IBM

z/OS V1.12 DFSMS Technical Update

- Introduces features and functions of z/OS V1.12 DFSMS
- Includes implementation hints and tips
- Provides working code and JCL samples

Mary Lovelace
Werner Bauer
Juliana Eneas
Michele Lanthier
Norbert Schlumberger
Jeanne Vangsness
Gerhard Weisshaar

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/OS V1.12 DFSMS Technical Update

September 2011

SG24-7895-00

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

First Edition (September 2011)

This edition applies to z/OS Version 1, Release 12 DFSMS (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

Preface

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 V1R12 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 enabling PTFs that have been integrated into z/OS DFSMS V1R12.

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 Redbooks publications about z/OS storage products, IBM Tivoli® Storage Productivity Center, Tivoli Storage Manager, and Scale Out NAS.

Werner Bauer is a Certified Consulting IT Specialist in Germany. He has 29 years of experience in storage software and hardware as well as with IBM S/390® and IBM z/OS. His areas of expertise include disaster recovery solutions based on IBM enterprise disk storage subsystems. Werner is a frequent speaker at storage conferences and European GUIDE and SHARE meetings. He has also written extensively in various DS8000® Redbooks publications. He holds a degree in Economics from the University of Heidelberg and a degree in Mechanical Engineering from FH Heilbronn.

Juliana Eneas is an IT MVS™ Storage Specialist in IBM Brazil. She has 10 years of experience with IBM mainframes. She has worked at IBM in z/OS Storage Support since 2004. Prior to joining IBM she worked as a Capacity Planner for a Brazilian bank. She holds a specialist degree in IT Project Management from University of São Paulo, Brazil. Her areas of expertise include DFSMS base and outsourcing MVS Storage projects.

Michele Lanthier is a Remote Technical Specialist working for IBM Canada in Global Technology Services. She provides support for z/OS DFSMS storage-related questions and problems to IBM clients from Canada, China, and the United States. Michele has more than 15 years of experience with large systems and has worked for IBM Canada for ten of those years.

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: Enhnacning DFSMSrmm” to copy data sets residing on tape from one media to another media including the update of the ICF usercatalog and DFSMSrmm after all data is successfully copied. Norbert also has experience in installing and using IBM System Storage® VTF® Mainframe. He has worked at IBM for 37 years.

Jeanne Vangsness is a DFSMS Software Technical Support Engineer specializing on the DFSMShsm product area. She has been with IBM for 26 years and has spent 19 of those years working with customers and developers on the DFSMS product set. She provides feedback to the DFSMShsm development team in a number of areas to help make improvements to the functionality of DFSMShsm. Jeannie has co authored several IBM Redbooks publications, including *Implementing an Open SAN,* SG24-6116 and *DFSMShsm Fast Replication,* SG24-7069 and has provided assistance in other IBM Redbooks publications.

Gerhard Weisshaar is a Senior Consultant working with ATICSTRO AG Training and Consulting in Germany. He has more than 20 years of experience in z/OS and its predecessors. His areas of expertise include implementation of storage and storage networking software and hardware, as well as z/OS Security Server.

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

Rich Conway
Bob Haimowitz
Emma Jacobs
International Technical Support Organization, San Jose Center

Stephen Branch
Brian Corkill
Rohan Kurane
Elaine Lai
Larry Law
Kim Nguyen
Glenn Wilcock
DFSMS development and support

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

z/OS V1.12 DFSMS release overview

This chapter provides a summary of the line items in DFSMS V1.12 by component and provides an overview of the contents in the remainder of the book.

1.1 DFSMS catalog

There are two major enhancements in V1.12 with respect to where you can place your catalogs and the maximum size that they can attain. Along with these two major enhancements the following new functions are provided:

- Additional ways to identify certain catalog service tasks waiting on the input/output table resources (SYSZTIOT).
- Correcting catalog entries for volumes are now ordered by RBA starting from 0 for data and index components regardless of their order in the DEFINE VOLUMES parameter.
- Expanded Partial Release to include data sets which span multiple volumes.
- Indirect volser support for zFS data sets.

1.1.1 Extended addressable catalogs

In z/OS V1R12 DFSMS there is new support for catalogs with extended addressability (EA). This makes it possible to define and use integrated catalog facility (ICF) basic catalog structures (BCS) with extended addressability (EA), allowing catalogs larger than 4 GB.

1.1.2 Contention checking

In z/OS V1R12, the catalog address space (CAS) is now designed to check for contention periodically. Based on an interval you specify and the reason for contention, CAS now writes a logrec record and terminates certain tasks that have waited for longer than the specified maximum when contention checking occurs. A new MODIFY CATALOG,CONTENTION command allows you to specify a different interval than the 30 minute default, or to disable CAS contention detection. This new function is intended to prevent tasks that take excessive time to complete, or that never complete, from affecting catalog performance.

1.2 IDCAMS enhancements

There are two main improvements related to IDCAMS for this new z/OS release. The first enhancement offers a performance increase by reducing unnecessary recalls for deletion of GDGs HSM migrated.

Another improvement in this z/OS version is the addition of DCOLLECT new attributes. DCOLLECT reflects the information about the attributes of OAM Protection and Job-name for EAV data sets that were introduced by DFSMS V1.11 and also reflects all the attributes that are supported by SMS Data Class.

1.3 VSAM and VSAM RLS enhancements

In this section we provide an overview of the VSAM and VSAM RLS enhancements in DFSMS V1.12.

1.3.1 Control area reclaim

Over time, VSAM key-sequenced data sets (KSDS) for which records are added and deleted have often become fragmented and have a significant number of empty Control Areas (CAs) that consume DASD space, increase the size of the indexes, and reduce performance.

In z/OS V1.12, DFSMSdfp allows you to specify that VSAM dynamically reclaim empty control areas for KSDSs, including catalogs and those used for record-level sharing (RLS), and reclaim the associated index records as needed. This new function helps you preserve performance, minimize space utilization for KSDSs, and improve application availability, and allows you to avoid the need to reorganize most KSDSs.

1.3.2 DFSMS disk striping

In z/OS 1R12, VSAM record level sharing (RLS) supports striped data sets. This brings you the benefits of VSAM striping, such as allowing single application requests for records in multiple tracks or control intervals (CIs) to be satisfied by concurrent I/O requests to multiple volumes. Using striped data sets can result in improved performance by enabling the transfer of data at rates greater than can be achieved using single I/O paths.

1.3.3 Extended addressable catalogs

DFSMS V1.12 has new support for catalogs with extended addressability (EA). This makes it possible to define and use integrated catalog facility (ICF) basic catalog structures (BCS) with extended addressability (EA), allowing catalogs larger than 4 GB.

1.4 DFSMSdss enhancements

In this section we provide a summary of the DFSMSdss enhancements in DFSMS V1.12.

1.4.1 Using Basic Sequential Access Method I/O

Starting with z/OS V1R12, DFSMSdss now uses BSAM instead of EXCP to read from and write to DFSMSdss dump data sets during DUMP, COPYDUMP, and RESTORE operations. DFSMSdss is designed to use larger blocks when possible for DUMP, COPYDUMP, and RESTORE operations, and to support Extended Format Sequential dump data sets on DASD for DUMP, RESTORE, and COPYDUMP. The use of larger block sizes improves performance for these operations, and using Extended Format dump data sets supports striping and compression. This allows DFSMSdss to support 256 K blocks when writing to and reading from a tape. Before z/OS V1R12, the maximum was 65,520 bytes.

1.4.2 Removal of IMBED, REPLICATE and KEYRANGE

The creation of new VSAM data sets with IMBED, REPLICATE and KEYRANGE attributes has been unsupported since z/OS V1.6. In z/OS V1.12, the system is designed to remove these attributes automatically from VSAM data sets logically dumped using DFSMSdss.

1.5 System Managed Storage enhancements

In this section we provide a summary of the SMS enhancements in DFSMS V1.12.

1.5.1 Health Check function

z/OS V1.12 enhances the z/OS Health Check function with two new Health Checks that are specific to the SMS subsystem. These new features check two best practices for CDS allocation; performance and availability are improved when the suggested settings are implemented.

1.5.2 Storage Group copy function

DFSMS V1.12 also provides an update for the Storage Group copy function, allowing the volume definitions to be copied across the new Storage Group being defined. This enhancement provides the ability to easily copy Storage Groups.

1.5.3 Trace parameters

VOLSELMSG and Trace parameters were modified to allow individual settings in z/OS 1.12. This allows different trigger criteria to be set for each of them.

1.6 PDSE

With DSMS V1.12 there are two new features for PDSEs: the ability to delete all members with the IDCAMS DELETE command and the ability to detect corrupted PDSEs in the LNKLST.

1.6.1 IDCAMS DELETE PDS and PDSE

This new feature allows the user to empty a partitioned data set with a single IDCAMS DELETE command, thus deleting all members. This removes the restriction of deleting only one member at a time. This should save administrative work in managing data sets.

1.6.2 Identify corrupt PDSE in LNKLST

This enhancement allows the user to attempt to restore the corrupt PDSE, and re-IPL the system if necessary, without taking a stand-alone dump to debug the problem.

1.7 DFSMShsm enhancements

In this section we provide a summary of the DFSMShsm enhancements in DFSMS V1.12.

1.7.1 Fast reverse restore

DFSMShsm supports fast reverse restore of a copy pool in both COPY and NOCOPY modes. However, fast reverse restore cannot be used in combination with the preserve mirror function.

1.7.2 Dump stacking

Dump stacking has always been limited to 99 volumes. This limit is now increased to 255 volumes. Tapes can be used more efficiently with stacking of up to 255 volumes on one single dump tape.

1.7.3 Multi-task recovery from dump

Dump volume recovery is now multi-tasked. A user can now instruct dump recovery to run 64 concurrent dump recovery commands at one time.

1.7.4 Define cross memory invocation function

In DFSMS V1.12, additional parameters were added to the DSSXMMODE command so that users can specify which functions will invoke DFDSS in cross memory mode and which will not.

1.7.5 DFSMShsm Fast Replication enhancements

These enhancements allow DFSMShsm to fully exploit the latest FlashCopy® features of the IBM System Storage DS8000 and complete the Phase 2 implementation.

1.8 DFSMSrmm

In this section we provide a summary of the DFSMSrmm enhancements in DFSMS V1.12.

1.8.1 DFSMSrmm EAV and IPv6 support

In z/OS V1R12, new support makes all data sets used by DFSMSrmm eligible for allocation in the extended addressing space of an EAV. This includes the DFSMSrmm journal and dynamically allocated temporary files. DFSMSrmm support for IPv6 is also implemented.

1.8.2 DFSMSrmm simplified monitoring and management

In z/OS V1R12, DFSMSrmm provides that when a retention limit is reached, it will be added to the ACTIVITY file. New reports created from the ACTIVITY and extract files can help you see why retention limits were triggered. Also, OPENRULE ignore processing is available for duplicate tape volumes, and DFSMS recovery from multivolume tape label anomalies is based on policy information you defined to DFSMSrmm.

This means you can exploit OPENRULE ignore processing for duplicate tape volumes. DFSMS automatically handles multi-volume label anomalies as described by messages

IEC709I, IEC710I, IEC711I and IEC712I. This enables automated recovery based on information defined to DFSMSrmm.

1.8.3 DFSMSrmm RAS enhancements

In z/OS 1R12, DFSMSrmm creates additional trace records for processing outside the subsystem address space to enable improved diagnostics.

1.8.4 DFSMSrmm hardware support

In z/OS 1R12, DFSMSrmm supports copies of logical tape volumes that are exported from virtual tape subsystems (VTS) such as the TS7700 virtualization engine, including copies that might be sent to a secure offsite storage facility for disaster recovery. This new function helps you manage exported copies and their locations in addition to managing original virtual tape volumes residing in a VTS.

1.8.5 DFSMSrmm competitive enhancements

In z/OS 1R12, DFSMSrmm allows you to define policy rules that govern whether residual data on tape volumes should be erased during close processing. This new function to erase unused data helps you ensure that tapes sent outside your installation contain only the data you wish to send.

1.9 Object Access Method

In this section we provide a summary of the OAM enhancement in DFSMS V1.12.

1.9.1 OAM RAS enhancements

In z/OS V1.12, OAM provides API support for the object storage and retrieval function (OSR) to run in a CICS® threadsafe environment. This is intended to allow exploiters to take advantage of the improved multitasking and throughput capabilities provided by threadsafe programming. Additionally, the volume recovery utility improves performance when recovering object data stored on optical and tape media.

2

Catalog support

This chapter describes the catalog enhancements provided in DFSMS V1.12.

2.1 DFSMS catalogs

A catalog is a data set that contains information about other data sets. It provides users with the ability to locate a data set by name, without knowing the volume where the data set resides. Thus, data sets can be moved from one device to another, without requiring a change in JCL DD statements that refer to an existing data set.

The basic catalog structure (BCS) is a VSAM key-sequenced data set. It uses the data set name of entries to store and retrieve data set information. For VSAM data sets, the BCS contains volume, security, ownership, and association information. For non-VSAM data sets, the BCS contains volume, ownership, and association information. When we talk about a catalog, we usually mean the BCS. The VSAM Volume Data Set (VVDS) can be considered an extension of the volume table of contents (VTOC). The VVDS is volume-specific, whereas the complexity of the BCS depends on your definitions. The relationship between the BCS and the VVDS is many-to-many. That is, a BCS can point to multiple VVDSs and a VVDS can point to multiple BCSs.

