher requires NIS/NIS+ or LDAP with Kerberos instead of PCNFSD, which serves as the mvslogin tool for Windows platform. Thus, if z/OS NFS server is SAF enabled and the network transmission protection is mandatory, the Windows platform is not recommended.

11.4.8 Setting up a secure NFS environment in our test environment

Using the methodology described in 11.4.7, "Configuring a secure z/OS NFS server " on page 482, set up a secure NFS environment in our lab by using the commands in this section.

Setting up a secure z/OS NFS client

Run the following commands:

```
AU mvsnfsc OWNER(ibmuser) OMVS(UID(100))
ALTUSER mvsnfsc PASSWORD(hugo1234) NOEXPIRED KERB(KERBNAME('mvsnfsc') PASSWORD
USER(mvsnfsc) NOINTERVAL
keytab add mvsnfsc -p HUGO1234 -k /etc/skrb/krb5.keytab  -v 1
```

Setting up a secure z/OS NFS server

Run the following commands:

```
RDEFINE REALM KERBDFLT KERB(KERBNAME(KRB390.IBM.COM) PASSWORD(juergen1)
RDEFINE FACILITY IRR.RUSERMAP UACC(READ)
SETROPTS RACLIST (FACILITY) REFRESH
AU (mvsnfss) OWNER(ibmuser) OMVS(UID(101))
PERMIT IRR.RUSERMAP CLASS(FACILITY) ID(mvsnfss) ACCESS(READ)
SETROPTS CLASSACT (FACILITY) REFRESH
SETROPTS RACLIST (FACILITY) REFRESH
ALTUSER mvsnfss PASSWORD(berta123) NOEXPIRED
KERB(KERBNAME(nfs/mcevsfd.mainz.de.ibm.com ))
keytab add -p BERTA123 -k /etc/skrb/krb5.keytab -v 1 nfs/mcevsfd.mainz.de.ibm.com
```

Note that in the above commands that mcevsfd is the host name of the z/OS NFS server.

Listing the principals of the z/OS client and server

Run the following command:

```
/etc/skrb:==>klist -k -e
 Key table: /etc/skrb/krb5.keytab
Principal: mvsnfsc@KRB390.IBM.COM
  Key version: 1
Encryption type: des-cbc-crc
Principal: mvsnfsc@KRB390.IBM.COM
  Key version: 1
Encryption type: des-hmac-sha1
Principal: mvsnfsc@KRB390.IBM.COM
```

```
  Key version: 1
Encryption type: des3-cbc-sha1-kd
Principal: mvsnfsc@KRB390.IBM.COM
  Key version: 1
Encryption type: aes128-cts-hmac-sha1-96
Principal: mvsnfsc@KRB390.IBM.COM
  Key version: 1
Encryption type: aes256-cts-hmac-sha1-96
Principal: nfs/mcevsfd.mainz.de.ibm.com@KRB390.IBM.COM
  Key version: 1
Encryption type: des-cbc-crc
Principal: nfs/mcevsfd.mainz.de.ibm.com@KRB390.IBM.COM
  Key version: 1
Encryption type: des-hmac-sha1
Principal: nfs/mcevsfd.mainz.de.ibm.com@KRB390.IBM.COM
  Key version: 1
Encryption type: des3-cbc-sha1-kd
Principal: nfs/mcevsfd.mainz.de.ibm.com@KRB390.IBM.COM
  Key version: 1
Encryption type: aes128-cts-hmac-sha1-96
Principal: nfs/mcevsfd.mainz.de.ibm.com@KRB390.IBM.COM
  Key version: 1
Encryption type: aes256-cts-hmac-sha1-96
```

Checking the Kerberos ticket-granting ticket of the principals

Run the following command:

```
/etc/skrb:==>kinit -t /etc/skrb/krb5.keytab nfs/mcevsfd.mainz.de.ibm.com
/etc/skrb:==>
/etc/skrb:==>kinit -t /etc/skrb/krb5.keytab mvsnfsc
/etc/skrb:==>
Note:The ticket is OK when there is no response.
BROWSE    /etc/skrb/krb5.conf
 [libdefaults]
 default_realm = KRB390.IBM.COM
 kdc_default_options = 0x40000010
 use_dns_lookup = 0
 ; Enable DES encryption types (AES, DES3 and DESD are disabled)
 default_tkt_enctypes = des-cbc-md5,des-cbc-md4,des-cbc-crc
 default_tgs_enctypes = des-cbc-md5,des-cbc-md4,des-cbc-crc
 [realms]
 KRB390.IBM.COM = {
 kdc = mcevsf.mainz.de.ibm.com:88
 admin_server = mcevsf.mainz.de.ibm.com:749
 }
 [domain_realm]
 .mainz.de.ibm.com = KRB390.IBM.COM
```

Securing the z/OS NFS client startup

Run the following command:

```
IEF403I MVSNFSC - STARTED - TIME=13.40.48
IEE252I MEMBER CTINFC00 FOUND IN SYS1.PARMLIB
GFSC700I z/OS NETWORK FILE SYSTEM CLIENT (HDZ1B1N) started. OA29985, GFSC4X0P, Sep
17 2009  12:04:56 .
```

Securing the z/OS NFS server startup

Run the following command:

```
IEF403I MVSNFSS - STARTED - TIME=13.39.18
IEE252I MEMBER CTINFS00 FOUND IN SYS1.PARMLIB
GFSA730I (MVSNFSS) NETWORK FILE SYSTEM SERVER KERBEROS INITIALIZATION SUCCESSFUL
GFSA348I (MVSNFSS) z/OS Network File System Server (HDZ1B1N, OA30216) started.
```

11.4.9 Completion messages for operator commands

This section provides information about the completion messages for operator commands of the z/OS NFS server in V1.11. This is in response to the customer generated FITS requirement MR0521071541.

The GFSA796I generic successful completion message now confirms when the operator command is successful.

There were a number of NFS Server operator commands that did not give a positive indication that the command was processed. Now all z/OS NFS server operator commands will have successful completion messages.

Here are some of the documented operator commands that now have a completion message:

- **/f nfss,SMF=ON** resumes the collection of SMF records.
- **/f nfss,SMF=OFF** suspends the collection of SMF records.
- **/f nfss,log=info/warn/error** switches between the levels of the trace log.

An example of the GFSA796I completion message is shown in Figure 11-18.

```
F NFSS,SMF=ON
GFSA796I (NFSS) SMF=ON: completed successfully.
```

Figure 11-18 GFSA796I completion message

11.4.10 NFS server mount symlink support

This section provides details about the mount symlink support of the z/OS NFS server in V1.11. Several examples are shown in 11.4.13, “Prefix examples and test results” on page 494.

Prior to z/OS NFS V1.11, due to differences in the mount path resolution process introduced with the NFSv4 protocol, the z/OS NFS server did not support symbolic links in the mount path. This line item will implement a solution for that problem and eliminate that restriction.

The z/OS NFS server supports access to both z/OS UNIX file systems and MVS data sets. The two data types are very different and require different management techniques. Therefore, they are treated as two different file system types and are managed independently. The NFS server distinguishes between the two because the z/OS UNIX file system paths are prefixed with an hfs prefix value (for example, `/hfs/u/mhlres5/mount`) and MVS data sets are not (for example, `mvsdsn`). The hfs prefix is not actually part of the path name. It is only intended as a trigger to tell the z/OS NFS server that the specified path is a z/OS UNIX path, not an MVS data set. Based on the presence, or absence, of a prefix, the NFS server invokes its appropriate data management functions.

In the NFSv2 and NFSv2 protocols, file systems were mounted using the separate mount protocol. In these protocols, the NFS client sends the entire mount path name to the NFS server as a single string in the mount request. If an hfs prefix is present at the front of the string, the prefix is removed and then the remaining path name is passed to the z/OS UNIX LFS for resolution. If the path contains a symbolic link, it is resolved during the resolution process. Some examples of the resolution process are shown in 11.4.12, “NFSv4 path name resolution characteristics and restrictions” on page 490.

By contrast, in the NFSv4 protocol, there is no actual mount request. Instead, mounts are handled as a sequence of lookup requests, one per mount path name qualifier. The sequence starts from the single common root of all the file systems. If a symbolic link is encountered in the path, that fact is returned to the NFS client, which must then restart its mount path resolution based on the symbolic link definition. This is where a problem arises for z/OS NFS. As stated above, the z/OS NFS server uses the hfs prefix to tell it whether to start its lookup processing in the z/OS UNIX path or in the MVS data set catalog. Because the symbolic link definition will not contain an hfs prefix, the z/OS NFS server will incorrectly interpret the path as an MVS data set path. This is why symbolic links are not currently supported in the mount path under NFSv4.

As stated, the basic issue with symbolic links in the mount path is their lack of the hfs prefix in the link definition. Merely requiring the customer to add the prefix in the link definition is not a viable solution because that would cause problems for local applications on the server system which attempted to use the symbolic link because the prefix value would look like an ordinary path qualifier, and would not be found. Having the NFS server pre-end symbolic link definitions with the hfs prefix before returning them to the NFS client is also not viable because the NFS server cannot tell whether the definition refers to the NFS server file system hierarchy or to a different path on the NFS client. Therefore, the only viable solution appears to be to remove the requirement for the hfs prefix.