There are two major enhancements in V1.12 with regard to where you can place your catalogs and the size that they can become. Along with these two major enhancements the following function is also provided:

- There are other ways to identify certain catalog service tasks waiting on the input/output table resources (SYSZTIOT).
- Correcting catalog entries for volumes are now ordered by RBA starting from 0 for data and index components regardless of the order they are in the DEFINE VOLUMES parameter.
- Partial Release is expanded to include data sets which span multiple volumes.
- There is indirect volser support for zFS data sets.

2.1.1 Larger size catalogs

Prior to DFSMS V1.12 the maximum size of catalogs is limited to 4 GB, which affects scalability. As volume sizes have increased, the need for larger catalogs has become apparent. The solution is to use the current extended addressability function in VSAM. This function requires the data sets to be SMS managed with the extended format (EF) and extended addressability characteristics.

Note: This BCS catalog enhancement does not apply to VVDS data sets, which continue to have a four-gigabyte size limit.

For extended addressable (EA) catalogs, the z/OS V1.12 enhancement allows catalogs to grow beyond the previous 4 GB size limit, using extended addressability. Extended addressability support already exists for VSAM data sets; this new support lets you define BCS catalogs as extended format data sets to give them extended addressability.

Restriction: Unlike other extended format VSAM data sets, which can also be striped or compressed, extended format ICF BCSs cannot be striped or compressed.

Requirements

Several requirements must be met to DEFINE EA catalogs. They must be SMS managed, meaning that you specify your data set requirements by using a Data Class, a Storage Class, and a Management Class.

For more information on SMS refer to “Using the Storage Management Subsystem” in the manual *z/OS DFSMS Using Data Sets*, SC26-7410, available from:

http://www-03.ibm.com/systems/z/os/zos/bkserv/r12pdf/

The Data Class is required to assign the extended attributes (see ❶ on Figure 2-1). They must be defined as Extended Format data sets. Choose option 4 in the command line on the ISMF PRIMARY OPTION MENU and press Enter to continue.

```
          ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R12
Selection or Command ===>

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

Figure 2-1 Main ISMF panel, highlighting option 4

The Data Class Application Selection panel is displayed next, as shown in Figure 2-2. Select option 1 to generate a list of available Data Classes in your installation. Note that if you need to modify one Data Class, replace `CDS Name .... 'ACTIVE'` with the Source Control Data Set (SCDS) name.

```
                        DATA CLASS APPLICATION SELECTION
 Command ===>

 To perform Data Class Operations, Specify:
   CDS Name . . . . . . 'ACTIVE'
                                (1 to 44 character data set name or 'Active' )
   Data Class Name  . . *         (For Data Class List, fully or partially
                                  specified or * for all)

 Select one of the following options  :
   1  1. List    - Generate a list of Data Classes
      2. Display - Display a Data Class
      3. Define  - Define a Data Class
      4. Alter   - Alter a Data Class

 If List Option is chosen,
    Enter "/" to select option      Respecify View Criteria
                                    Respecify Sort Criteria
Use ENTER to Perform Selection;
 Use HELP Command for Help; Use END Command to Exit.
```

Figure 2-2 Data Class application selection

Issue the command `D SMS` to determine the SCDS name (Figure 2-3).

```
D SMS
IGD002I 10:28:14 DISPLAY SMS 273
SCDS = SYS1.SMS.SCDS ⇦
ACDS = SYS1.SMS.ACDS
COMMDS = SYS1.SMS.COMMDS
DINTERVAL = 150
REVERIFY = NO
ACSDEFAULTS = NO
```

Figure 2-3 D SMS command

The mechanism for requesting Extended Format for VSAM data sets is to use the SMS data class parameter `Data Set Name Type` value EXTENDED and the additional subparameter `If Extended` set to REQUIRED or PREFERRED. To allow the extended format data sets to become larger than 4 GB, you also need to set the data class parameter `Extended Addressability` value to YES (see ❷ on Figure 2-4 on page 11).

```
                                DATA CLASS ALTER                Page 2 of 5
 Command ===>

 SCDS Name . . . : SYS1.SMS.SCDS
 Data Class Name : EAVEXT

 To ALTER Data Class, Specify:

   Data Set Name Type  . . . . . EXT    (EXT, HFS, LIB, PDS, Large or blank)
❷   If Ext  . . . . . . . . . . . P      (P=Preferred, R=Required or blank)
     Extended Addressability . . Y      (Y or N)
     Record Access Bias  . . . . U      (S=System, U=User or blank)
   Space Constraint Relief . . . N      (Y or N)
     Reduce Space Up To (%)  . .        (0 to 99 or blank)
     Dynamic Volume Count  . . .        (1 to 59 or blank)
   Compaction  . . . . . . . . .        (Y, N, T, G or blank)
   Spanned / Nonspanned  . . . .        (S=Spanned, N=Nonspanned or blank)
   System Managed Buffering  . .             (1K to 2048M or blank)
   System Determined Blocksize   Y      (Y or N)
   EATTR . . . . . . . . . . . . . O    (O=Opt, N=No or blank)
 Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
 Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 2-4 Data Class Alter panel

2.1.2 EAS catalogs

Although the ICF BCS is a VSAM data set, catalogs were restricted by the system from residing in extended addressable space (EAS) on an extended addressable volume (EAV) in z/OS V1R10 and V1R11 without the coexistence of APAR OA31449. In z/OS V1R12 this restriction has been removed to allow Catalogs definitions in the extendable addressable space.

Defining a catalog in EAS

To enable a catalog to use EAS, you define the catalog using the EATTR(OPT) parameter, which can be specified either in your JCL (see Example 2-1) or in an SMS Data Class. The pertinent `Data Class Alter` panel is shown in Figure 2-5 on page 12.

Example 2-1 JCL example of EATTR parameter

```
//DEFCAT EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
   DEFINE USERCATALOG -
       (NAME(SYSA.MHLEAV3.UCAT) -
       SHAREOPTIONS(3,4) -
       ICFCATALOG -
       VOLUME(MLDE65) -
       CYL(10 10) -
       EATTR(OPT))
/*
```

```
                              DATA CLASS ALTER                 Page 2 of 5
Command ===>

SCDS Name . . . : SYS1.SMS.SCDS
Data Class Name : EAVEXT

To ALTER Data Class, Specify:

  Data Set Name Type  . . . . . EXT    (EXT, HFS, LIB, PDS, Large or blank)
    If Ext  . . . . . . . . . . P      (P=Preferred, R=Required or blank)
    Extended Addressability . . Y      (Y or N)
    Record Access Bias  . . . . U      (S=System, U=User or blank)
  Space Constraint Relief . . . N      (Y or N)
    Reduce Space Up To (%)  . .        (0 to 99 or blank)
    Dynamic Volume Count  . . .        (1 to 59 or blank)
  Compaction  . . . . . . . . .        (Y, N, T, G or blank)
  Spanned / Nonspanned  . . . .        (S=Spanned, N=Nonspanned or blank)
  System Managed Buffering  . .             (1K to 2048M or blank)
  System Determined Blocksize   Y      (Y or N)
  EATTR . . . . . . . . . . . . O      (O=Opt, N=No or blank)
Use ENTER to Perform Verification; Use UP/DOWN Command to View other Panels;
Use HELP Command for Help; Use END Command to Save and Exit; CANCEL to Exit.
```

Figure 2-5 Data Class Alter - EATTR

EATTR parameters

The EATTR parameters are as follows:

No — No extended attributes. This data set cannot have extended attributes and cannot reside in EAS.

Opt — Extended attributes are optional. This data set can have extended attributes and can optionally reside in EAS.

Blank — Defaults are used. For non-VSAM data sets the EATTR value used by the system is equivalent to NO. For VSAM data sets the EATTR value used by the system is equivalent to OPT. For those data sets that were created prior to EAS support, DFSMS is used.

Verifying catalog located in EAS

You can verify whether your data set resides in EAS by looking at your IEHLIST under the VOLUME section. In Example 2-2 you see from the LOW-CCHH and HIGH-CCHH values that this particular catalog is located in the EAS of volume MLDE65.

Example 2-2 LISTCAT of catalog showing EAS specification

```
LISTING FROM CATALOG -- SYSA.MHLEAV2.UCAT
VOLUME
  VOLSER------------MLDE65     PHYREC-SIZE--------10240     HI-A-RBA-------
  DEVTYPE------X'3010200F'     PHYRECS/TRK------------5     HI-U-RBA-------
  VOLFLAG------------PRIME     TRACKS/CA-------------15
  EXTENTS:
  LOW-CCHH-----X'FF0D0000'     LOW-RBA----------------0     TRACKS---------
  HIGH-CCHH----X'FF16000E'     HIGH-RBA---------7577599
```

Example 2-3 shows a LISTCAT output showing the EATTR(OPT) is specified for catalog SYSA.MHLEAV2.UCAT.

> **Note:** Using the EATTR(OPT) does not mean that the catalog will automatically be placed in the EAS. Only when there is no space available below the 65535 cylinder line will the catalog be created in EAS.

Example 2-3 LISTCAT of catalog showing EATTR parameter

```
LISTC ENTRY(SYSA.MHLEAV2.UCAT) ALL CAT(SYSA.MHLEAV2.UCAT)            00005118
IDCAMS  SYSTEM SERVICES                                            TIME: 11:59:01
                              LISTING FROM CATALOG -- SYSA.MHLEAV2.UCAT
CLUSTER ------- 0000000000000000000000000000000000000000000000
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2010.278
       RELEASE----------------2     EXPIRATION------0000.000
     SMSDATA
       STORAGECLASS -----EAVEXT     MANAGEMENTCLASS---MCDB22
       DATACLASS ---------EAVMV     LBACKUP ---0000.000.0000
       CA-RECLAIM---------(YES)
       EATTR--------------(OPT)
       BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
```

Example 2-4 is the IEHLIST utility JCL we ran against the volume where the catalog was created to list the EATTR(OPT) parameter specified.

Example 2-4 IEHLIST JCL

```
//IEHLIST EXEC PGM=IEHLIST
//INPUT DD UNIT=3390,VOL=SER=MLDC65,DISP=OLD
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
   LISTVTOC FORMAT,VOL=3390=MLDC65
   LISTVTOC   DUMP,VOL=3390=MLDC65
```

Example 2-5 is the IEHLIST output showing the EATTR(OPT) parameter specified.

Example 2-5 IEHLIST of catalog showing EATTR parameter

```
SYSA.MHLEAV2.UCAT                            MLDE65      1  2010.278    00.000
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST BLK(TTTT-
S   E         0            CYLS             100
EATTR   JOB        STEP      CREATE TIME
OPT     MHLRES3U   DEFCAT    20:33:40.639307
EXTENTS  NO       LOW(C-H)      HIGH(C-H)
          0      65646  0       65750 14
```

Changed messages

There are two changed messages (IGD17080I and IGD17163I) in support of catalogs residing in EF.

Message IGD17080I is issued when trying to make user catalogs or master catalogs EF. For z/OS V1R12, this EF restriction is removed and this message will no longer have references to catalogs.

Message IGD17163I is issued for compression requests. With DFSMS V1.12 user catalogs or master catalogs as a reason for the failed compression request is added because catalogs cannot be compressed.

Coexistence

z/OS V1R12 systems can share EAVs with back-level systems (z/OS V1R10 and z/OS V1R11), but only after explicit action is taken to begin allowing data sets to reside in the EAS of an EAV.

Applying APAR OA29932 will allow all z/OS V1R10 and z/OS V1R11 users with D/T2107 devices configured with EAV volumes to be able to share those volumes with a z/OS V1R12 System.

APAR OA29877 provides the coexistence code for z/OS V1R10 and z/OS V1R11 to recognize additional non-VSAM EAS eligible data set types in z/OS V1R12.

Ensure you have all the coexistence APARs to allow the use of EAVs. Note that there are also EAS coexistence APARs for products such as DFSMShsm and DFSMSdss. PTFs can be found in the PSP ZOSV1R12 DFSMS; additional maintenance will be added as it becomes available.

2.2 Contention detection for catalogs

This enhancement provides a way to identify certain catalog service tasks waiting on the input/output table resources (SYSZTIOT) to see if any are waiting beyond the default or specified wait time. The first time the system detects a CAS task waiting on the SYSZTIOT resource, it creates a SYMREC record to the logrec data set and the IEC393I message. If the task is still waiting the next time the system checks the CAS for tasks waiting on the SYSZTIOT resource, the system issues only system message IEC393I displaying information about the waiting task or tasks. You can specify a different wait time using the following command:

```
MODIFY CATALOG,CONTENTION(SYSZTIOT,wait_time)
```

The system will verify that the command to change the contention time has been successfully changed with the following messages:

```
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

The MODIFY CATALOG,REPORT command is also enhanced to display the wait time that is in effect, in a new CONTENTION line in the report output, as shown in Example 2-6.

Example 2-6 Modify Catalog Report command showing contention time

```
MODIFY CATALOG,REPORT
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I CATALOG REPORT OUTPUT 822
*CAS*********************************************************
*  CATALOG COMPONENT LEVEL   = HDZ1C10                      *
*  CATALOG ADDRESS SPACE ASN = 0037
*  SERVICE TASK UPPER LIMIT  =  180                         *
*  SERVICE TASK LOWER LIMIT  =   60                         *
*  HIGHEST # SERVICE TASKS   =   28                         *
*  # ATTACHED SERVICE TASKS  =   28                         *
.
```

```
.
.
*  NOTIFY EXTENT             =   80%                          *
*  DEFAULT VVDS SPACE        = ( 10, 10) TRKS                 *
*  CONTENTION SYSZTIOT TIME  =    10                          *
*  ENABLED FEATURES          = DSNCHECK DELFORCEWNG SYMREC    *
*  ENABLED FEATURES          = UPDTFAIL                       *
*  DISABLED FEATURES         = VVRCHECK AUTOTUNING BCSCHECK   *
*  INTERCEPTS                = (NONE)                         *
*CAS*************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

NOTIFY EXTENT

Another interesting piece of information from the MODIFY CATALOG,REPORT command in Example 2-6 is the NOTIFY EXTENT field. It is not new, but it is good to have it specified so you will be aware when the threshold is reached; this can helping you to avoid an outage.

The NOTIFY EXTENT value is the percentage of maximum extents at and above which message IEC361I will be issued. The system always issues message IEC361I whenever the percentage utilization for a catalog exceeds 90%. If this value is zero, it is displayed as (NONE), which means that no messages will be issued until 90% is reached or exceeded.

2.2.1 Specifying a wait time for catalog service task contention

You can use the new MODIFY CATALOG,CONTENTION command to specify a wait time in minutes for catalog service tasks to wait in contention for an event, after which a notification message will be issued. The command has two parameters:

- Reason class, which specifies the group of events that the wait-time field will apply to. Currently the only supported reason class is SYSZTIOT.
- Wait time, which specifies the length of wait in minutes before the message is issued.

The wait time value must be an integer in the range of 5 - 9999. The default value is 10, for 10 minutes. You can specify a value of 0 to disable the wait time notification function. The following command will change the notification period to 20 minutes from the default of 10 minutes:

```
MODIFY CATALOG,CONTENTION(SYSZTIOT,20)
```

2.2.2 Detecting catalog SYSZTIOT contention

To monitor contention for the task input/output table resources (SYSZTIOT), the system checks the catalog address space (CAS) for tasks waiting for the SYSZTIOT beyond the specified wait time (by default, 10 minutes). Once a task is identified as waiting beyond the specified wait time the system writes a SYMREC record to the logrec data set and issues an IEC393I message displaying information about the waiting task or tasks.

Example 2-7 shows the error message you will receive if the catalog has endured contention for your specified contention period.

Example 2-7 IEC393I message

```
IEC393I   CATALOG CONTENTION
*WARNING: CATALOG TASK CONTENTION WAIT-TIME WAS EXCEEDED.
*RESOURCE(SYSZTIOT)
```

```
JOBNAME(JOBNAME)
ASID(ASID)TCB(TCBaddr) or ASID(003C)#TASKS( #)
```

If the task is still waiting after 5 more minutes, this message is repeated and a new SYMREC record is written. This message will be repeated and a new SYMREC will be written every 15 minutes thereafter if nothing changes. If a new hung task is identified or an old hang is resolved, the notification is reset to the new state at the time of the next system check (within 30 seconds).

2.2.3 Commands to analyze SYSZTIOT contention

If you receive system message IEC393I notifying you that one or more tasks are waiting on the SYSZTIOT resource, you can use the information in the message to determine if any action is needed to resolve the wait. Note that receiving this message does not mean that an error or problem exists; you might not need to take any action at all. If you cancel any of the jobs listed, consider first taking a dump of the CAS.

TAKEDUMP command

Use the following command to take a dump of the CAS:

```
F CATALOG,TAKEDUMP
```

The result of the command is shown in Example 2-8. The DSN, STORCLAS, MGMTCLAS, DATACLAS in the output reflects your installation definitions.

Example 2-8 CATALOG TAKEDUMP command output

```
DUMPID=004 REQUESTED BY JOB (CATALOG )
DUMP TITLE=CAS DYNAMIC DUMP-IGG0CLGA RC246 RSN100
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
IEF196I IGD101I SMS ALLOCATED TO DDNAME (SYS00008)
IEF196I         DSN (DUMP.D110220.H23.SC64.CATALOG.S00004        )
IEF196I         STORCLAS (SCDUMP) MGMTCLAS (MCDB22) DATACLAS (
)
IEF196I         VOL SER NOS= SBOXZ1
IEF196I IGD104I DUMP.D110220.H23.SC64.CATALOG.S00004         RETAINED,
IEF196I DDNAME=SYS00008
IEA611I COMPLETE DUMP ON DUMP.D110220.H23.SC64.CATALOG.S00004 880
DUMPID=004 REQUESTED BY JOB (CATALOG )
FOR ASID (0036)
```

Information on waiting jobs

To get more information about the waiting jobs, use the information in message IEC393I.

Job name information

To get job name information and to gather additional information about the waiting task issue the following command:

```
F CATALOG,LISTJ(jobname)
```

Executing service tasks

To gather information about all currently executing service tasks, issue this command:

```
F CATALOG,LIST
```

Redrive SYSZTIOT holder

After getting task information, you can redrive the holder of the SYSZTIOT so that the waiting task can complete by issuing the following command:

```
F CATALOG,END(taskid),REDRIVE
```

Terminate waiting task

If the END(taskid),REDRIVE command does not resolve the contention, you can use the following command to terminate the waiting task:

```
F CATALOG,ABEND(taskid)
```

SYSZTIOT holding task

To list all the tasks currently holding SYSZTIOT, issue the GRS command:

```
D GRS,RES=(SYSZTIOT,*)
```

This is useful if the task with an exclusive hold on SYSZTIOT is not a CAS task. By default, the system lets a task wait for the SYSZTIOT task for 10 minutes before writing a SYMREC record to the logrec data set and issuing message IEC393I. You can specify a different wait time using the following command:

```
MODIFY CATALOG,CONTENTION(SYSZTIOT,wait_time)
```

If you specify a non-default SYSZTIOT wait time on the MODIFY command, you must re-specify that value after each system IPL. A non-default value does, however, persist through restarts of the CAS task.

You can display the current SYSZTIOT wait time using the MODIFY CATALOG,REPORT command as shown in Example 2-6 on page 14.

2.3 EAS VVDS

For your VVDS to take advantage of the EAS of an EAV volume you must allocate this data set using the access method services. After the volume has been initialized, use the ICKDSF INIT command to define the indexed VTOC. Use the DEFINE CLUSTER of IDCAMS to define the data set using the EATTR parameter.

Example 2-9 is an example of how to define VVDS with the EATTR parameter.

Example 2-9 Define cluster

```
DEFINE CLUSTER -
           (NAME(SYS1.VVDS.VMLDA60) -
           VOLUMES(MLDA60) -
           NONINDEXED -
           CYLINDERS(1 1) -
           EATTR(OPT))
```

After the cluster is defined you can see that the EATTR(OPT) is specified in the IEHLIST for this volume, as shown in Example 2-10.

Example 2-10 IEHLIST for volume showing EATTR parameter

```
---------------DATA SET NAME----------------  SER NO  SEQNO  DATE.CRE  D
SYS1.VVDS.VMLDA60                             MLDA60      1  2010.272 0
SMS.IND   LRECL  KEYLEN  INITIAL ALLOC  2ND ALLOC    EXTEND       LAST
```

```
              0            TRKS              15
EATTR   JOB        STEP      CREATE TIME
OPT     MHLRES3A   STEP1     18:53:37.603859
```

Note: It is a good practice to allocate the VVDS data set using the access method services right after you initialize the volume to avoid gaps that can be caused by dynamic VVDS allocation.

2.4 DEFINE RECATALOG support enhancement

Prior to DFSMS V1.12, when using the DEFINE RECATALOG command, the volumes were entered into the catalog record mostly in the specified order. The volume containing the VVR with RBA 0 was first, but after that the prime volumes were in the same order as in the DEFINE VOLUME parameter. When extent constraint relief for VSAM came about, an index was built over internal control blocks that map the extents. The DEFINE RECATALOG caused the index build to fail because VSAM Open expected the volumes in ascending RBA sequence. Not having an extent control block index might present performance problems for VSAM should a data set be re-cataloged with the volumes out of RBA sequence. Also, some DSS-like ISV products had difficulty with the re-cataloged data sets.

2.4.1 Multivolume and striped data set support

The solution was to place the volumes in the catalog record in ascending RBA sequence. This new function is in response to APAR OA24010. The IDCAMS DEFINE RECATALOG command is enhanced for multivolume and striped data sets, as follows:

- It automatically creates catalog entries with correctly ordered volume lists, while eliminating any duplicate volumes that might have been specified.
- It makes it easier to recatalog multivolume and striped VSAM data sets.

The new support is invoked by IDCAMS DEFINE RECATALOG command. Catalog entries for volumes are now ordered by RBA starting from 0 for data and index components regardless of the order in which they are listed in the DEFINE VOLUMES parameter. Candidate volumes follow the prime volumes. Prime volumes are those that have space assigned to them. There is no need for the user to do anything new or different. An example of this support is shown in Example 2-11.

Example 2-11 Define cluster showing multiple volumes

```
DEFINE CLUSTER -
     ( NAME(TEST.DS1) -
       INDEXED -
       RECATALOG ) -
     DATA -
      ( VOLUME(SMS001 SMS002 SMS003) ) -
     INDEX -
      ( VOLUME(SMS001 SMS003 SMS002) )
```

This enhancement puts the index portion volumes in the following order:

```
SMS001 SMS002 SMS003
```

For additional details see *z/OS DFSMS Managing Catalogs,* SC26-7409.

2.4.2 New reason codes for message IDC3009I

There are some new return and reason codes that indicate when volumes are missing in the user-supplied volume list. Since the catalog now orders the volumes by RBA, it can check to see if there are any gaps. Previously, only the first prime volume could be detected as missing. Here we list the message IDC3009I reason codes for return code 86 (Recatalog error).

Reason Code 24 DEFINE RECATALOG detected that a volume for a data component is missing from the volume list.

Reason Code 26 DEFINE RECATALOG detected that the last volume for a data component is missing from the volume list.

Reason Code 28 DEFINE RECATALOG detected that the last volume of a stripe for a data component of a striped VSAM data set is missing from the volume list.

Reason Code 30 DEFINE RECATALOG detected that a volume for an index component is missing from the volume list.

Reason Code 32 DEFINE RECATALOG detected that the last volume for an index component is missing from the volume list.

Reason Code 34 DEFINE RECATALOG detected that the last volume of a stripe for an index component of a striped VSAM data set is missing from the volume list.

2.5 Partial Release on multiple volumes

VSAM Partial Release is when unused space is released at the end of a data set. In prior releases, VSAM Partial Release only releases space on volumes where the high use RBA and the high allocated RBA are on the same volume. Another way of saying this is that VSAM Partial Release does not support releasing space that spans multiple volumes. Partial Release is expanded in V1.12 to include data sets that span multiple volumes. This only applies to SMS and Extended Format (EF) data sets.

2.5.1 EAV review

Extended Address Volume (EAV) was introduced in z/OS V1R10. EAV allows DASD storage volumes to be larger than 65,520 cylinders. The space above the first 65,520 cylinders is referred to as cylinder-managed space. Tracks in cylinder-managed space use extended addressing space (EAS) techniques to access these tracks. Data sets that are able to use cylinder-managed space are referred to as being EAS-eligible.

Multicylinder unit (MCU) is the smallest unit of disk space in cylinders that can be allocated in cylinder-managed space and mapping of the cylinder-managed space happens in MCUs. An MCU is a 21 cylinder unit. This means that all requested allocations in the cylinder-managed space will be rounded up to the nearest increment of 21 cylinders.

2.5.2 Partial Release implementation

In non-EAV data sets the starting point for freeing space must be a CA boundary; therefore, all space after the last used CA boundary up to the high allocated RBA is freed. In the case of EAV data sets, Partial Release must start at an MCU boundary and will free all space after the last used MCU boundary to the high allocated RBA.

Partial Release is activated on a VSAM CLOSE request, when either the SMS management class or the JCL DD SPACE=(,,RLSE)) statement is set for Partial Release. Space can also be release when DFSMShsm is performing space management or when an authorized program issues the PARTREL macro.

Partial release restrictions include:

- Partial release processing is supported only for extended format data sets.
- Data sets must be specified through SMS Management Class or by JCL specifying the SPACE=(,,RLSE) parameter on the DD statement.
- Only the data component of the VSAM cluster is eligible for partial release.
- Alternate indexes opened for path or upgrade processing are not eligible for partial release. The data component of an alternate index when opened as a cluster could be eligible for partial release.
- Partial release processing is not supported for temporary close.
- Partial release processing is not supported for data sets defined with guaranteed space.
- The data set must be opened for OUTPUT processing.

Attention: Partial release requires that the primary volumes in the data set be in ascending order by RBA. For example, the first volume could have RBAs 1 to 1000, the next 1001 to 2000, and so on. If the primary volumes appear out of order, partial release issues an error message and releases no space.

2.5.3 SMS Management Class options

Your options for Partial Release in SMS Management Class are as follows:

YES	Release unused space automatically during the Space Management cycle.
CONDITIONAL	Unused space can be released automatically only if a secondary allocation exists for the data set.
YES IMMED	Release unused space when a data set is closed or during the Space Management cycle, whichever comes first.