This line item will resolve this issue by expanding the z/OS NFS file system type management function by adding an mvs prefix and a customer configurable path resolution heuristic. The new mvs prefix provides the capability for the customer to explicitly specify a prefix for identifying MVS data sets, like the hfs prefix does for z/OS UNIX files. The new customer configurable method allows the customer to specify how to interpret absolute path names that do not have a prefix specified. By setting the heuristic to indicate that MVS data sets now require a prefix and z/OS UNIX files do not, reversing the current algorithm, symbolic links in z/OS UNIX mount paths can then be supported. Making just this change is not desirable because it would have a large impact on the existing mount statements. Therefore, additional options (shown in 11.4.11, “New NFS server site attributes” on page 490) are provided.

All path type resolution (checking) processing will use the presence, or absence, of a leading slash (/) to indicate whether the path is an absolute or relative path. If the slash is present, the first qualifier after the slash will be compared against the hfs and mvs prefixes to determine whether it matches one of them. If so, then the path type will be considered to be explicitly resolved via the prefix. If no match is found, or no slash was present, the implicit path type resolution heuristic is used.

To avoid causing unnecessary customer impacts, the default settings will cause the file system type resolution algorithm to give the same results as the resolution algorithm in previous releases.

11.4.11 New NFS server site attributes

The new MVS prefix and the customer configurable path resolution heuristic will be controlled by several new NFS server site attributes:

HFSPREFIX(prefix) This attribute is being introduced as a synonym for the HFS(prefix) site attribute to be more compatible with the other new attributes and to better indicate its purpose. The default value of the prefix is /hfs.

MVSPREFIX(prefix) This attribute specifies an MVS data set prefix to be appended to the front of an MVS data set name for a mount path directory. The default value of the prefix is /mvs.

IMPPREFIX(impprefix) This attribute specifies how a mount path that does not have a path type prefix should be interpreted, where impprefix is:

NONE An explicit prefix must always be specified for an absolute path. Implicit prefix resolution is not valid in this case.

HFS If no explicit prefix is present, assume the path is a z/OS UNIX file system.

MVS If no explicit prefix is present, assume the path is an MVS data set. This is the default and corresponds to the way the z/OS NFS server functions in previous releases.

HFS, MVS If no explicit prefix is present, first assume the path is a z/OS UNIX file system. If no matching z/OS UNIX file system can be found, assume that it is an MVS data set.

MVS, HFS If no explicit prefix is present, first assume the path is an MVS data set. If no matching high level qualifier can be found, assume that it is a z/OS UNIX file system.

Note: When this setting is used, the MVS HLQ referenced in path names where no prefix is explicitly specified must preexist. Otherwise, the algorithm will default to the second option (HFS) when the MVS HLQ is not found.

The above attributes apply to all NFSv2, NFSv2, and NFSv4 path names and their resolution processing. This includes directories specified in the exports list and check list in the exports file, as well as mount and root lookup objects.

An example of how these attributes are specified in the z/OS NFS server attribute file is shown in Example 11-8 on page 501.

Attention: The above attributes must be specified before the 'PUBLIC(…)' site attribute and will apply to the resolution of any public paths specified by this attribute.

11.4.12 NFSv4 path name resolution characteristics and restrictions

To better understand the path name resolution process, some examples based on NFSv4 compounds will be used. The first thing to note is that in NFSv4, path names are split into a sequence of lookup operations, one per path name qualifier. The lookup operations may be combined in a single NFSv4 compound request, or split across multiple requests, at the NFS client's discretion.

For simplicity, the following examples split the lookup operations across multiple compound requests. However, the same results apply if they were combined in a single compound:

- Assume that this compound request (PUTROOTFH; LOOKUP <hfsprefix> ; GETFH) is followed by this compound request (PUTFH; LOOKUP <abc>). The lookup in the first compound request detects that the HFS prefix is specified and will set the file system type for this file handle to z/OS UNIX. GETFH returns the file handle for the z/OS UNIX root file system.

  Regardless of the IMPPREFIX() setting, even if the lookup at the second qualifier fails, the search at this point is locked into the z/OS UNIX file system type and will not switch to the MVS file system type, or vice versa.

- Assume that this compound request (PUTROOTFH; LOOKUP <mvsprefix> ; GETFH) is followed by this compound request (PUTFH; LOOKUP <abc>). The lookup in the first compound detects that the mvs prefix is specified and sets the file system type for this file handle to MVS. GETFH returns a special file handle denoting the MVS root file system.

  Regardless of the IMPPREFIX()setting, even if the lookup at the second qualifier fails, the search at this point is locked into the MVS path and will not switch to the z/OS UNIX file system type, or vice versa.

- Assume that this compound request (PUTROOTFH; LOOKUP <abc> ; GETFH) is followed by this compound request (PUTFH; LOOKUP <def>). The lookup in the first compound detects that no prefix was specified and uses the IMPPREFIX heuristic for determining whether abc is a z/OS UNIX or an MVS object and sets the file system type for this file handle accordingly. GETFH returns that file handle.

  Regardless of the IMPPREFIX() setting, even if the lookup at the second compound fails, the search at this point is locked into a file system type and will not switch to the other file system type, or vice versa.

  Note: If IMPPREFIX(MVS,HFS) is specified, and <abc> is an MVS HLQ, <abc> must pre-exist so that it can be found in the MVS catalog during the lookup processing. Otherwise, the algorithm will switch to the second option and assume <abc> to be a z/OS UNIX object and set the file system type accordingly.

  This MVS HLQ pre-existence requirement is a new restriction that does not exist in previous z/OS NFS releases, but it only applies when IMPPREFIX(MVS,HFS) is set and the path name being parsed does not include an explicit prefix.

- Assume that this compound request (PUTROOTFH; GETFH) is followed by this compound request (PUTFH; LOOKUP <abc>). Because the first compound does not include a lookup operation, no file system type determination can be made at this point. GETFH returns the special file handle denoting the z/OS NFS root file handle.

  In this case, the file system type determination will be deferred until the lookup in the second compound. The PUTFH with the special z/OS NFS Root file handle will be treated the same as though it were a PUTROOTFH.

All of the above examples apply identically if PUTPUBFH is substituted for PUTROOTFH.

Processing characteristics

The following additional processing characteristics and restrictions must be considered when using the new path name prefix processing support provided by the new NFS server site attributes:

1. When both path options are available based on the IMPPREFIX() site attribute (for example, IMPPREFIX(mvs,hfs) or (hfs,mvs)), only the existence/nonexistence of the first path name qualifier is used to determine whether the second option is tried. In other words, if the first path qualifier exists, but the next one does not, the object will be considered to not exist and the mount/lookup will fail.

   Assume ibmuser.abc is a valid MVS HLQ and /ibmuser is valid in z/OS UNIX space, but /ibmuser/abc does not exist and that IMPPREFIX(hfs, mvs) is set. The mount request **`mount ... -o'server:"/ibmuser/abc[/d][,servattr]"[,clntattr]' /mntpt`** fails because /ibmuser is found in z/OS UNIX space locking in the path type to z/OS UNIX. Then when the next path qualifier (for example, abc) lookup is processed, it will fail, causing the mount to fail.

2. Prior to this release, an MVS mount to a HLQ (for example, a.b.c) for which no data sets exist, was considered valid and would mount to that HLQ, allowing the first data set to be created via NFS. As of this release, with the site attribute IMPPREFIX(MVS,HFS), NFSv4 mounts to such a HLQ will fail on the MVS side and will then attempt to mount z/OS UNIX node /a.b.c. If that z/OS UNIX node does not exist, the mount will fail. If this behavior is not desired, either an mvs prefix must be specified on the path, or one of the other IMPPREFIX site attribute values must be used.

3. For IMPPREFIX(HFS,MVS), if the object does not exist (it not z/OS UNIX or MVS), it will create the mount point as a new MVS HLQ with no entries, just as an MVS mount does in prior releases.

   On the other hand, for IMPPREFIX(MVS, HFS), if the object does not exist (it is not z/OS UNIX or MVS), it will try MVS first, then z/OS UNIX, then fail, just as a z/OS UNIX mount for a non-existent object does in prior releases.

   Once NFS has switched to option 2, it cannot switch back to option 1.

4. For NFSv2 and NFSv2 mount requests, NFS clients send the entire mount path to the NFS server as a single string. By contrast, for NFSv4 mount requests, NFS clients send a series of lookup requests (there is no mount request) to the NFS server for one path qualifier at a time. Consequently, the NFS server does not know whether additional path qualifiers will follow. This can produce some unexpected results. Assume ibmuser is a valid MVS HLQ and /ibmuser is valid in the z/OS UNIX space. When you run **`mount ... -o'server:"/ibmuser[/...][,servattr]"[,clntattr]' /mntpt`**, it will be interpreted as a z/OS UNIX mount if IMPPREFIX(hfs) or IMPPREFIX(hfs,mvs) is specified, but it will be interpreted as an MVS mount if IMPPREFIX(mvs) or IMPPREFIX(mvs,hfs) is specified, even if additional path qualifiers are specified on the mount statement. This mount will fail if additional path qualifiers are specified unless ibmuser/ is a valid MVS PDS or PDSE data set name.