Consider this example: A one hundred cylinder data set is defined with best fit in an SMS management class where Partial Release is active. SMS has selected a storage group with three volumes (let us call them A, B, and C), each with 40 cylinders, and each currently empty. Sixty cylinders worth of data is written to the data set prior to closing. All data in volume A is below the calculated starting RBA and will not be processed. Volume B has both used and unused space. In this case, all of the unused space in volume B above the calculated starting RBA will be freed. Lastly, in volume C the data set will be scratched and the volume will be returned to the candidate list.

2.5.4 Partial release example

The LISTCAT in Example 2-12 is an example of an ESDS data set over two volumes not using Partial Release. You can see that there are 750 tracks allocated, and two volumes, but only 15 tracks that are actually used.

Example 2-12 LISTCAT of ESDS over two volumes not using Partial Release.

```
Command: LISTCAT ENT(dataset name) ALL
Result:
ALLOCATION
SPACE-TYPE------CYLINDER     HI-A-RBA--------37888000
SPACE-PRI-------------50     HI-U-RBA-----------40960
SPACE-SEC-------------50
VOLUME
VOLSER------------MLD20C     PHYREC-SIZE--------10240     HI-A-RBA--------378880
DEVTYPE------X'3010200F'     PHYRECS/TRK------------5     HI-U-RBA-----------409
VOLFLAG------------PRIME     TRACKS/CA-------------15
EXTENTS:
LOW-CCHH-----X'0C3D0000'     LOW-RBA----------------0     TRACKS---------------750
HIGH-CCHH----X'0C6E000E'     HIGH-RBA--------37887999
VOLUME
VOLSER------------MLD10C     PHYREC-SIZE--------10240     HI-A-RBA--------------0
DEVTYPE------X'3010200F'     PHYRECS/TRK------------5     HI-U-RBA--------------0
VOLFLAG-------CAND-SPACE     TRACKS/CA-------------15
```

Example 2-13 shows the same size data set with the same amount of data, but this time using partial release. You will notice that there are only 15 tracks and the second volume is a candidate volume. The system will not allocate space on candidate volumes until VSAM needs to extend to the candidate volume.

Example 2-13 LISTCAT of ESDS over two volumes using Partial Release

```
ALLOCATION
SPACE-TYPE------CYLINDER     HI-A-RBA----------757760
SPACE-PRI-------------50     HI-U-RBA-----------40960
SPACE-SEC-------------50
VOLUME
VOLSER------------MHLS1A     PHYREC-SIZE--------10240     HI-A-RBA----------757760
DEVTYPE------X'3010200F'     PHYRECS/TRK------------5     HI-U-RBA-----------40960
VOLFLAG------------PRIME     TRACKS/CA-------------15
EXTENTS:
LOW-CCHH-----X'01410000'     LOW-RBA----------------0     TRACKS----------------15
HIGH-CCHH----X'0141000E'     HIGH-RBA----------757759
VOLUME
VOLSER-----------------*     PHYREC-SIZE------------0     HI-A-RBA--------------0
DEVTYPE------X'3010200F'     PHYRECS/TRK------------0     HI-U-RBA--------------0
VOLFLAG--------CANDIDATE     TRACKS/CA--------------0
```

2.6 Indirect volser support for zFS data sets

A zFS file system uses a VSAM linear data set and, like all VSAM data sets, needs to be cataloged using a real volume serial number (volser). Using a real volser has brought up cloning and maintenance issues. These issues prevented those using zFS data sets from migrating these linear data sets. Indirect volser support for zFS data sets was requested to work around cloning and maintenance issues.

Linear data sets contain data that can be accessed as byte-addressable strings in virtual storage. A linear data set does not have imbedded control information that other VSAM data sets hold. You also can access linear data sets by using the z/OS data in virtual (DIV) access method or window services. VSAM record-level sharing (RLS) is not available for VSAM linear data sets.

In z/OS V1.12, the indirect volser support will allow you to clone the zFS file systems by using different volumes on different systems for a single catalog entry. The IDCAMS DEFINE RECATALOG commands now accept a symbolic value on the VOLUMES parameter for a zFS data set. You can use the symbolic value in a catalog entry when you want to use different volsers for different systems. The symbolic value can be set to a different volume serial number in an IEASYMxx parmlib member on each system. This will enable you to clone the zFS file systems to the volumes on the different systems. To use an indirect VOLSER for zFS data sets, you must set up a system symbolic in an IEASYMxx member of SYS1.PARMLIB.

2.6.1 Define zFS symbolic

The following example shows how to define a zFS using a symbolic. In this example, the symbolic name is &VOL01 and it is assigned a specific volume serial of MLDC65 to be used on that system. This definition must be added to the IEASYMxx member of SYS1.PARMLIB.

Example 2-14 SYSDEF for symbolic name VOL01

```
SYSDEF SYMDEF(&VOL01='MLDC65')
```

Note: The symbol must be set to a whole volume number.

Example 2-15 is the definition for the VSAM LDS using the real volume serial of MLDE65.

Example 2-15 Define Cluster using real volume serial

```
//DEFINE  EXEC PGM=IDCAMS
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
   DEFINE  CL (NAME(MHLEAV2.ZFS.TEST5)-
             CYL(2 1)-
             VOLUMES(MLDE65) -
             DATACLASS(EAVEXT) -
             STORCLAS(EAVEXT) -
             LINEAR)
```

Run the USS IOEAGFMT utility shown in Example 2-16 against this data set. This utility will format the LDS as a zFS and set the zFS indicator in the VVR catalog record for this data set.

Example 2-16 IOEAGFMT utility to format zFS

```
//FORMAT   EXEC   PGM=IOEAGFMT,REGION=0M,
// PARM=('-aggregate MHLEAV2.ZFS.TEST5 -compat')
//SYSPRINT DD     SYSOUT=*
//STDOUT   DD     SYSOUT=*
//STDERR   DD     SYSOUT=*
//SYSUDUMP DD     SYSOUT=*
//CEEDUMP  DD     SYSOUT=*
```

Use the DFSMSdss JCL shown in Example 2-17 to copy this zFS data set from volume MLDE65 to the system symbolic (&VOL01). Notice that the PHYSINDYNAM (PIDY) was used to create an uncataloged copy of the data set. The PHYSINDYNAM keyword specifies the input volume that is to be dynamically allocated to a physical copy or dump operation. A nonspecific volume serial number cannot be specified by using an asterisk (*). Only one volume can be specified for a FULL, TRACKS, or DATASET COPY. The device described by the volser must be the same type as the output device specified on the OUTDD or OUTDYNAM keyword.

Example 2-17 DFSMSdss parameters to copy zFS

```
//COPYZFS EXEC PGM=ADRDSSU
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
       COPY DS(INC(MHLEAV2.ZFS.TEST5)) -
       PIDY ( -
       (MLDE65) -
       ) -
       OUTDY ( -
       (MLDC65) -
       ) -
       REPLACEU -
       ALLDATA(*) -
       ALLEXCP
```

Run an IDCAMS DELETE NOSCRATCH to delete the catalog entry of this data set as shown in Example 2-18.

Example 2-18 DELETE NOSCRATCH

```
//DELETE   EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=A
//SYSIN    DD   *
      DELETE MHLEAV2.ZFS.TEST5 NOSCRATCH
```

Use IDCAMS DEFINE RECATALOG and the system symbolic (indirect volser) against the data set to create this data on the MLDC65 volume.

Example 2-19 DEFINE RECATALOG

```
DEFINE CLUSTER   -
    (NAME(MHLEAV2.ZFS.TEST5) -
    VOL(&VOL01) -
    LINEAR -
    RECATALOG)
```

A subsequent ICDCAMS LISTCAT will show that this data set is now using a device type of X'00000000' (see ❶ in Example 2-20).

Example 2-20 LISTCAT of zFS showing device type.

```
CLUSTER ------- MHLEAV2.ZFS.TEST5
...
DATA ---------- MHLEAV2.ZFS.TEST5.DATA
...
VOLUME
  VOLSER------------&VOL01     PHYREC-SIZE---------4096     HI-A-RBA---------147
  DEVTYPE------X'00000000'❶   PHYRECS/TRK-----------12     HI-U-RBA---------147
  VOLFLAG------------PRIME     TRACKS/CA-------------15
  EXTENTS:
  LOW-CCHH-----X'0C2D0000'     LOW-RBA----------------0     TRACKS--------------
  HIGH-CCHH----X'0C2E000E'     HIGH-RBA---------1474559
```

3

IDCAMS enhancements in DFSMS V1.12

This chapter describes the IDCAMS enhancements introduced in DFSMS V1.12.

There are two main improvements related to IDCAMS for this new z/OS release. The first enhancement offers a performance increase by reducing unnecessary recalls for deletion of GDGs HSM migrated.

Another improvement in this z/OS release is the addition of new DCOLLECT attributes. DCOLLECT reflects information about the attributes of OAM Protection and Job-name for EAV data sets that were introduced by V1R11 DFSMS and also reflects all the attributes that are supported by SMS Data Class.

3.1 DFSMS Catalog HDELETE for DELETE GDG Force

This IDCAMS enhancement gives you the option to delete an entire Generation Data Group with the FORCE command, even if there are migrated data sets. Starting with z/OS V1.12, after the DELETE command is issued the with FORCE optional parameter for a GDG base, the migrated data sets will not be recalled to be deleted; instead, an HDELETE command is issued to DFSMShsm, allowing the deletion without a recall.

This enhancement saves both CPU processing cycles and space because there is no need to have available space to recall the data set before deletion and no HSM recall processing is put in place.

3.1.1 IDCAMS DELETE GDG and FORCE parameters

In this section we present an overview of the IDCAMS DELETE command optional parameters GDG and FORCE.

GDG Specifies that the entry to be deleted is a generation data group (GDG) entry. To delete a generation data group that is not empty, you must specify either the FORCE or the RECOVERY parameter. When FORCE is issued, all SMS-managed generation data sets pointed to by the GDG base are scratched. Generation data sets are also removed from the catalog when you use FORCE.

FORCE Specifies whether entries that are not empty should be deleted. FORCE lets you delete generation data groups, tape library entries, and user catalogs without first ensuring that these entries are empty.

When deleting a generation data group using FORCE:

- The GDG entry is deleted even though it might point to non-VSAM entries in the catalog.
- Each SMS-managed non-VSAM data set entry pointed to by the GDG base entry is deleted before the GDG base entry is deleted. The non-VSAM data set is scratched.
- Each non-SMS-managed non-VSAM data set entry pointed to by the GDG base entry is deleted before the GDG base entry is deleted. However, the non-VSAM data set's space and contents on the volume are undisturbed.
- Non-SMS-managed non-VSAM data set entries in the user catalog are deleted, but the data sets are not scratched. A non-SMS-managed non-VSAM data set can be located with its DSCB in the volume's VTOC.

For more about IDCAMS commands and DELETE parameters, see *DFSMS Access Method Services for Catalogs,* SC26-73942.

3.1.2 Deleting Generation Data Sets using the MASK parameter

You can delete Generation Data Sets (GDSs) using the IDCAMS DELETE command along with the MASK parameter. This parameter enables deletion of a large number of GDSs at the same time.

To delete all generations on a GDG without deleting the GDG base, the command **DELETE MASK** deletes all the GDSs that match the mask in only one command.

An example of an IDCAMS DELETE using the MASK keyword is shown in Example 3-1.

Example 3-1 IDCAMS DELETE MASK command

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  DELETE MHLRES4.SMS.GDGTEST.ZOS111.G* MASK
```

Note: The GDSs that are HSM migrated will be recalled before the individual deletion.

3.1.3 Deleting a Generation Data Group base

To delete a GDG base and its GDSs, one option is to first issue an IDCAMS DELETE with the MASK parameter for the GDSs as shown in Example 3-1, and then delete the GDG bases with a GENERATIONDATAGROUP parameter as shown in Example 3-2.

Example 3-2 IDCAMS DELETE GDG

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  DELETE MHLRES4.SMS.GDGTEST.ZOS112 GDG
```

With those two steps (the deletion MASK for GDSs and then DELETE GDG for base) the GDG specified on the DELETE command is removed from the system. However, as mentioned previously, the GDSs migrated by HSM will be recalled before delete.

One option for deleting this GDG without impacting HSM is to issue DELETE GDG FORCE.

3.1.4 Deleting Generation Data Group using Force

When an entire GDG is to be deleted you can use the IDCAMS DELETE command along with the GDG and FORCE parameters to cause the GDG to be deleted with one command. Sample JCL to delete the entire GDG using the GDG and FORCE parameters is shown in Example 3-3.

Example 3-3 IDCAMS DELETE GDG FORCE

```
//STEP01   EXEC PGM=IDCAMS
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN    DD *
  DELETE MHLRES4.SMS.GDGTEST.ZOS112 GDG FORCE
```

Prior to z/OS V1.12 any GDS that had been migrated by DFSMShsm had to be recalled before it could be deleted using the DELETE GDG FORCE command.

With DFSMS V1.12 when a DELETE GDG FORCE command is issued, it is not necessary to recall a migrated GDS before the GDG can be deleted. An HDELETE is issued for the migrated GDS which updates the HSM MCDS and deletes the migrated GDS. The GDS does not have to be recalled prior to being deleted.

For more information about Generation Data Group, see *DFSMS Using Data Sets,* SC26-7410.

3.1.5 Migration and coexistence considerations

There are no coexistence requirements or migration actions to use this new capability.

For z/OS versions prior to z/OS V1.12 the IDCAMS DELETE GDG FORCE command will work like before, requiring recall of the data set prior to deletion.

3.2 DFSMS AMS DCOLLECT enhancements

DCOLLECT is an IDCAMS command to gather system information and write machine-readable records. The records produced by DCOLLECT can be used as input for a variety of applications and jobs.

IDCAMS DCOLLECT gathers data about volume, data set, and system attributes, and places the read-only values in a sequential file that can serve as input to other programs or applications.

A simple way to generate a report with DCOLLECT as input is to use the NaviQuest SMS tool.

In z/OS V1R12, DFSMS Access Method Services adds a number of attributes to the output produced by the DCOLLECT command.

3.2.1 DCOLLECT attribute enhancements

The DCOLLECT enhancement in z/OS V1.12 presents new attributes for three record types:

- Data Class records
- Data set records
- Storage Class records

Data Class record attributes

As of DFSMS V1.12, DCOLLECT output includes the following new attributes for the Data Class type records:

- VSAM SPEED and REUSE
- Tailored compression
- CICSVR forward log
- RLS greater than 4K cache
- Block size limit
- Dynamic volume count
- RLS 64 bits virtual (RLS above the bar)
- Tape performance scaling
- Tape performance segmentation
- System managed buffering
- Tape encryption (key-label, key-code)
- CA reclaim
- EATTR value

Data Class type record new attributes

- **DDCSPEED:** This field indicates how the data set is to be loaded. The default is zero (the recovery mode); possible values are:

  0 Recovery mode; the data set is preformatted when it is loaded.

1 Speed mode; the data set is loaded without reformatting.

- **DDCCT:** This field indicates the method of compressing a data set. It can be GENERIC or TAILORED. The default value is GENERIC; possible values are:

  0 The data set is compressed using Generic Dictionary Building Block (DBB) compression, which is derived from a defined set of compression algorithms in data set SYS1.DBBLIB.

  1 The data set is compressed using a tailored dictionary. When the initial data is written to a data set, the tailored dictionary is built and imbedded to the data set. This type of compression applies only to sequential data sets, not to VSAM KSDSs.

- **DDCLOGNM:** The ID that identifies the CICS Forward Recovery Log stream. It is a string of 1 - 26 characters in length.
- **DDCDSCF:** This field shows how an RLS data set is cached. It mapped constants DCDDSCFA (0), DCDDSCFU (1), DCDDSCFN (2). The default value is zero; possible values are:

  0 Data set caching with all the requests.

  1 Data set caching with updated request only.

  2 Caching of directory entries only.

- **DDCDVCS:** Dynamic Volume Count specified.
- **DDCBLMT:** Block Size Limit specified.
- **DDCRBYTE:** A one byte flag for RLS attributes.
- **DDCA2GB:** This field shows whether the SMSVSAM address space can be above the 2 GB bar, and thus able to take advantage of 64 bit addressable virtual storage during VSAM RLS buffering. The default value is zero(0); possible values are:

  0 VSAM RLS buffering is limited to storage below the bar.

  1 VSAM RLS buffers can reside above the 2 GB bar.

- **DDCTAPE1:** An eight-byte field that contains the support of TAPE attributes.
- **DDCPSCA:** This performance scaling field shows the tape usage selection. It is one byte in length. The default value is zero; possible values are:

  0 The tape selection will use the tape at its full capacity (same as 2).

  1 The tape selection is called to optimal performance.

  2 The tape selection will use the tape at its full capacity.

- **DDCPSEG:** This performance segmentation field shows how a cartridge is divided into a fast access segment and a slower access segment. The fast access segment will be filled in first, after which the slower access segment will be filled. The default value is zero; possible values are:

  0 The segmentation format is disabled (same as 2).

  1 The segmentation format is enabled.

  2 The segmentation format is disabled.

- **DDCVSP:** This system-managed buffering (SMB) field shows how many buffers are to be used and how they are processed for VSAM applications that request the use of non-shared resources (NSR). The value range is 1 - 2048 M.
- **DDCVSPU:** Contains one byte unit of the SMB value.
- **DDCVSPV:** Contains the SMB value in length of 3 bytes.

- **DDCKYLB1:** The first type Key Label field that is used for secure cryptographic hardware to encrypt the data to be dumped.
- **DDCKLBL1:** Length of key label 1.
- **DDCKLBN1:** The first type Key Label. It can be 1 - 64 characters long.
- **DDCKYCD1:** This is the first type of Key Code that is used for Encoding Mechanism Key Label Performance Segmentation.
- **DDCRECLM:** This field shows whether the free CA space on DASD can be reclaimed for VSAM and BSAMRLS. When the free space is placed for a new CA, a free CA space in the free chain is reused. The default value is zero (ENABLE); possible values are:

  0 CA reclaim is enabled.

  1 CA reclaim is disabled.

  2 Caching directory entries only.

Data set type record attributes

With DFSMS V1.12, the DCOLLECT output has the following new attribute for the data set type DCOLLECT records:

- **DCDDS9F1:** A one-byte field flag for DCSB Format 9. The following are indicated if the specified bit is 1:

  DCDDS9CR Format 9 DSCB has been built by DADSM CREATE DCDJBNMC. This field contains the name of the JOB that created the data set described by its Format 8 DSCB.

  DCDSTNMC This field contains the name of the STEP that created the data set described by its Format 8 DSCB.

  DCDTIMEC This field contain the value in microseconds since midnight, local time, that the date set described by its Format 8 DSCB was recreated.

Note: With z/OS V1.12 DCOLLECT the data set type records and related data set creation records will contain information for EAS-eligible data sets. This is because the Format 9 DSCB exists only for EAS-eligible data sets.

Storage Group record attributes

With the enhancements in DFSMS V1.12, the DCOLLECT output will have the following new attributes added for the Storage Group type DCOLLECT records:

- Storage Group type record
  - OAM Retention Protection
  - OAM Deletion Protection

Storage Group type record new attributes

- **DSGFRETP:** This flag indicates whether Retention Protection is enabling for a given object. When it is enabled, 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 change to a later date. The possible values are:

  1 Retention Protection is enabled.

  0 Retention Protection is disabled.

- **DSGFDELP** This flag indicates whether Deletion Protection is enabling for a given object. When it is enabled, OAM will not allow that object to be deleted prior to its expiration date.

Deletion Protection is different from Retention Protection in that Deletion Protection can be turned on and off by the installation, and Deletion Protection has no restriction on the expiration date changing.

1 Deletion Protection is enabled.

0 Deletion Protection is disabled.

For more details about the new DCOLLECT attributes see *z/OS DFSMS Access Method Services for Catalogs*, SC26-7394.

3.2.2 SMS NaviQuest

To display the new attributes, we used the NaviQuest tool to create reports that use DCOLLECT as input.

Start NaviQuest by selecting Option 11, Enhanced ACS Management, from the ISMF Primary Option Menu. This displays the NaviQuest Primary Option Menu shown in Figure 3-1 on page 31. When you select one of the options from the NaviQuest Primary Option Menu, each successive panel guides you through the choices available for that function.

```
 ------------------------------------------------------------------------
            ENHANCED ACS MANAGEMENT - NaviQuest PRIMARY OPTION MENU
 Enter Selection or Command ===>

 Select one of the following options and press Enter:

   1  Test Case Generation
   2  ACS Test Listings Comparison
   3  Enhanced ACS Test Listing
   4  Test Case Update with Test Results
   5  SMS Report Generation
   6  Model Commands Generation
   7  Batch Testing/Configuration Management
   X  Exit









  F1=Help    F2=Split   F3=End     F4=Return  F7=Up      F8=Down    F9=Swap
 F10=Left   F11=Right  F12=Cursor
```

Figure 3-1 Enhanced ACS Management NaviQuest Primary Option Menu

For the DCOLLECT report we selected option 5 SMS Report Generation from the main NaviQuest menu.

The next panel, SMS Report Generation Selection Menu displays the report options related to DCOLLECT data (Figure 3-2).

```
SMS REPORT GENERATION SELECTION MENU
Enter Selection or Command ===>

Select one of the following options and press Enter:

  1  Data Set Report from Saved ISMF List
  2  DASD Volume Report from Saved ISMF List
  3  Tape Volume Report from Saved ISMF List
  4  Data Set Report from DCOLLECT Data (D Records)
  5  DASD Volume Report from DCOLLECT Data (V Records)
  6  DFSMS Configuration Report from DCOLLECT Data (DC, SC,
     MC, SG, BC, VL, AG, DR, LB, CN, or AI records)
```

Figure 3-2 SMS Report Generation Option Panel

The first three options are used if you created an online data source using the ISMF panel. The three last options apply to DCOLLECT input data.

Using DCOLLECT as input, you can generate the following reports:

- Data Set Report (D Records)
- DASD Volume Report (V Records)
- SMS Configuration Report

The enhancements in this z/OS version are related to DFSMS Configuration type records (Data Class and Storage Group records) and data set type records. To present those new attributes, we selected option 4 (Data Set Report) and option 6 (DFSMS Configuration Report).

3.2.3 NaviQuest IDCAMS report panel examples

There are some mandatory parameters and some optional parameters that can be coded to create a DCOLLECT report. To create the input to the following NaviQuest reports, a simple DCOLLECT job was run, listing all data sets on the current system.

SMS Data Class and Storage Group record attributes

The DCOLLECT data extract to generate this type of report must have a mandatory specific field coded indicating that the data to be extracted is SMS related.

Example 3-5 contains JCL used to create DCOLLECT SYSOUT for the new SMS DCOLLECT attributes.

Example 3-4 DCOLLECT JCL example for SMS data

```
//DCOLLECT EXEC PGM=IDCAMS
//DCOLLOUT  DD DSN=HLQ.DCOLLECT.ZOS12,DISP=(,CATLG),
//         LRECL=32756,RECFM=VB,UNIT=SYSALLDA,SPACE=(CYL,600,RLSE)
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD  *
  DCOLLECT OFILE(DCOLLOUT) VOLUMES(*) SMSDATA(ACTIVE)
```

For more information about DCOLLECT parameters for IDCAMS, see *DFSMS Access Method Services for Catalogs,* SC26-73942.

To generate SMS configuration reports from the DCOLLECT data extracted, select Option 6 from the SMS Report Generation Selection Menu shown in Figure 3-3.

```
SMS REPORT GENERATION SELECTION MENU
Enter Selection or Command ===>

Select one of the following options and press Enter:

  1  Data Set Report from Saved ISMF List
  2  DASD Volume Report from Saved ISMF List
  3  Tape Volume Report from Saved ISMF List
  4  Data Set Report from DCOLLECT Data (D Records)
  5  DASD Volume Report from DCOLLECT Data (V Records)
  6  DFSMS Configuration Report from DCOLLECT Data (DC, SC,
     MC, SG, BC, VL, AG, DR, LB, CN, or AI records)





 F1=Help    F2=Split   F3=End     F4=Return  F7=Up      F8=Down    F9=Swap
F10=Left   F11=Right  F12=Cursor
```

Figure 3-3 SMS Report Generation Selection Menu

Data Class new attributes

As shown in Figure 3-4, select the fields to be included in the report by specifying either Yes or No (`Y` or `N`):

Y Include this record in the report.

N Do not include this record in the report.

```
DFSMS CONFIGURATION REPORT FROM DCOLLECT DATA ENTRY PANEL
Command ===>

To generate report, specify the following information and press Enter:
  DCOLLECT Data  . . 'MHLRES4.DCOLLECT.EXTRACT'
  Data Set to Hold Report
    ===> 'MHLRES4.NAVIQ.REPORT'
    Replace Contents if DSN Exists . . N  (Y or N)   Page Length  . . 100

Specify information to be included in the report: (Y or N)

  Data Classes (DC) . . . . Y       Storage Group Volumes (VL)
  Storage Classes (SC)  . .         OAM Drives (DR)  . . . . . .
  Management Classes (MC)           OAM Libraries (LB) . . . . .
  Storage Groups (SG) . . .         Cache Names (CN) . . . . . .
  Base Configuration (BC)           ACS Accounting (AI)  . . . .
  Aggregate Groups (AG) . .
```

Figure 3-4 NaviQuest DFSMS Configuration Report from DCOLLECT Data Panel - DC Selection

Using the Data Class selection shown here, the report will show you all the related Data Class information in the DCOLLECT data set.

The report will be saved in the specified data set in the field `Data Set to Hold Report` and also will be displayed on the screen after it is created.

For each defined Data Class, the new fields will be displayed as shown in Figure 3-5 on page 34.

```
-----------------------------------
-- Dataclas Record : WELCHRLS   --
-----------------------------------

Last Updated: MHLRES1  Date: 2010/04/06
Time Updated: 12:05
Description:
.
.
.
Override Space           :
.
.
.
System Managed Buffering :
System Determ Blocksize  :
.
.
.
Performance Scaling      :
Performance Segmentation :
.
.
.
FRlog                    :
RLS CF Cache Value       : ALL
RLS Above the 2-GB Bar   : NO
.
CA Reclaim               : YES
Key Label 1              :
Encoding for Key Label 1 :
Key Label 2              :
Encoding for Key Label 2 :
```

Figure 3-5 New Data Class NaviQuest fields

Not all fields that were added to DCOLLECT are shown on the NaviQuest panel, but all those fields listed in “Data Class type record new attributes” on page 28 are available in DCOLLECT output and can be used for other applications that can have DCOLLECT data sets as input.