   Note: To avoid this potential problem, do *not* define MVS high level qualifiers and z/OS UNIX root nodes with the same name.

5. Assume that the following symbolic link is defined on the NFS server:

   ```
   ls -s /a symlink ,
   ```

/a exists both as an MVS HLQ and as a z/OS UNIX directory, and IMPPREFIX(MVS, HFS) is set in the NFS server site attribute file. Now assume the following mount command is issued from a remote client to the server:

```
mount   ...  -o'server:"/hfs/[c/]symlink[/d][,servattr]"[,clntattr]'   /mntpt
```

If the mount request is issued as an NFSv2 and NFSv2 mount request, the path will be handled as a single string entity and will be resolved by z/OS UNIX, resolving the symbolic link to the z/OS UNIX /a directory, ignoring the IMPPREFIX() setting. This is effectively no change from prior releases.

The same thing is true for NFSv4 mount requests if the symbolic link is the last name in the path and it is just followed by processing attributes.

Note: If the NFS client is z/OS, the znfsclient processing attribute will always be automatically added by the client identifying it as z/OS.

However, if the mount request is issued as an NFSv4 mount request and the symbolic link is not the last name in the path, or it is not followed by any processing attributes, then the symbolic link will be identified as such back to the NFS client. The client will then read the link data and reinitiate the path resolution. In this case, assuming the link is defined as an absolute path, then the path type resolution will come into play based on whether a prefix is included and based on the implicit prefix resolution heuristic. If there is no prefix processing attribute present using NFSv4, the z/OS NFS does not know whether it is the last qualifier and the path resolution for the symbolic link will fail.

Note: This can cause the symbolic link to resolve into MVS space, not just z/OS UNIX space.

6. The Implicit prefix heuristic also applies to the exports file, for example, for export entries that do not include an explicit prefix. The IMPPREFIX( ) site attribute will be used to determine the specified path. If both the HFS and the MVS options are specified, then the Export entry will apply to both types of file systems, assuming that the specified entry exists in both file system spaces. Internally, the entry is assumed to exist in both spaces. Its true existence is not checked until the export entry is used for validation a mount.

   Here is an example of this scenario:

   Assume IMPPREFIX(hfs,mvs) is specified in the site attribute file. Assume the export entry is /ibmuser. If /ibmuser exists in the root z/OS UNIX file system, the above entry will export that directory. If a high level qualifier (HLQ) IBMUSER exists in the MVS catalog, this HLQ will also be exported. If only HFS or MVS is desired, the export entry should be specified with the appropriate prefix.

7. When the z/OS NFS server restarts, it attempts to recover mount points recorded in the MHDB. If the hfs or mvs prefix or implicit prefix site attributes were changed before the restart, the new mount points reflect the new HFS and MVS prefixes. Implicit prefix changes have no effect. MHDB entries are always treated as though they have explicit prefixes.

   NFSv4 mount points that were established without specifying the mvsmnt processing attribute were not recorded in the MHDB. The NFS client attempts to re-establish these mount points when it receives a NFS4ERR_FHEXPIRED response from the NFS server. However, the NFS client has no knowledge of the changed prefix site attributes and will use the original mount name string in this attempt. This can result in the NFS client not being able to reestablish the mount points.

Note: This only applies when the z/OS NFS server prefix site attributes are changed during the server restart. Otherwise, the NFS client should be able to re-establish the mount points.

8. If the IMPPREFIX(NONE) site attribute is specified, then all path names, including those in the exports file (if used), must be specified with a prefix.

NFSv4 name resolution using a symbolic link summary

A lookup is done by the NFS client for all the parts of the path. When the z/OS NFS server responds that it is a symbolic link, the NFS client looks up its absolute path, which has no prefix. Therefore, the implicit prefix IMPPREFIX() becomes important. Using both MVS and z/OS UNIX symbolic links, use either IMPPREFIX(MVS,HFS) or IMPPREF(HFS,MVS).

z/OS NFS server messages

This section lists the NFS server messages.

GFSA444E

The syntax of this message is:

```
GFSA444E   Invalid IMPPREFIX specification in line d_digit.
```

Explanation: The IMPPREFIX site attribute specified on line d_digit in the site attribute file contains an invalid value.

System Action: The Network File System server initialization fails and the server terminates.

System Programmer Response: Check and correct the IMPPREFIX value specified in the Site Attribute File and then restart the NFS server.

GFSA445E

The syntax of this message is:

```
GFSA445E  Site Attribute IMPPREFIX(NONE) is set. Required path prefix is missing
in the path name: text.
```

Explanation: The IMPPREFIX site attribute indicates that all path names must be specified with an explicit hfs or mvs prefix, but this path name was specified without a prefix. The path type (z/OS UNIX or MVS) cannot be determined. text is the path name string that is in error.

System Action: The request fails. NFS processing continues.

System Programmer Response: Either change the IMPPREFIX value specified in the Site Attribute file, or add an explicit prefix to this path name.

11.4.13 Prefix examples and test results

In this section, we show symlink prefix examples from our lab environment.

z/OS UNIX mounts

The following z/OS UNIX mount using a symlink with NFSv2 and the prefix /hfs was successful:

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount')
parm('WTSC64:/hfs/u/mhlres5/nfs/symlink,vers(3)')
```

The same mount using a symlink but NFSv4 instead of NFSv2 failed:

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount')
parm('WTSC64:/hfs/u/mhlres5/nfs/symlink')
ASYNCHRONOUS MOUNT FAILED FOR FILE SYSTEM MARY.
```

Syslog output

The syslog output is:

```
BPXF028I FILE SYSTEM MARY WAS NOT MOUNTED.  RETURN CODE = 00000081, REASON CODE =
6E050002
```

Reason code 6E050002

The meaning of the z/OS NFS reason code 6E050002 is:

6E	z/OS NFS client
05	NFS protocol error
0002	ENOENT No such file or directory

Probable cause: A component of a specified path name does not exist, or the path name is an empty string.

Action: Ensure that the file/directory exists.

Note: The hfs prefix is the default of hfs(/hfs), which is the same as HFSPREFIX(/hfs), *but* the new independent implicit prefix IMPPREFIX() is not specified; its default is MVS.

Symlink absolute path mount

Mounting the absolute path of the symlink with NFSv4 with the /hfs prefix instead the following mount is successful:

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount')
parm('WTSC64:/hfs/u/mhlres5/hoerner')
```

Using the implicit prefix IMPPREFIX(hfs,) the following mount is successful because the absolute path underneath is z/OS UNIX:

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount')
parm('WTSC64:/hfs/u/mhlres5/nfs/symlink')
```

Using IMPPREFIX(none), the following mount failed because it cannot be determined whether it is MVS or z/OS UNIX that is missing the prefix:

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount')
parm('WTSC64:/u/mhlres5/nfs/symlink')
ASYNCHRONOUS MOUNT FAILED FOR FILE SYSTEM MARY.
```

Syslog output

The syslog shows the following output:

```
BPXF028I FILE SYSTEM MARY WAS NOT MOUNTED.  RETURN CODE = 00000081, REASON CODE =
6E050002
```

Note: In this case, we expect to see the GFSA445E message, but the BPXF028I RC81 RSN6E05002 message is issued instead. At the time of the writing of this book, the GFSA445E message was not shown in the syslog or in the z/OS NFS server trace log.

At the time of the writing of this book, when the z/OS NFS server starts with the wrong value of IMPPREFX(one), a parse failure is detected and the z/OS NFS server does not start. This startup sequence is shown in Figure 11-19.

```
$HASP373 NFSS     STARTED
IEF403I NFSS - STARTED - TIME=16.50.24 - ASID=0097 - SC64
GFSA949I (NFSS) Command DEBUG9 not valid.
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
GFSA403I (NFSS) Parse failed in line 1 235
"IMPPREFIX(one)".
GFSA402I (NFSS) Read failed for the attribute data set.
GFSA949I (NFSS) Command DEBUG9 not valid.
GFSA934I (NFSS) NFSLOG1 or NFSLOG2 DD statements not defined.
IEE252I MEMBER CTINFS00 FOUND IN SYS1.IBM.PARMLIB
GFSA401I (NFSS) CANNOT OPEN THE ATTRIBUTES DATA SET.
-NFSS              GFSAMAIN    16    664    .00    .00    .01
 0      0      0      0     0     2
IEF404I NFSS - ENDED - TIME=16.50.26 - ASID=0097 - SC64
```

Figure 11-19 Using the wrong value for IMPPREFIX

Note: In this case, we expect to see the GFSA444E message in the syslog, but the GFSA403I message is issued instead. The GFSA444E message is just issued to the z/OS server trace log.

At the time of the writing of this book, the GFSA403 message is issued as an Info message. It is IBM's intention to change this message to an Error message and you might need to update your system automation accordingly.

At the time of the writing of this book, the startup sequence shown in Figure 11-19 indicates a problem with the z/OS server log data sets. The GFSA934I message is misleading in this context because it disappears when using a correct IMPPREFIX value, which means that there is no problem with the both z/OS NFS server trace logs.

MVS mounts

With no HFS(prefix), no HFSPREFIX(), no IMPPREFIX(), and no prefix in the path name of the mount, the following MVS mount is successful because the default of IMPPREFIX() is mvs:

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount/MVS')
parm('WTSC64:/mhlres5.SC63')
```

The following mount is successful because the prefix in its path name is /mvs, which is the default when not using IMPPREFIX():

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount/MVS')
parm('WTSC64:/mvs/mhlres5.SC63')
```

Using IMPPREFIX(hfs), the following mount failed because it was not able to find the path as z/OS UNIX:

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount/MVS')
parm('WTSC64:/mhlres5.SC63')
ASYNCHRONOUS MOUNT FAILED FOR FILE SYSTEM MARY.
```

The syslog shows the appropriate reason code:

```
BPXF028I FILE SYSTEM MARY WAS 630
NOT MOUNTED.  RETURN CODE = 00000081, REASON CODE = 6E050002
```

With IMPPREFIX(hfs,mvs), the same mount is successful because the second attempt searched for a MVS data set:

```
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount/MVS')
parm('WTSC64:/mhlres5.SC63')
```

11.5 NFS V1.11 client enhancements

This section describes the line items and enhancements of the z/OS NFS client in V1.11.

11.5.1 NFS client message globalization

This section discusses the message globalization of the z/OS NFS client in V1.11.

The z/OS NFS client only supports the generation of console messages in English. The IBM corporate globalization strategy for the z/OS platform is to provide National Language Support (NLS) for console messages (for example, Japanese translation only). This line item will modify the z/OS NFS client message support to become compliant with that strategy.

11.5.2 Security negotiation in the z/OS NFS client

The security negotiation is done via the SECINFO NFS operation in the z/OS NFS client.

Note: The SECINFO support in the z/OS NFS server was previously delivered via APAR OA19876.

The support for this line item will enable the client to negotiate security policy with NFS servers. In addition, it will also allow the z/OS NFS client to be able to dynamically retry any requests that failed because of changes in the security policy on the server name space. This situation will therefore enable the z/OS NFS client to operate better with other NFS implementations.

The NFSv4 protocol facilitates the usage of multiple security mechanisms and thereby offers the flexibility of application of different local security policies at the server. This flexibility requires that an NFS client be able to dynamically negotiate the appropriate security mechanism based on mechanism availability and policy deployment at the server. This line item will implement the support for security negotiation in the z/OS NFS client and the ability to administer the server name space with the deployment of different security policies for the mount points in the server's file system name space.The protocol therefore provides for an NFS client to ask what security the server requires for a given file object. This communication between the client and the server (also known as security negotiation) is achieved by the use of the SECINFO operation of the NFSv4 protocol. With this operation, the client indicates the object for which it seeks to know the permitted security flavors to access that object, and the NFS server, via the results of the SECINFO operation, specifies the permissible RPC security flavors for that object and any required information.

Security negotiation via SECINFO in the z/OS NFS client

The flexibility provided by the existence of multiple security mechanisms on the server's name space presents the challenge of negotiating the appropriate security mechanism between the client and server. The problem stems from the fact that it is unreasonable to require that clients have advance knowledge of the security policy deployed at the server's end. In addition to that, it is also possible for the security policy to change across file system boundaries on the server.

Prior to the support of this line item, the z/OS NFS client required the specification of a valid Kerberos security flavor for a secure NFSv4 mount in the **mount** command. If a security flavor was not provided in the **mount** command, the NFS client assumed the default security flavor to be that of system authentication(sys). Additionally when the NFS client received the error NFS4ERR_WRONGSEC from the server because the server's security policy was different from what the client was currently using for that file object, the NFS client failed that request and did not have the mechanism to retry the failed request with the security flavor conforming to the security policy in use at the server's end.

z/OS NFS client implementation of security negotiation via SECINFO

This line item implements the security negotiation in the z/OS NFS client. This security negotiation using the SECINFO operation will be performed by the z/OS NFS client in the instances shown in this section.

Mount point establishment

When a security flavor is not specified for an NFSv4 mount, the z/OS NFS client will query the supported security flavors on the NFS server using the SECINFO operation. Among the security flavors returned by the server, it chooses a security flavor based on the following order of preference:

- sys: System authentication.
- krb5: Kerberos V5 security providing integrity protection on the RPC header.
- krb5i: Kerberos V5 security providing integrity protection on the RPC header and the RPC data.
- krb5p: Kerberos V5 security providing encryption protection on the RPC data and integrity protection on the RPC header.

The chosen security flavor will serve as the designated flavor for the NFS client for all future accesses to this mount point.

NFS4ERR_WRONGSEC handling

An NFS request is failed with NFS4ERR_WRONGSEC by the NFS servers, if the security policy on the mount point at the server's end does not allow the authentication flavor with which the request was issued. Upon encountering this error, the z/OS NFS client will negotiate security with the NFS server. This action will be accomplished by issuing a SECINFO operation to query the server supported security flavors. The z/OS NFS client will then choose a security flavor from the server supported flavors based on the order of preference specified in “Mount point establishment” on page 498 and will retry the failing request with this newly chosen security flavor. This security flavor will now serve as the designated security flavor for all future accesses to that mount point.

Using the secure() sec keyword

While there are no changes to the **mount** command, the implication of not specifying the sec keyword in the parameters of the **mount** command has changed. Prior to the support of this line item, not specifying the ***sec*** keyword caused the z/OS NFS client to use the system authentication(sys) flavor for all RPC calls. However with the support of this line item, not specifying the sec flavor will cause the z/OS NFS client to negotiate security with the z/OS NFS server and use the most preferred flavor among the flavors being returned by the NFS server. The NFS client's preferred flavors in the descending order of preference are as follows:

- sys: System authentication (No Integrity or Privacy protection).
- krb5: Provides checksum protection on the RPC header.
- krb5i: Provides checksum protection on the RPC header and the RPC data.
- krb5p: Provides checksum protection on the RPC header and encryption on the RPC data.

For more information about the sec keyword, see *NFS Guide and Reference*, SC26-7417 and “Mount attribute secure()” on page 456. Example 11-7 is an example of the **secure() mount** command.

Example 11-7 Examples of the secure() mount command

```
in the z/OS NFS server export list:
/hfs/u -sec=krb5

in the UNIX mount command:
mount -t nfs4 -o sec=krb5 ,proto=tcp mvshost1:/u/smith/mnt

in the TSO mount command:
mount filesystem(mary) type(nfs) mountpoint('/u/mhlres5/mount')
parm('WTSC64:/hfs/u/mhlres5/hoerner,secure(krb5)')
```

11.5.3 NFS client reason codes

This section provides details about the more meaningful reason codes of z/OS NFS in V1.11.

This line item makes new reason codes available to z/OS UNIX system services and modifies certain NFS client reason codes, which is more useful to service and customer users in identifying the root causes of the associated errors.

Some errors are returned to the z/OS UNIX system services file system caller with reason codes that are the NFS “file-and-line” type, requiring that analysis be done to identify the code point reporting the error. Analysis was done to identify error scenarios that can be converted to fixed and published reason codes that can be meaningful to customers and service representatives. Not all z/OS NFS error conditions are candidates for this change.

Note: The format of the NFS reason codes has changed in z/OS NFS V1.11.

The z/OS NFS reason codes now are generic eight-digit hexadecimal codes that provide an indication of the problem location. They appear now in the following new format 6*nxxyyyy*, where:

6D Indicates NFS client and server common shared modules (that is, GFSNxxxx, for example, GFSNETRC).

6E Indicates NFS client modules (that is, GFSCxxxx).

6F Indicates NFS server modules (that is, GFSAxxxx).

xx This is a two-digit hexadecimal number with one of the following values:

- **00 - 0F** The values have a special meaning. This range applies only to the z/OS NFS client reason codes. For more information, see “Special reason codes (xx is 00-0F)” in *NFS Guide and Reference*, SC26-7417.
- **10 - 3F** Identifies the NFS client module where the reason code was generated. This range is intended for IBM Service use only.
- **40 – FF** Identifies the NFS server module where the reason code was generated. This range is intended for IBM Service use only.

yyyy This is a four-digit hexadecimal number with one of the following values:

- **0000 - 7FFF** NFS reason codes.
- **0000 - 0FFF** Reason codes that match UNIX System Services JRxxxx reason codes.
- **1000 - 3FFF** Global reason codes that have the same meaning independent of the module ID.
- **4000 - 4FFF** Module specific reason codes. A given value has different meanings depending on the module ID.

Examples of the NFS V1.11 reason codes

Several of the more meaningful NFS reason codes are shown in Table 11-1.

Note: For more information about the more meaningful z/OS NFS reason codes, see Chapter 21, “Reason Codes” of *Network File System Guide and Reference*, SC26-7417.

Table 11-1 Example of meaningful NFS client global reason codes

yyyy	Name	Description
100D	JRNfs_UnicodeFail	The Unicode Conversion Service fails on translation. Probable cause: The code page IDs (ccsid) in the z/OS NFS parameters (specified as clnccsid() in the BPXPRMxx NFS section or in the MOUNT command) do not match the code page IDs that are available in Unicode Conversion Service. Action: Set ccsids to the proper value and restart the z/OS NFS client.

yyyy	Name	Description
100E	JRNfs_NoLocalHost	The z/OS NFS component cannot obtain the local host name. Probable cause: There is an error in the resolver configuration file (HOSTname keyword). Action: Ensure that the resolver configuration file is correct and restart the z/OS NFS client.
100F	JRNfs_NoLocalDomain	The z/OS NFS component cannot obtain a local domain name. Probable cause: There is an error in the TCP/IP configuration data sets (keywords Domain or DomainOrigin). Action: Ensure that the TCP/IP configuration data sets are correct and restart the z/OS NFS client.