Storage Group new attributes

To produce a report using as input the DCOLLECT data extract based on the sample job in Example 3-4 on page 32, set the Storage Group option to `Y` as shown in Figure 3-6 on page 35.

```
DFSMS CONFIGURATION REPORT FROM DCOLLECT DATA ENTRY PANEL
Command ===>

To generate report, specify the following information and press Enter:
  DCOLLECT Data  . . 'MHLRES4.EXTDC.TEST8'
  Data Set to Hold Report
    ===> 'MHLRES4.NAVIQ.REPORT'
    Replace Contents if DSN Exists . . Y  (Y or N)   Page Length  . . 100

Specify information to be included in the report: (Y or N)

  Data Classes (DC) . . . .         Storage Group Volumes (VL)
  Storage Classes (SC)  . .         OAM Drives (DR)  . . . . . .
  Management Classes (MC)           OAM Libraries (LB) . . . . .
  Storage Groups (SG) . . . Y       Cache Names (CN) . . . . . .
  Base Configuration (BC)           ACS Accounting (AI)  . . . .
  Aggregate Groups (AG) . .
```

Figure 3-6 NaviQuest DFSMS Configuration Report from DCOLLECT Data Panel - SG Selection

Using this Storage Group selection, the report will show you all the Storage Group information in the DCOLLECT data set, including the new attributes.

The report will be saved as specified in the field `Data Set to Hold Report` and also will be displayed on the screen after it is created.

For each Storage Group in the system, listed on DCOLLECT, the new fields shown in Figure 3-7 will be displayed.

```
-----------------------------------
-- Storgrp Record : CPBSG0    --
-----------------------------------

Last Updated: MHLRES1 Date: 2010/09/29
Time Updated: 20:30
Description:

Storage Group Type      : UNKNOWN
.
.
.
.
OAM Deletion Protection : N/A
OAM Retention Protection: N/A
```

Figure 3-7 Storage Group new fields on NaviQuest

The two fields in Figure 3-7 shown in bold are related to the new attributes on Storage Group record.

Data Set record NaviQuest Panel

To generate Data Set reports from DCOLLECT data, specify option 4 from the SMS Report Generation Selection Menu. Select the fields to be included in the report with a number for each field to indicate the order in which they are to be printed as shown in Figure 3-8.

```
DATA SET REPORT FROM DCOLLECT DATA ENTRY PANEL
Command ===>

To generate report, specify the following information and press Enter:
  DCOLLECT Data  . . 'MHLRES4.EXTDC.TEST8'
  Data Set to Hold Report
    ===> 'MHLRES4.NAVIQ.REPORT'
    Replace Contents if DSN Exists . . Y  (Y or N)  Page Length    100
  Max Length of DSN Print  . . . . . . 15               Totals    N  (Y or N)

Specify fields in numeric order (max width of report is 133 characters):
                          Length                                  Length
                                                                   More:     +
  Data Set Name  . . . 1   (45)    Chg Ind  . . . . . .      (7)
  Alloc Space  . . . . 2   (10)    DS Org . . . . . . .      (6)
  % Used . . . . . . . .    (6)    Rec Fmt  . . . . . .      (7)
  Block Unused . . . .      (9)    Record Length  . . .      (6)
  Blk Sz CI Size . . .      (8)    SMS Managed  . . . . 3    (5)
  Expire date  . . . . 4   (11)    Volume Serial  . . .      (7)

Use DOWN command to Scroll Forward; Use UP Command to Scroll Backward;
```

Figure 3-8 NaviQuest DFSMS Configuration Report from DCOLLECT Data Panel - Data Set Selection

For this report, one line is generated for each data set found in the DCOLLECT input. Each line cannot exceed 133 characters, so only select the desired fields for the report. If more than 133 characters were selected, the report will not be built and the following error message will be displayed:

```
You have exceeded the maximum record length with the number of variables you
have chosen. Please select fewer variables for the report. You cannot exceed
133 in length.
```

The NaviQuest panel does not present the new attributes of Data Set record for selection.

Note: At the time of the writing of this book, the new attributes for Data Set record were not added to the NaviQuest panel. Other than that, the DCOLLECT has all the fields recorded as described.

For more details about using NaviQuest with IDCAMS or ISMF inputs, see *z/OS DFSMS Storage Administration Reference (for DFSMShsm, DFSMSdss, DFSMSdfp)*, SC26-7402.

3.2.4 Migration and coexistence considerations

There is no coexistence requirement or migration action to use this new enhancement.

At the time of this writing there is one APAR OA33492 for NaviQuest SMSDATA panels to include these new fields. The APAR must be applied to have the Storage Group and Data Class new fields available in the NaviQuest panels.

4

VSAM and VSAM RLS

This chapter covers the enhancements to VSAM and Record Level Sharing. These enhancements include:

- Control area reclaim feature for VSAM and VSAM RLS KSDSs
- Disk striping support for VSAM RLS
- VSAM trace improvements

4.1 VSAM enhancement summary

This new z/OS release enhances VSAM and VSAM RLS to reclaim empty space (control areas) in key-sequenced data sets (KSDS). CA reclaim applies to both SMS and non-SMS data sets and eliminates a major reason for KSDS reorganizations.

The new version also provides access to volume striped data sets for VSAM RLS. Previously that access was allowed only for VSAM data sets, but in DFSMS v1.12 the RLS restriction was removed.

The new interface extends the usability of VSAM Record Management Trace to support data sets that are allocated dynamically and allows the user to enable VSAM Record Management Trace without taking the application or data set offline.

4.2 Control area reclaim for VSAM and VSAM RLS

The control area (CA) reclaim enhancement reduces the need for outages of VSAM-based applications that are the result of reorganization demands.

Erasing records from a KSDS might result in space for empty CAs that is not reclaimed. Over time, this can produce DASD space that is fragmented and performance that is degraded due to an index structure that is unnecessarily complicated by the empty CAs. To reclaim the empty CA space and streamline the index structure, you can reorganize the KSDS. To reduce the need for reorganizing a KSDS, you can cause empty CA space on DASD to be reclaimed automatically, so that it can be reused at a later time when a CA split is required. The CAs that have been reclaimed are available to be used for new records without any processing to obtain new space. A CA split that reuses a reclaimed CA will not change the high used RBA.

4.2.1 Key-sequenced data sets

In a key-sequenced data set, logical records are placed in the data set in ascending collating sequence by a field, called the *key*. Figure 4-1 shows that the key contains a unique value, such as an employee number or invoice number, that determines the record's collating position in the data set.

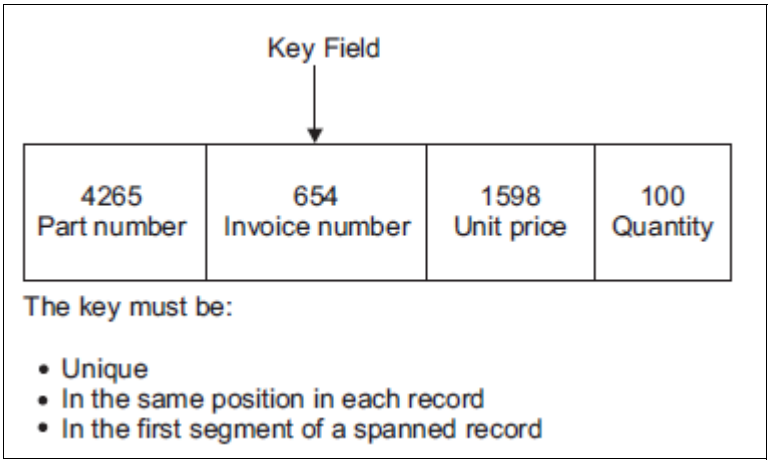

Figure 4-1 VSAM KSDS Key insertion model

The key must be in the same position in each record, the key data must be contiguous, and each record's key must be unique. After it is specified, the value of the key cannot be altered, but the entire record can be erased or deleted. For compressed data sets, the key itself and any data before the key will not be compressed.

Fragmentation occurs when records are erased from a KSDS; the fragmentation increases until a VSAM reorganization is needed to remove the fragmented spaces and provide new contiguous space for new record additions. This is problematic in production environments with a 24x7 availability requirement because sometimes it is necessary to stop CICS, DB2®, or even a whole Sysplex system.

KSDS applications

Key-sequenced data sets are best for the following kinds of applications:

- Applications that require each record to have a key field
- Applications that require both direct and sequential access
- Applications that use high-level languages that do not support RBA use
- Online applications
- Applications whose data you want to access by an alternate index

More about KSDS and VSAM general information can be found *z/OS V1R12 DFSMS Using Data Sets,* SC26-7410.

4.2.2 CA reclaim

With the z/OS V1.12 release, DFSMS provides the functionality to have the space fragmentation reduced by CA reclaim enhancement. This function reduces the need for KSDS reorganizations. Because free CAs are reclaimed and placed in a free chain, there will be fewer CIs in the index structure, resulting in quicker access.

There are additional overhead savings because when space is required for a new CA, reusing the CAs that have been placed on the free chain means fewer calls to EOV to extend the KSDS. This will also reduce the number of new extents needed for a KSDS.

Empty CAs in a KSDS

Adding and deleting logical data records to a KSDS can lead to CI and CA splits. Eventually an entire Data CA might become empty in the sense that it does not contain any active logical data records any longer because all the records were deleted over time (see Figure 4-2 on page 40).

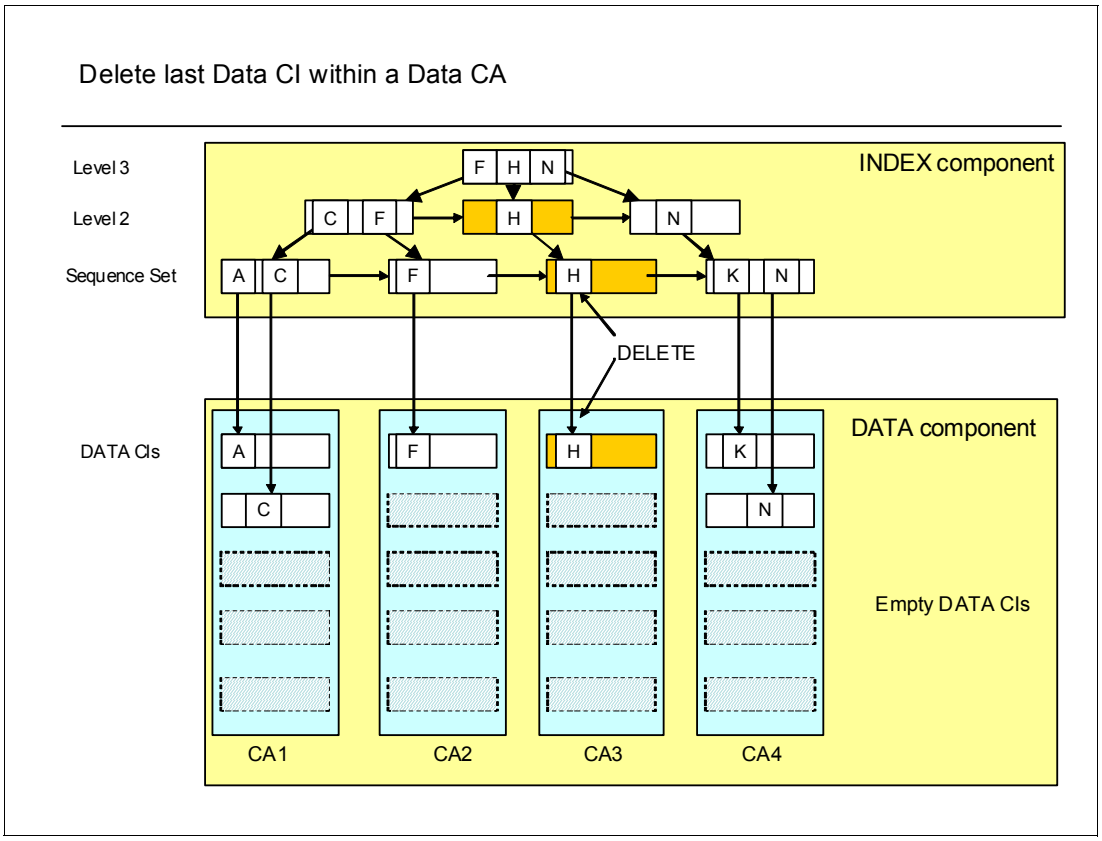

Figure 4-2 KSDS with empty Data CA

Figure 4-2 shows a logical KSDS structure with a Data component and an Index component, which together form a KSDS Cluster. The Index contains all the data keys and pointers to the lower Index levels. The lowest Index level, the Sequence set, covers a corresponding Data CA. A key within a sequence set index CI points to the related Data CI. The Sequence set also has a logical horizontal pointer chain, which is utilized when reading the KSDS sequentially in ascending key order. The Data component contains the actual data and is organized in Control Areas (CAs), which contain a fixed number of Data CIs.