11.5.4 Converting MemMgmt S0C1 to S0F4 or U801 abend

The line item discusses converting the NFS client V1.11 memory management S0C1 abend to a S0F4 or U801, which is an internal only change within the NFS client memory management error handler.

Convert S0C1 abends in the z/OS NFS client into S0F4 or U801 abends so that the UNIX System Services LFS provides better reliability, availability, and serviceability (RAS) on the source of the Memory Management failure. When you receive either a S0F4 or an U801 abend, a dump will be captured.

11.6 Sample z/OS NFS server attribute file

This section illustrates an example of the z/OS NFS server attribute file.

The z/OS NFS server attribute file is defined in the z/OS NFS server started task by the //NFSATTR DD statement, for example, `//NFSATTR  DD DISP=SHR,DSN=OS390NFS.SC64MVS.PARMS(ATTRIB`, and contains the attributes the z/OS NFS server use during its startup, as shown in Example 11-8.

Example 11-8 Example of the z/OS NFS server attribute file

```
nodelegation
# DELEGATION tells the server to enable delegation support.
# NODELEGATION tells the server not to enable delegation support.
# If neither is specified, the default is NODELEGATION.
DlyDTimeout(10)
#DlyDTimeout(10) Specifies the minimum Delay Detection time value
# in sec before the Delay Detection mechanism observes a delay in
# an external call/function and prints message GFSA1030W on
# console.
# The maximum value for DlyDTimeout is 60 seconds.
# The minimum value for DlyDTimeout is 5 seconds.
# Any value of DlyDtimeout from 1 to 4 seconds is
# converted to 5 seconds.
# The Default value is 10 seconds.
# If DlyDTimeout is set to 0 the DelayDetection mechanism
#    is turned OFF
```

```
hfsprefix(/hfs)
# The HFSPREFIX (or HFS) attribute specifies a z/OS UNIX file
# system prefix to be imbedded in the mount directory path name.
# The default value of the z/OS UNIX file system prefix is /hfs.
# Mount requests received by the Network File System beginning
# with this prefix value are identified as mount requests for
# z/OS UNIX.  The prefix value is not part of the path name.
#
# Note: The file system must be mounted locally by z/OS UNIX.
#       Otherwise, the client mount fails.
# hfsprefix(prefix)  or  hfs(prefix)
mvsprefix(/mvs)
# The MVSPREFIX attribute specifies an MVS data set prefix
# to be imbedded in the mount directory path name.
# The default value of the MVS data set prefix is /mvs.
# Mount requests received by the Network File System beginning
# with this prefix value are identified as mount requests for
# MVS data sets.  The prefix value is not part of the path name.
# mvsprefix(prefix)
impprefix(mvs)
# The IMPPREFIX attribute specifies how mount directory path
# names which do not have either an HFSPREFIX or an MVSPREFIX
# are to be interpreted.  If no prefix is present in the path
# name, the path is to be interpreted according to the IMPPREFIX
# setting:
#    HFS     - assume it is a z/OS UNIX file system.
#    MVS     - assume it is an MVS data set.
#    HFS,MVS - try z/OS UNIX file system first.  If that fails
#              or is not found, try MVS data set.
#    MVS,HFS - try MVS data set first.  If that fails or is not
#              found, try z/OS UNIX file system.
# The default value is MVS, which matches pre-z/OS V1.11
# functionality.
#
# Note: For the MVS,HFS setting, the MVS data set must exist
#       if the path is referring to an MVS data set.  Prior to
#       this attribute being added, it was not necessary for the
#       data set to pre-exist.  However, now, if it does not
#       exist, the 2nd option (HFS) will be used.
```

11.7 Migration and coexistence considerations

This section discusses the migration steps and coexistence consideration when installing z/OS NFS V1.11.

11.7.1 Sysplex considerations

Installing z/OS NFS V1.11 will automatically replace any older release of z/OS NFS.

If a customer installation contains a multi-system z/OS sysplex, and has NFS servers/clients activated on multiple systems of the sysplex, there is no requirement for all of the systems in the sysplex to be at the same NFS release level. However, for down-level releases to access an NFS server site attribute file in which any of the new NFS server site attributes is specified, or for a down-level release to access a mount handle database written by this release, co-existence APAR OA25864 must be installed on the down-level release.

11.7.2 NFS server mount symlink support

The NFS Server Mount Symlink Support line item removes the symlink restriction from NFSv4 mounts. Removal of that restriction requires the introduction of several new NFS server site attributes. Because the z/OS NFS server only accepts defined site attributes and will fail in its startup processing if it encounters any “undefined” attributes, it is necessary to install the coexistence APAR OA25864 for down-level z/OS NFS server releases so that they can tolerate, and ignore, the new site attributes. Installing this APAR allows the customer to roll back to an older supported release and use the new release's site attribute file.

Because the use of a prefix to identify z/OS UNIX file systems is now optional based on the new site attribute settings, the prefix has been removed from the z/OS UNIX file system entries in the Mount Handle Database (MHDB). To facilitate compatibility with MHDBs created by prior releases, entries that include a prefix can still be processed by this release. However, for down-level releases to be able to properly interpret the entries created by this release, it is necessary to install this coexistence APAR. This APAR adds code to add the prefix to any z/OS UNIX entries that do not have a prefix.

The above two coexistence items are in the same coexistence APAR OA25864.

Note: When you use the new function in the, for example, exports file or mount statements in test scripts, it is necessary to remove any exploitation of the new function before rolling back to a prior release. This action cannot be handled by coexistence APAR OA25864.

11.8 NFSv4 hints and tips

In this section, we provide useful hints and tips about NFSv4.

11.9 The mvsmnt mount processing attribute

With z/OS V1.9, the NFS client issues the mount using NFSv4 as the default. Therefore, just change the mount using NFSv2 when NFSv4 fails and NFSv4 is not needed.

When using NFSv4, no mount action is sent to the z/OS NFS server, as was done for NFSv2. Instead, a sequence of lookup requests are sent within one or multiple compound requests.

The z/OS NFS server processing attribute mvsmnt needs to be added when using NFSv4 to display this mount. You can add this attribute by running the following z/OS NFS server command:

```
/f NFSSprocname,list=mounts
```

NFSv4 mount points that were established without specifying the mvsmnt processing attribute are not recorded in the MHDB. The NFS client attempts to re-establish these mount points when it receives a NFS4ERR_FHEXPIRED response from the NFS server.

Note: To re-establish the NFSv4 mount points after a failover or stop and start the z/OS NFS server, specify the mvsmnt processing attribute.

11.10 SAP admin user request for root access during an installation

This section contains an example of when an SAP admin user runs an SAP installation connected to a z/OS NFS server using NFSv4.

The SAP admin user that requests root access during installation while the files are created should have UID0 and GID0.

11.10.1 NFSv4 and groupname

The GID(0) groupname of the appropriate remote z/OS directory where the NFS client is connected needs to be the same GID(0) groupname that is on the NFS client system.

The groupname can be displayed by running the ID command on both z/OS and the NFS client.

Displaying groupnames

To display the groupname of root user on z/OS, run the following command:

```
/etc:==>id
UID=0(ADAMM) GID=0(rootgrp) groups=3(SYS1)
```

To display the groupname of a root user on Linux on System z, run the following command:

```
ihlscob2:ß # id
UID=0(root) GID=0(root) groups=0(root)
```

Each command shows a different GID(0) groupname. The USERIDALIASTABLE option on z/OS can be used to assign or map a groupname as an alias to the z/OS groupname.

11.10.2 Displaying OMVS options

The following command displays the path and name of the USERIDALIASTABLE:

```
/D omvs,options
```

An example of the OMVS options are shown in Example 11-9.

Example 11-9 Output of the OMVS options

```
BPXO043I 04.23.04 DISPLAY OMVS 623
OMVS     000F ACTIVE             OMVS=(NF)
CURRENT UNIX CONFIGURATION SETTINGS:
MAXPROCSYS      =       4000    MAXPROCUSER     =        500
MAXFILEPROC     =      65535    MAXFILESIZE     =   11000000
MAXCPUTIME      =       1000    MAXUIDS         =        400
```

```
MAXPTYS          =        256
MAXMMAPAREA      =       4096    MAXASSIZE       =   41943040
MAXTHREADS       =      10000    MAXTHREADTASKS  =       5000
MAXCORESIZE      =    4194304    MAXSHAREPAGES   =     131072
IPCMSGQBYTES     =     262144    IPCMSGQMNUM     =      10000
IPCMSGNIDS       =        500    IPCSEMNIDS      =        500
IPCSEMNOPS       =         25    IPCSEMNSEMS     =         25
IPCSHMMPAGES     =        256    IPCSHMNIDS      =        500
IPCSHMNSEGS      =         10    IPCSHMSPAGES    =     262144
SUPERUSER        = BPXROOT       FORKCOPY        = COW
USERIDALIASTABLE= /etc/tablename
PRIORITYPG VALUES: NONE
PRIORITYGOAL VALUES: NONE
MAXQUEUEDSIGS    =       1000    SHRLIBRGNSIZE   =   67108864
SHRLIBMAXPAGES   =       4096    VERSION         = /
SYSCALL COUNTS   = NO            TTYGROUP        = TTY
SYSPLEX          = NO            BRLM SERVER     = N/A
LIMMSG           = ALL           AUTOCVT         = OFF
RESOLVER PROC    = RESOLVER
AUTHPGMLIST      = NONE
SWA              = BELOW
```

Add the Linux on System z GID(0) groupname root as an alias to the groupname of the z/OS GID(0) groupname rootgrp and browse /etc/tablename for verification. Example 11-10 shows an output example for /etc/tablename.

Example 11-10 Example of /etc/tablename

```
userids
KLKLEIN  klklein
JOEDOE   Joe_Doe
:groups
ROOTGRP  root
```

Activating the new or changed USERIDALIASTABLE parameter

The USERIDALIASTABLE parameter can be set as a static value in the PARMLIB data set by performing the following action:

```
BROWSE    SYS1.PARMLIB(BPXPRM00)
*********************************
  MAXPROCSYS(4000)
  MAXPROCUSER(500)
  MAXUIDS(400)
USERIDALIASTABLE('/etc/tablename')
```

The USERIDALIASTABLE parameter can be set dynamically by running the following command:

```
/SETOMVS USERIDALIASTABLE='/etc/tablename'
```

Displaying groupnames

To display the groupname of the z/OS ROOT user again, run the following command:

```
/:==>id
UID=0(ADAMM) GID=0(root) groups=3(SYS1)
```

The display now shows that the GID(0) groupname is the same.

NFS domain name on the NFS client and z/OS server

The NFS domain name needs to be the same on the NFS client and z/OS server. The domain name on z/OS can be displayed by running the following **nslookup** command, which produces the output that follows:

```
tso nslookup 9.155.3.96
Server:  deibp9eh1--31ndcr.wan.ibm.com
Address:  9.64.163.21
Name:    mcevsfd.mainz.de.ibm.com         domain name
Address:  9.155.3.96
```

Note: The **nslookup** command is issued through the DNS server. A wrong domain name will be shown when the DNS is not configured correctly. Therefore, use the **hostname** command in the z/OS UNIX shell instead to be certain of your information:

```
/u/hoerner:==>hostname
MCEVSF.mainz.de.ibm.com
```

Displaying active domains

To display the active domain on AIX as the NFS client, run the following command:

```
[0]:root@t18:/usr/hoer # chnfsdom
Current local domain: mainz.de.ibm.com
```

To display the active domain on a Linux on System z NFS client, run the following command:

```
ihlscob2:ß # hostname -f
ihlscob2§boeblingen.de.ibm.com

ihlscob2:ß # hostname -d
boeblingen.de.ibm.com
```

Attention: A potential problem could be that, in the `/etc/idmapd.conf` file, the DOMAIN still points to “localdomain”(default) and not to, for example, “mainz.de.ibm.com”, which is the case on z/OS. After setting DOMAIN to the correct value and restarting the NFS client (so idmapd rereads the configuration), ID mapping works as expected, and you can now see UID0 and GID0 for the newly created file on Linux on System z.

Linux on System z and the idmap daemon

On Linux on System z, the idmap daemon idmapd must be active. Check whether the idmapd is running by running the following command:

```
# ps -ef|grep idmap
root     21685 21449  0 16:29 pts/2    00:00:00 grep idmap
```

To enable idmapd permanently, run the following command:

```
chkconfig idmapd on
```

To debug idmapping, run the following command:

```
rcp idmapd -f -vvvvv
```

Mapping related messages are shown on the console.

Note: When using NFSv4, the Network Information Service (NIS) is helpful in setting up the heterogeneous environment.

11.10.3 Using the z/OS NFS client connected to a SUSE Linux server

This section shows the results of our test when issuing an NFSv4 mount to a SUSE Linux server.

We ran this test on a NFS server with Open SUSE 11.1 based on the 2.6.27.29-0.1-default kernel, and using the x86_64 nfs-kernel-server-1.1.3-18.2.1 architecture. We also ran this test on SLES 10, which is based on the 2.6.16.21-0.8-default kernel, and using the S/390x (IBM System z10) architecture.

The following export entry is defined on the SUSE Linux NFS server:

```
/home   *(fsid=0,root_squash,sync,no_subtree_check,insecure)
```

The following mount was issued using the path /home:

```
mount -t nfs -o "ip_address:/home" -f ip_address:/home /mount_point
```

This mount fails and shows the following message in the z/OS syslog:

```
BPXF028I FILE SYSTEM ip_address:/home WAS NOT MOUNTED.
RETURN CODE = 00000081, REASON CODE = 6E050002
```

The mount was successful when using a different path on the SUSE Linux NFS server:

```
mount -t nfs -o "ip_address:/" -f ip_address:/home /mount_point
```

Our test shows that when there are two or more different export entries defined, the first entry can be accessed but the second one cannot be accessed.

Explanation and solution

The Linux NFSv4 server exports the data somewhat differently than other NFS servers.

For NFSv2 or V3, when you mount to `linux:/home`, you go to `/home` on the Linux server, which is normal.

But for NFSv4, when you mount to `linux:/`, it automatically takes you to `/home` on the linux server, because the / in the mount command for NFSv4 to the Linux server is where the export starts.

In our test, `linux:/home` is used in the mount command. With NFSv4, you want to mount to `linux:/home/home`.

BIND function of Linux

Using the BIND function of Linux, it is possible to export more than one file system. The BIND is similar to a symlink in UNIX System Services, but it is not possible to use it as a mount option; you must define it in the `/etc/fstab` file of Linux as static or as a --bind statement, as shown in Example 11-11 on page 508.

An example of how to use the BIND function is shown in Example 11-11. All commands are performed as the root user on Linux on System z (SLES10) server.

Example 11-11 Using the Linux BIND function

```
On the Linux on System z Server:
# ls /tmp/
gconfd-a3808  gconfd-root  kde-a3808  kde-root  ksocket-root  orbit-a3808
orbit-root  xefnltmpjbRBHj  xefnltmpM2jWdU  xefnltmpN8iX8P  xefnltmpzYUbHk
# ls /home
a2553  a3383  a3662  a3745  a3783  a3808  test
# mkdir /nfsexports
# mkdir /nfsexports/home
# mkdir /nfsexports/tmp
# ls /nfsexports/home
# ls /nfsexports/tmp
# mount --bind /tmp/ /nfsexports/tmp
# ls /nfsexports/tmp
gconfd-a3808  gconfd-root  kde-a3808  kde-root  ksocket-root  orbit-a3808
orbit-root  xefnltmpjbRBHj  xefnltmpM2jWdU  xefnltmpN8iX8P  xefnltmpzYUbHk
# mount --bind /home/ /nfsexports/home
# ls /nfsexports/home
a2553  a3383  a3662  a3745  a3783  a3808  test
Define in the /etc/exports:
/nfsexports     *(fsid=0,root_squash,sync,no_subtree_check,insecure)
/nfsexports/home        *(ro,nohide,root_squash,sync,no_subtree_check,insecure)
/nfsexports/tmp *(rw,nohide,root_squash,sync,no_subtree_check,insecure)
TSO mounts:
mount FILESYSTEM(ip_address:/home) parm('ip_address:/home') MOUNTPOINT('/mnt')
type(nfs)
mount FILESYSTEM(ip_address:/tmp) parm('ip_address:/tmp') MOUNTPOINT('/mnt2')
type(nfs)
```

Attention: The mount --bind will be lost after a reboot. Therefore, it should be defined in the /etc/fstab:

```
/home     /nfsexports/home   none   rw,bind     0        0
/tmp      /nfsexports/tmp    none   rw,bind     0        0
```

Note: This BIND method does not work with SLES11, but other NFSv4 clients do not have a problem (for example, RHEL 5.3, AIX, or SUN). The Bugzilla report number for this problem is Bug 56666, which can be found at the following website:

https://bugzilla.linux.ibm.com

A

OAM implementation hints and tips

In this appendix, we describe information that we found useful in the implementation of OAM in our DFSMS V1.11 environment.

Check lists

Here are some items you should check:

- Ensure that the CBROAMxx parmlib member contains all OAM tape related options.
- Ensure that the OAM procedure points to the right CBROAMxx parmlib.
- Modify the ACS routines:
  - Follow the instructions in the section “Creating DB2 Databases for Object Tables and Directories” in Chapter 3, “Migrating, Installing, and Customizing OAM”, in *DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426.
  - You might need to add STEPLIB to the JCLs to make them work with DB2. Talk with your DB2 administrator. Usually, the libraries that you will need to add to your job, if required, are in the xxxxMSTR DB2 main task. If that is the case, your job will likely look like the one shown in Figure A-1.

```
//STEP1    EXEC PGM=IKJEFT01,REGION=4096K
//STEPLIB  DD DSN=DBxxx.SDSNEXIT,DISP=SHR
//         DD DISP=SHR,DSN=DBxxx.SDSNLOAD
//SYSPRINT DD   SYSOUT=*
//SYSTSPRT DD   SYSOUT=*
//SYSTSIN  DD   *
```

Figure A-1 Sample JCL to submit jobs

  - Ensure that the correct Storage Class/Management Class is assigned to objects for tape in environments, that is, either STORE, CHANGE, or CTRANS.
  - Update the ALLOC environment to assign a Storage Class if you are planning to use an IBM Tape Library.
  - Ensure that the Storage Group routine assigns a tape Storage Group to OAM tape allocations that you want use.