The characters in Figure 4-2 symbolize the key that is used to find the logical data records. For example, when locating the logical record with the key E the Index chain follows from F in Level 3 to F in level 2 and finally to F in the Sequence Set CI which point to the Data CI in CA 2. Note that the character for the key is the highest key within a CI, so the logical record with the E key is in the CI which has F as its highest key.

Figure 4-2 shows that the last record in CA 3 in an active Data CI is about to be deleted. We assume that H is the key of the one and only logical record which is left in the H Data CI in CA 3.

Figure 4-3 on page 41 shows the structure after the H record is deleted.

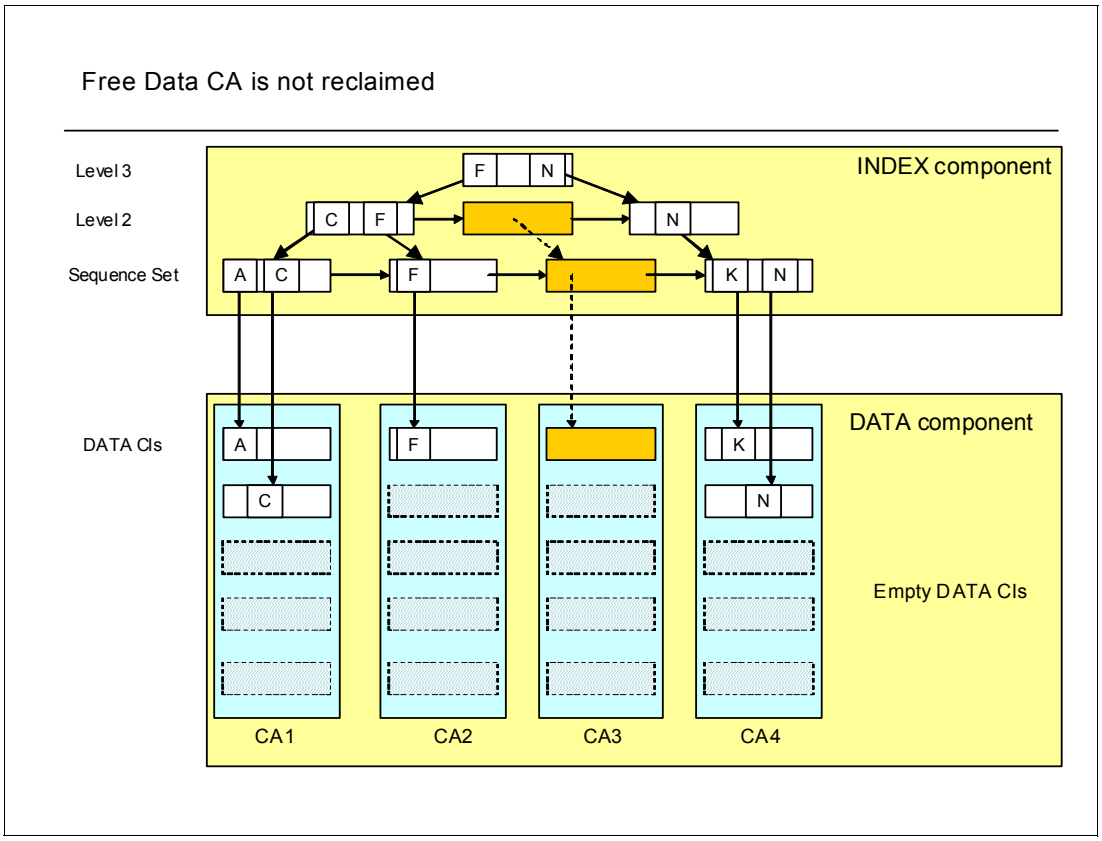

Figure 4-3 Empty Data CA is not always reused

Before CA reclaim the basic pointer structure in the Index component is unchanged. Even the Level 2 index CI stays where is was before. When a new record is added with key P a new Data CA is created, and after formatting the new Data CA plus a corresponding sequence set index CI, the record with the key P is added. The empty CA 3 is still unused and empty. This applies also to the empty Index CI in Index Level 2. Even when sequentially browsing through the sequence set chain the empty sequence set CI is accessed as well.

Only when a new logical record with a key between F and K is added will the empty CA plus its Sequence Set CI be reused. But the key must fit exactly between the last logical key in CA 2 and the first logical key in CA 4.

CA reclaim process

With CA reclaim an effort was made to unchain the empty Index Set CIs, including the Sequence Set CIs, and provide the empty Data CA for the next logical record, which has a key larger than N or the highest key in CA 4 when CA 4 is filled up. It is also reused when a CA split occurs.

Figure 4-4 on page 42 shows the basic logic behind the CA reclaim process.

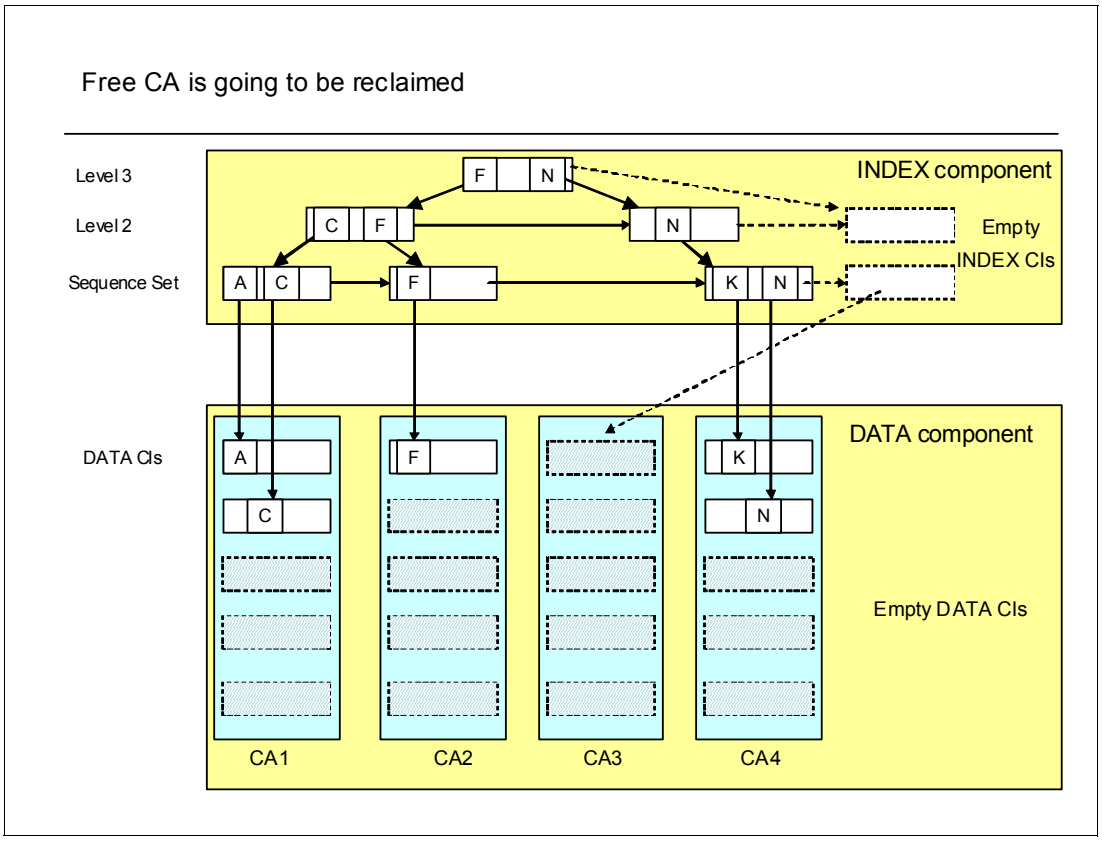

Figure 4-4 CA reclaim after it became empty

As soon as the last logical record is deleted in CA 3, VSAM unchains the corresponding Index CIs and positions these Index CIs to be available for the expanding Data component – either through a key that points beyond the CA 4 or through a CA. The dashed lines indicate the potential structure of the reused Data CA 3 and its index pointers. Note that the Data component and the Index component do not expand beyond their original size.

You can request that empty CA space for a KSDS be reclaimed automatically. In previous releases, VSAM KSDSs must be “reorganized” on a regular basis in order to do the following:

- Reclaim space previously used by deleted records
- Improve sequential read performance

CA reorganization process

The CA reorganization process consists of the following functions:

1. Closing the KSDS
2. Unloading the KSDS to a backup file
3. Deleting and redefining the KSDS
4. Reloading the file from the backup file
5. Reopening the KSDS

This process affected many aspects of a z/OS environment, such as:

- Customer application data
- IMS™ and CICS files

- Catalogs
- Control data sets for many IBM products such as DFSMShsm and DFSMSrmm

The last empty control interval within the control area is not CI reclaimed. The high key value for that control interval is maintained and becomes the highest key for any record that can be inserted into that control area. When CA reclaim is not enabled, a KSDS might have many empty control areas and might continue to grow in size. This occurs when applications continually add records with keys that are in ascending sequence, followed by another or the same application that deletes old records. During the deletion processing, the high-key value that was associated with that CA will be maintained, requiring that only records falling within that high-key range are eligible for insertion into that control area. If the record keys are always getting higher, no new records will qualify for insertion into those empty control areas. The result is a data set in which a majority of the space is occupied by empty control intervals.

4.2.3 Enabling CA reclaim functionality

To implement CA reclaim, empty sequence set and high level index records are placed on a free list after the last record in the CA is erased. The reclaimed CA can then be reused when new records are inserted in the data set, provided that the size of CA needed is less than or equal to that of the reclaimed empty CA.

Pre-existing empty CAs will not be reclaimed by CA reclaim. Once all sharing systems are upgraded to z/OS V1R12 and enabled for the CA reclaim function, the existing data sets should be reorganized to remove any pre-existing non reclaimable empty CAs.

Benefit: CA reclaim addresses the main reasons for reorganizations, thereby helping to make data sets available 24 hours a day.

System default

The system default is CA Reclaim disabled; the system programmer can choose to activate the CA reclaim functionality or keep the system default. Also, in sysplex multi-level systems, some co-existence PTFs should be applied to have this work on z/OS V1.12 systems. Coexistence is discussed in more detail at the end of this chapter.

After the z/OS V1.12 upgrade is done the CA reclaim must be enabled at the system level.

Parmlib updates

A new system level parameter in the IGDSMSxx parmlib member is required to enable this function:

```
CA_RECLAIM({NONE | DATACLAS | DATACLASS})
```

This parameter specifies whether to use CA reclaim for KSDSs based on the CA reclaim attribute in their data classes, as follows:

`NONE` Indicates that none of the KSDSs will be using CA reclaim, regardless of the data class specification. This is the default. If CA_RECLAIM is not specified in SYS1.PARMLIB, the CA_RECLAIM(NONE) default will be in effect.

`DATACLAS | DATACLASS` Indicates that SMS-managed KSDSs and non-SMS-managed KSDSs will go by the data class specification of CA reclaim=Y|N at define time or subsequent ALTER setting. Only the resultant catalog setting of CA Reclaim=Y will do CA reclaim; CA Reclaim=N will not do CA reclaim. This enables you to specify that some KSDSs will not use CA reclaim.

SMS command

The SMS console command SETSMS CA_RECLAIM(NONE | DATACLAS | DATACLASS) has the same options as those of the PARMLIB but takes effect immediately, without an IPL.

CA reclaim is disabled by default on a system level, but is enabled by default for all KSDSs without having to redefine the data set. Once you enable CA reclaim by command or at IPL time with the parmlib command, it will take effect for all KSDSs.

IDCAMS ALTER

IDCAMS ALTER can be used to disable or enable individual KSDSs (SMS or non-SMS) after DEFINE time as follows:

```
ALTER 'ksdsname' RECLAIMCA|NORECLAIMCA
```

Note: ALTER does not immediately disable/enable CA reclaim, unlike the **SETSMS** command. ALTER immediately updates the catalog but the attribute in the catalog is not looked at by the system until the next time the control block structure is built, which means it will not take effect until all the current OPENs against the control block structure have been closed and a subsequent first OPEN is issued.

Data Class considerations

A new field was created on the Data Class panel to reflect CA reclaim functionality, as shown in Figure 4-5. When creating a new Data Class you can set the field CA RECLAIM (49) to YES to enable automatic CA Reclaim for the Data Class data sets; set CA RECLAIM to NO to disable CA Reclaim for the data sets.

```
DATA CLASS LIST
Command ===>                                                Scroll ===> CSR
                                                            Entries 1-2 of 2
                                                            View in Use
CDS Name : ACTIVE

Enter Line Operators below:

     LINE       DATACLAS CA
     OPERATOR   NAME     RECLAIM
    ---(1)----  --(2)--- -(49)--
                DCCANO   NO
                DCCAYES  YES
```

Figure 4-5 DATA CLASS settings for CA Reclaim

The new field shows whether the option is enabled for data sets using the Data Class specified.

Attention: The default definition for the CA RECLAIM field in the Data Class panel is YES. However, to have that function work on the system, the CA Reclaim option must be set on the z/OS System. That function is disabled on system by default.

The CA reclaim attribute in data classes defaults to YES.