– Here are some Automatic Class Selections (ACSs) you can use to perform a quick test.

  • Here is a sample ACS routine to select an OAM Management Class (MGMTCLAS) (Figure A-2).

```
PROC MGMTCLAS
/*********************************************************************/
/*          ACS ROUTINE TO SELECT MANAGEMENT CLASS (MC)              */
/*********************************************************************/
/* HISTORY:                                                          */
/* SAMPLE TO YOU TRY IN YOUR TEST ENVIRONMENT.                       */
/* FIX IT TO FIT YOUR TEST.                                          */
/*********************************************************************/
/* FILTLIST DEFINITIONS                                              */
/*********************************************************************/

FILTLIST TOGRP1 INCLUDE(TOGRP1.**)         /* OAM PE         */
WRITE 'MC ACS GETS CONTROL &ACSENVIR=' &ACSENVIR           /* @02 */
/*********************************************************************/
/* OAM MANAGEMENT CLASS STUFF                                        */
/*********************************************************************/

   IF   (&DSN = &TOGRP1)
   THEN DO
    SELECT
     WHEN   (&ACSENVIR = 'STORE')
       DO
       WRITE 'STORE COLLECT NAME  = ('&DSN')  ='
       IF   (&STORCLAS = 'OBJDASD') THEN SET &MGMTCLAS = 'TOGRP1D1'
       IF   (&STORCLAS = 'OBJTAPE') THEN SET &MGMTCLAS = 'OBJTAPE'
       END
     WHEN   (&ACSENVIR = 'CTRANS')
       DO
       WRITE 'CTRANS COLLECT NAME  = ('&DSN')  ='
       IF   (&STORCLAS = 'OBJDASD') THEN SET &MGMTCLAS = 'TOGRP1T1'
       IF   (&STORCLAS = 'OBJTAPE') THEN SET &MGMTCLAS = 'OBJTAPE'
       IF   (&STORCLAS = 'OBJTAP2') THEN SET &MGMTCLAS = 'OBJTAPE'
       END
     OTHERWISE EXIT CODE(25)
    END
   END
END
```

Figure A-2 MGMTCLAS example

- Here is a sample ACS routine to select an OAM Storage Class (STORCLAS) (Figure A-3).

```
PROC STORCLAS
/*********************************************************************/
/*               ACS ROUTINE TO SELECT STORAGE CLASS (SC) */
/* HISTORY: */
/* SAMPLE TO YOU TRY IN YOUR TEST ENVIRONMENT. */
/* FIX IT TO FIT YOUR TEST. */
/*********************************************************************/
/* FILTLIST DEFINITIONS */
/*********************************************************************/

FILTLIST TOGRP1  INCLUDE(TOGRP1.**)              /* OAM PE  TL1     */
FILTLIST TOGRPL2 INCLUDE(TOGRP*.COL*.TL2*.**)    /* OAM PE  TL2     */
/*********************************************************************/
/* CLASS TRANSITION FOR OAM                                          */
/*********************************************************************/
 IF &ACSENVIR = 'CTRANS' THEN DO
  WRITE 'COLL    NAME  = ('&DSN')  ='
  SELECT
  WHEN (&DSN = &TOGRP1)
     SET &STORCLAS = 'OBJDASD'
/*   SET &STORCLAS = 'OBJTAPE'  */
  WHEN (&DSN = &TOGRPL2)
     SET &STORCLAS = 'OBJTAP2'
  OTHERWISE EXIT CODE(13)
  END
  WRITE 'SC      NAME  = ('&STORCLAS')  ='
 END
IF &ACSENVIR = 'STORE' THEN DO
  WRITE 'OBJECT  NAME  = ('&MEMN')  ='
  WRITE 'COLLECT NAME  = ('&DSN')  ='
  SELECT
    WHEN (&DSN = &TOGRP1)
     SET &STORCLAS = 'OBJDASD'
 /*  SET &STORCLAS = 'OBJTAPE' */
    OTHERWISE EXIT
  END
EXIT
END
END
```

Figure A-3 STORCLAS example

- Here is a sample ACS routine to select an OAM Storage Group (STORGRP) (Figure A-4).

```
PROC STORGRP
/*********************************************************************/
/*             ACS ROUTINE TO SELECT STORAGE GROUP (SG) */
/* HISTORY: */
/* SAMPLE TO YOU TRY IN YOUR TEST ENVIRONMENT. */
/* FIX IT TO FIT YOUR TEST. */
/*********************************************************************/
/* MAIN STORAGE GROUP SELECTION ROUTINE /
/*********************************************************************/

FILTLIST TOGRP1 INCLUDE(TOGRP1.**)

/* OAM  PE */
   SELECT
     WHEN (&DSN = &TOGRP1)
        DO
         IF (&STORCLAS = 'OBJDASD' OR &STORCLAS = OBJTAP*)
           THEN
             SET &STORGRP = 'TOGRP1'
           ELSE
             EXIT CODE(82)
        END
   END
END
```

Figure A-4 STORAGEGROUP example

ISMF used in our test

In Storage Group, we used the constructs shown in Figure A-5.

```
 Panel  List  Utilities  Scroll  Help
------------------------------------------------------------------------------
                              STORAGE GROUP LIST
Command ===>                                                  Scroll ===> CSR
                                                            Entries 68-74 of 74
                                                            View in Use
CDS Name : SYS1.SMS.SCDS

Enter Line Operators below:

     LINE       STORGRP  SG                OAM DEL  OAM RET
     OPERATOR   NAME     TYPE              PROTECT  PROTECT
    ---(1)----  --(2)--- -------(3)------  -(47)--  -(48)--
                TOGRP1   OBJECT            ENABLE   DISABLE
                TOGRP1B1 OBJECT BACKUP     -------  -------
```

Figure A-5 Storage Groups used to test.

Internally, TOGRP1 OBJECT has the structure shown in Figure A-6.

```
 OBJECT STORAGE GROUP ALTER            Page 1 of 2
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : TOGRP1

To ALTER Storage Group, Specify:
  Description ==> OBJECT STORAGE GROUP FOR TOGRP1
              ==>

  Qualifier  . . . . . TOGRP1    (1 to 8 character qualifier)
  Cycle Start Time . . NONE  (0-23 or NONE)  End Time . .     (0-23 or blank)
  OSMC Processing System . . . . SC70      (? for list of OSMC System names)

  Library Names   (1 to 8 Characters each):
   ===>           ===>           ===>           ===>
   ===>           ===>           ===>           ===>

   ALTER     SMS Storage Group Status . . N  (DEFINE - Y, ALTER - Y or N)
                        OBJECT STORAGE GROUP ALTER            Page 2 of 2
 Command ===>

 SCDS Name . . . . . : SYS1.SMS.SCDS
 Storage Group Name  : TOGRP1

 To ALTER Storage Group, Specify:
   Volume Full Threshold  . . . .       (0-9999)
   Drive Start Threshold  . . . .       (0-9999)
   Volume Full at Write Error . .       (Y or N)

   OAM Deletion Protection  . . . Y     (Y=Enable or N=Disable)
   OAM Retention Protection . . . N     (Y=Enable or N=Disable)
```

Figure A-6 Structure of OBJECT STORAGE GROUP TOGRP1

TOGRP1B1 OBJECT BACKUP has the structure shown in Figure A-7.

```
 OBJECT BACKUP STORAGE GROUP ALTER
Command ===>

SCDS Name . . . . . : SYS1.SMS.SCDS
Storage Group Name  : TOGRP1B1

To ALTER Storage Group, Specify:
  Description ==> OBJECT BACKUPTOGRP1
              ==>
  Cycle Start Time . . NONE  (0-23 or NONE)  End Time . .     (0-23 or blank)
  OSMC Processing System . . . . SC70      (? for list of OSMC System names)

  Library Names   (1 to 8 Characters each):
   ===>           ===>           ===>           ===>
   ===>           ===>           ===>           ===>
  Volume Full Threshold  . . . .       (0-9999)
  Drive Start Threshold  . . . .       (0-9999)
  Volume Full at Write Error . .       (Y or N)
   ALTER     SMS Storage Group Status . . N  (DEFINE - Y, ALTER - Y or N)
```

Figure A-7 Structure of OBJECT BACKUP STORAGE GROUP TOGRP1B1

B

APARs to be reviewed

In this appendix, we provide the text of APARs that have been mentioned in the book and APARs that should be reviewed.

APAR II13752

This APAR describes the Tracking Facility, its output, how to send the output to IBM for analysis, and how to pull the current parmlib member to aid in reducing the amount of data the Tracker collects.

The Tracking Facility has been included with the BCP element of z/OS in the Console Availability Feature (JBB7727) and has been included in all following BCP elements of z/OS releases.

It has several purposes:

- Identify products that continue to use functions that will not be supported in future z/OS releases.
- Identify products that would benefit from changing services or using different keywords on existing services.

Beginning with z/OS V1.6 (HBB7709), IBM began removing support for 1 byte Console IDs. Console Names or 4 byte Console IDs (introduced in MVS/ESA 4.1.0) must be used instead. This became the first usage of the Tracking Facility.

IBM testing organizations, ESP accounts, and customers should run with this facility active to help identify products that will need to be changed. The facilities ONWITHABEND option should *not* be specified in z/OS V1.9 or above because some users of the facility are not reporting problems. To learn more about the facility, see Appendix A, “Tracking Facility”, in *z/OS MVS Planning: Operations*, SA22-7601.

To activate or deactivate the facility, use the SETCON TRACKING command described in *z/OS MVS System Commands,* SA22-7627. After the facility is active, it records instances of the event. The DISPLAY OPDATA,TR command is used to display the recorded events by way of the CNZ1001I message (see *z/OS MVS Commands*, SA22-7627 and *z/OS MVS System Messages Volume 4 (CBD-DMO)*, SA22-7634).

Before contacting IBM support concerning Tracker questions, send an email to consoles@us.ibm.com. If you do need to contact IBM support, attempt to determine the appropriate IBM organization to contact. For example, if the Tracked entry indicates SMS, then DF/SMS support would be the best place to start.

To notify IBM about any recorded instances, send an email to consoles@us.ibm.com that provides the output produced by the DISPLAY OPDATA,TR command (CNZ1001I message). Include any information that would help identify which product or vendor caused the events to be tracked. IBM might notify the product owner or ask you to notify the vendor and request that updates be made.

Note that this is a staged evolution. The changes do not have to be made immediately, as existing code will continue to work as it does today.

The Tracking Facility supports an exclusion list that informs the facility which instances have already been reported and should no longer be tracked. This list is updated from the data sent to IBM. Current lists are available for download at the following address:

`http://www-1.ibm.com/servers/eserver/zseries/zos/downloads/`

This web page will include the date when the list was last updated. After IBM has updated the Exclusion List, download and install the latest list to help reduce duplicate notifications. When migrating to a new z/OS release, obtain the Exclusion List that is appropriate for that Release.

After the list has been downloaded, the “Special Processing” section should be reviewed and appropriate action(s) taken.

Place the list in the parmlib and use the SET CNIDTR command (see *z/OS MVS System Commands*, SA22-7627) to activate it. Although the SET CNIDTR command might be issued before the SETCON TRACKING command is, tracking might be activated and tracking instances might occur before the SET CNIDTR command has completely activated the exclusion list. Note that events that have been recorded before the activation of the new list will remain recorded and displayed in CNZ1001I, even if excluded in the new list. To remove the old instances, the facility must be deactivated and re-activated using the new Exclusion list.

IBM encourages you to update your copy of the Exclusion List for your organization’s specific applications. The list that can be downloaded contains details and examples of proper syntax for updating the list.

Here is an example of the Tracker report output (note that some columns were compressed to fit in this APAR description):

```
CNZ1001I 15.47.47 TRACKING DISPLAY
STATUS=ON       NUM=2    MAX=1000 MEM=n/a EXCL=0     REJECT=0
--TRACKING INFORMATION-- -VAL- JOBNAME  PROGNAME+OFF-- ASID NUM
WTO: IEC350I CATALOG ADD    00 CATALOG  IGGOCLX0 80BAC   1A   1
WTO: IEF677I WARNING MES    00 JES2     IEFNB903  C9AA   13   1
------------------------------------------------------------------
TO REPORT THESE INSTANCES, SEND THIS MESSAGE VIA E-MAIL TO
CONSOLES@US.IBM.COM. FOR ADDITIONAL INFORMATION OR TO OBTAIN A
CURRENT EXCLUSION LIST, SEE APAR II13752.
```

APAR II14458

APAR II14458 explains that the INDEX CISIZE VSAM data set might change when defined or redefined on z/OS V1.10.

When a VSAM data set or ICF catalog is defined or redefined on z/OS V1.10, the cisize index might be increased. This can occur if the data CA size was in tracks and the amount specified was not 1, 3, 5, 7, 9, or 15 tracks. For example, if the space requested was TRK(2,2), the allocation amount will be TRK(3,3) on z/OS V1.10. Because the data CA size is now larger, the minimum cisize index might increase. For those applications such as IMS or CICS that have or might have static LSR pool definitions, a change to the cisize index might cause a data set to no longer open after being redefined.

There is a new tool that is now available to discover those data sets that will have a larger CA size. You can obtain this tool at the following address:

`ftp://public.dhe.ibm.com`

Look in the `servers/storage/support/software/dfsms/ as INDXCI10.JCL.CNTL.TRSD` directory. This tool will need to be downloaded in binary mode to a data set with the following attributes:

```
LRECL=1024,BLKSIZE=6144,RECFM=FB,DSORG=PS
```

After downloading the tool, you will need expand it using either AMATERSE or TRSMAIN. This tool consists of REXX exec and a JCL to execute in batch. Review the comments in the JCL for more information about the tool.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this book.

IBM Redbooks

For information about ordering these publications, see “How to get Redbooks” on page 522. Note that some of the documents referenced here may be available in softcopy only.

- *z/OS V1R7 DFSMS Technical Update*, SG24-7225
- *z/OS Version 1 Release 10 Implementation*, SG24-7605
- *z/OS Version 1 Release 11 Implementation*, SG24-7729

Other publications

These publications are also relevant as further information sources:

- *DFSMS Access Method Service for Catalogs,* SC26-7374
- *DFSMS Managing Catalogs,* SC26-7409
- *DFSMSrmm Managing and Using Removable Media*, SC26-7404
- *High Level Assembler Programmer's Guide*, SC26-4941
- *IBM Health Checker for z/OS User’s Guide,* SA22-7994
- *IBM System Storage VTF Mainframe Release Notes*, GC53-1189
- *ISS Network Authentication Service Administration*, SC24-5926
- *MVS Device Validation Support*, SA22-7586
- *Network File System Guide and Reference*, SC26-7417
- *UNIX System Services Planning*, GA22-7800
- *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS DFSMSdfp Advanced Services*, SC26-7400
- *z/OS DFSMSdfp Utilities*, SC26-7414
- *z/OS DFSMS Using Magnetic Tapes*, SC26-7412
- *z/OS IBM CS V2R10.0: IP Configuration Guide*, SC31-8725
- *z/OS Integrated Security Services Network Authentication Service Administration,* SC24-5926
- *z/OS MVS JCL Reference*, SA22-7597
- *z/OS MVS System Messages, Volume 7 (IEB-IEE)*, SA22-7637
- *z/OS Support for Unicode: Using Conversion Services*, SC33-7050
- *z/OS UNIX System Services Planning*, SC28-1890
- *z/OS V1.11 DFSMSdfp Advanced Services,* SC26-7400

- *z/OS V1.11 DFSMSrmm Implementation and Customization Guide*, SC26-7405
- *z/OS V1R11 DFSMS Storage Administration Reference*, SC26-7402
- *z/OS V1R11 DFSMS Using Data Sets*, SC26-7410
- *z/OS V1R11 JCL Reference*, SA22-7597
- *z/OS V1R11 Migration*, GA22-7499
- *z/OS DFSMS Managing Catalogs*, SC26-7409

Online resources

These websites are also relevant as further information sources:

- Linux problems

  https://bugzilla.linux.ibm.com/
- Kerberos Network Authentication Service (V5)

  http://www.faqs.org/rfcs/rfc1510.html
- Kerberos V5 GSS

  http://www.faqs.org/rfcs/rfc4121.html/
- Migration Assistant Tracker exclusion list

  http://www-03.ibm.com/servers/eserver/zseries/zos/downloads/
- NFSv2 Request for Comments

  http://www.faqs.org/rfcs/rfc1813.html
- NFSv4 Request for Comments

  http://www.faqs.org/rfcs/rfc3010.html

How to get Redbooks

You can search for, view, or download Redbooks, Redpapers, Technotes, draft publications and Additional materials, as well as order hardcopy Redbooks publications, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

X

Z

Redbooks

z/OS V1.11 DFSMS Release Guide

(1.0” spine)
0.875”<->1.498”
460 <-> 788 pages

z/OS V1.11 DFSMS Release Guide

Understand the features and functions of z/OS DFSMS V1.11

Contains implementation hints and tips

Provides functional, hands-on code samples

Each release of DFSMS builds upon the previous version to provide enhanced storage management, data access, device support, program management, and distributed data access for the z/OS platform in a system-managed storage environment.

This IBM Redbooks publication provides a summary of the functions and enhancements in z/OS V1R11 DFSMS. It provides you with the information that you need to understand and evaluate the content of this DFSMS release, along with practical implementation hints and tips. Also included are enhancements that were made available through an enabling PTF that has been integrated into z/OS DFSMS V1R11.

This book was written for storage professionals and system programmers who have experience with the components of DFSMS. It provides sufficient information so that you can start prioritizing the implementation of new functions and evaluating their applicability in your DFSMS environment.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-7768-00 ISBN 0738434418