CA Reclaim is disabled for all KSDSs on the system when CA_RECLAIM(NONE) is specified in the IGDSMSxx parmlib member or a SETSMS CA_RECLAIM(NONE) command is issued and one of the following occurs:

- A system IPL occurs
- RLS VSAM address space is recycled
- A SET SMS=xx command is issued

CA reclaim setting

Use the console command D SMS, OPTIONS to check whether CA Reclaim is enabled on a system. Figure 4-6 shows the result if CA Reclaim is disabled (the default); Figure 4-7 shows the result if it is enabled.

```
D SMS,OPTIONS
IGD002I 13:32:12 DISPLAY SMS 852
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
.
.
.
.
CA_RECLAIM = NONE
PDSE_SYSEVENT_DONTSWAP = NO
IGD002I 13:32:12 DISPLAY SMS
```

Figure 4-6 CA Reclaim system default setting - disabled

```
D SMS,OPTIONS
IGD002I 13:35:07 DISPLAY SMS 857
ACDS     = SYS1.SMS.ACDS
COMMDS   = SYS1.SMS.COMMDS
.
.
.
.
CA_RECLAIM = DATACLAS
PDSE_SYSEVENT_DONTSWAP = NO
IGD002I 13:35:07 DISPLAY SMS
```

Figure 4-7 CA reclaim enabled by command or PARMLIB parameter

4.2.4 CA reclaim considerations

An example of a data set that should benefit greatly from CA Reclaim is one where many records are deleted. CA Reclaim provides for improved DASD space utilization, but it requires additional I/O to keep track of the reclaimed CAs so that they can be reused.

The CA Reclaim can be conveniently disabled for only specific data sets, if necessary. Simply issue an ALTER command followed by a VSAM CLOSE and a VSAM OPEN. The following data sets are examples of those that would not derive much benefit from using CA reclaim:

- Data sets with frequent erases but very few or no PUTs that cause CA splits. The overhead of maintaining the free CAs might exceed the faster index search.

- Data sets whose erased records are re-inserted to bring back the CAs that CA reclaim has labored to put on the free chains for reuse.

Classify KSDS

KSDSs can be classified into three groups with respect to CA reclaim:

- CAs are never completely erased. Using CA reclaim will have no impact.
- CAs are emptied; however, the similar record key ranges are re-inserted in a timely manner, reusing the CA. Using CA reclaim might have a slight performance impact.
- CAs are emptied and erased, and key ranges are not reinserted in a timely manner (if ever). Using CA reclaim will provide performance and space utilization improvements.

Most KSDSs will likely benefit from CA reclaim, but to give more control to applications, all KSDSs can be enabled or disabled on an individual basis, using the IDCAMS ALTER command.

Restrictions: CA Reclaim cannot reclaim space for the following:

- Partially empty CAs
- Empty CAs that already existed when CA reclaim was enabled
- CAs with RBA 0
- CAs with the lowest and highest keys of the KSDS
- Index CIs with the highest keys in each index level
- Data sets processed with global shared resources (GSR)
- Existing data sets with the IMBED or REPLICATE attributes

Note: Reclaimed space remains in the allocated space for the data set and the tracks are not released.

4.2.5 CA reclaim options reference

Table 4-1 presents a summary of the options for CA reclaim, based on system PARMLIB, DATA CLASS, and ALTER settings.

Table 4-1 CA reclaim system options

PARMLIB Parameter coded - after IPL	SETSMS CA_RECLAIM command issued	DATA CLASS CA RECLAIM (49)	ALTER CA reclaim command issued	CA reclaim final status
NONE	NONE	NO	NORECLAIMCA	NO
NONE	NONE	NO	RECLAIMCA	NO
NONE	NONE	YES	NORECLAIMCA	NO
NONE	NONE	YES	RECLAIMCA	NO
NONE	DATACLAS	NO	NORECLAIMCA	NO
NONE	DATACLAS	NO	RECLAIMCA	YES
NONE	DATACLAS	YES	NORECLAIMCA	NO
NONE	DATACLAS	YES	RECLAIMCA	YES
DATACLAS	NONE	NO	NORECLAIMCA	NO

PARMLIB Parameter coded - after IPL	SETSMS CA_RECLAIM command issued	DATA CLASS CA RECLAIM (49)	ALTER CA reclaim command issued	CA reclaim final status
DATACLAS	NONE	NO	RECLAIMCA	NO
DATACLAS	NONE	YES	NORECLAIMCA	NO
DATACLAS	NONE	YES	RECLAIMCA	NO
DATACLAS	DATACLAS	NO	NORECLAIMCA	NO
DATACLAS	DATACLAS	NO	RECLAIMCA	YES
DATACLAS	DATACLAS	YES	NORECLAIMCA	NO
DATACLAS	DATACLAS	YES	RECLAIMCA	YES
NONE	no command	NO	NORECLAIMCA	NO
NONE	no command	NO	RECLAIMCA	NO
NONE	no command	YES	NORECLAIMCA	NO
NONE	no command	YES	RECLAIMCA	NO
DATACLAS	no command	NO	NORECLAIMCA	NO
DATACLAS	no command	NO	RECLAIMCA	YES
DATACLAS	no command	YES	NORECLAIMCA	NO
DATACLAS	no command	YES	RECLAIMCA	YES

4.2.6 IDCAMS DATATEST

The IDCAMS DATATEST can help you get the statistics about CA utilization and reclaim.

DATATEST reads the sequence set from the index component and the entire data component of the KSDS cluster.

EXAMINE DATATEST output will indicate how fragmented the data set is by listing the current count of empty (un-reclaimed) CAs in message IDC01728I.

```
IDC01728I FOUND n EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED
```

After the reorganization of the data set, in a z/OS V1.12 version after CA reclaim is enabled, message IDC01728I will not appear anymore. That means that all empty control areas are available to be reclaimed.

4.2.7 Message IDC11724I

The existing EXAMINE DATATEST or DATATEST INDEXTEST message will continue to display CAs that have been orphaned, but it will be shown as a minor error (RC=4), not a major error (RC=8), because this message is only about DASD space not used, which does not cause data integrity problems.

```
IDC11724I DATA COMPONENT CA NOT KNOWN TO SEQUENCE SET
```

Tip: Some abnormal situations, such as ABENDs or logical errors, can cause a CA split or CA reclaim to be interrupted. A logical error results in VSAM backing out the CA split that is in progress and returning a nonzero return code to the PUT that caused the CA split. When a CA split is aborted in an abnormal condition, an unreachable new CA (commonly referred to as an orphan CA) can be left behind within the data set. This in itself does not cause a data integrity problem and the data set is fully functional and accessible.

However, if the EXAMINE also detects data integrity problems, it will also display other messages and display return code = 8 (and not 4). Because EXAMINE INDEXTEST DATATEST checks both the index and data components, its return code is more accurate in reflecting the true severity of the situations it detects than that of EXAMINE DATATEST NOINDEXTEST.

LISTCAT output

The IDCAMS LISTCAT output is the same regardless of whether CA Reclaim is active or not. The output is different only if you run the same job in a previous z/OS version.

On z/OS prior to V1.12, or z/OS V1.12 after a data set reorganization, with CA reclaim enabled, the information about empty CAs is not displayed (Figure 4-8).

```
INDEXTEST BEGINS
             879 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KEY LENGTH:     9.1
              36 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KEY LENGTH:     7.4
 CURRENT INDEX CISIZE IS  4096, RECOMMENDED MINIMUM INDEX CISIZE IS  3584
          6 DATA COMPONENT CIS ARE ESTIMATED TO BE UNREACHABLE
 INDEXTEST COMPLETE - NO ERRORS DETECTED
 DATATEST BEGINS
 DATATEST COMPLETE - NO ERRORS DETECTED
 879 CONTROL INTERVALS ENCOUNTERED
 DATA COMPONENT CONTAINS 18787 RECORDS
 DATA COMPONENT CONTAINS 30 DELETED CONTROL INTERVALS
 MAXIMUM LENGTH DATA RECORD CONTAINS 15616 BYTES
 92 PERCENT FREE SPACE
FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure 4-8 IDCAMS Examine SYSOUT with no Empty CAs to be reclaimed

IDCAMS VERIFY option

You can use IDCAMS VERIFY to verify a VSAM data set. When you issue this command, IDCAMS opens the VSAM data set for output, issues a VSAM VERIFY macro call, and closes the data set. The IDCAMS VERIFY and the verification by VSAM OPEN are the same. Neither changes the data in the verified data set.

The IDCAMS VERIFY command can be used to complete an interrupted CA reclaim. If a CA reclaim is interrupted and CA reclaim recovery is not performed, the data set is still accessible.

4.2.8 SMF recording

SMF type 64 records will count how many CAs have been reused by CA splits (not reclaimed) since the data set was defined.

VSAM and VSAM RLS will update SMF type 64 records showing how many CAs have been reused for a KSDS. The new field SMF64DAU was added to SMF 64 record in the Statistics, at OPEN Time (Example 4-1).

Example 4-1 SMF type 64 records related to CA reclaim

```
Offset Name Length Description
121 SMF64FD1 1 EOV DIAGNOSTIC FIELD 1
122 SMF64FD2 1 EOV DIAGNOSTIC FIELD 2
123 1 RESERVED
124 SMF64DAU 4 Change in the number of CA-reclaimed control areas
reused in the KSDS since the last EOV or CLOSE (32 bit integer)
128 4 RESERVED
```

For more information about the SMF record changes, see *z/OS MVS System Management Facilities (SMF), SA22-7630*.

4.2.9 Migration and coexistence considerations

The CA reclaim data set attribute is preserved across all supported releases during IMPORT/EXPORT processing.

KSDSs defined with the IMBED attribute will not be eligible for CA reclaim processing regardless of the CA reclaim data set attribute for the KSDS.

In a sysplex with various z/OS levels, the toleration PTFs must be applied to pre-z/OS V1.12 systems prior to CA reclaim activation. Failure to do will result in data set breakage.

The toleration APARs available now are:

- OA25108 (RLS related)
- OA26256 (IDCAMS related)
- OA26466 (VSAM related)
- OA27557 (SMS related)
- OA33397 (RLS related)

4.2.10 Practical scenarios

In this scenario we show examples of IDCAMS LISTCAT jobs and CA Reclaim after the function is enabled.

IDCAMS LISTCAT output for CA reclaim

LISTCAT output will indicate if CA reclaim is enabled for this data set (unless disabled by the system option) as shown in Figure 4-9.

```
SMSDATA
  STORAGECLASS ---SXPXXS02   MANAGEMENTCLASS---(NULL)
  DATACLASS ------KSCR000Y   LBACKUP ---0000.000.0000
  CA-RECLAIM---------(YES)   BWO STATUS------00000000     BWO TIMESTAMP---00000
00:00:00.0
  BWO---------------(NULL)
```

Figure 4-9 LISTCAT output showing the SMSDATA with CA-RECLAIM

IDCAMS EXAMINE DATATEST JCL example

IDCAMS EXAMINE DATATEST output will indicate how fragmented the data set is by listing current count of empty or un-reclaimed CAs:

```
IDC01728I FOUND n EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED
```

The IDCAMS EXAMINE JCL in Example 4-2 will help to visualize the improvements after the CA reclaim is enabled, and also to see if a reorganization is needed to recover more space.

Example 4-2 IDCAMS EXAMINE DATATEST JCL

```
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN    DD    *
  EXAMINE  -
       NAME(xxx.xxxxxx) -
       INDEXTEST         -
       DATATEST
```

IDCAMS VERIFY JCL example

The IDCAMS VERIFY job shown in Example 4-3 forces the VSAM to an OPEN and CLOSE macro call. After that, the ALTER command for CA Reclaim enabling or disabling is put in place.

Example 4-3 IDCAMS Verify JCL

```
//STEP2  EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN    DD    *
  VERIFY DATASET('MHLRES4.SMS.CAREYES.RLS01')
```

Comparative DATATEST jobs output

In this section we show examples related to the CA Reclaim work, using an RMM CDS data set as reference.

Figure 4-10 is the RMM CDS configuration used as input for the jobs in this scenario.

```
-----------------  VSAM KSDS Cluster utilization (ADDONS)      ----------------
COMMAND ===>                                                   SCROLL ===>

Cluster         ===> RMM.CONTROL.DSET                            SHROPTNS : 3,3
 Data           ===> RMM.CONTROL.DSET.DATA                       Threshold: 75
  REC-TOTAL     ===> 47492                SPLITS-CI   ===> 82
  REC-DELETED   ===> 4384                 SPLITS-CA   ===> 5
  REC-INSERTED  ===> 1787                 HI-A-RBA    ===> 479232000
  REC-UPDATED   ===> 51974                HI-U-RBA    ===> 39137280
  REC-RETRIEVED ===> 4233218              Percent used===> 8.16
  Space Type    ===> CYLINDER             Space Prim. ===> 600
  Space Sec.    ===> 60                   Extents     ===> 1
  Free Space    ===> 449067008            Percent used===> 6.29
 Index          ===> RMM.CONTROL.DSET.INDEX
  REC-TOTAL     ===> 50                   SPLITS-CI   ===> 4
  REC-DELETED   ===> 0                    SPLITS-CA   ===> 0
  REC-INSERTED  ===> 0                    HI-A-RBA    ===> 38707200
  REC-UPDATED   ===> 94                   HI-U-RBA    ===> 102400
  REC-RETRIEVED ===> 0                    Percent used===> 0.26
  Space Type    ===> CYLINDER             Space Prim. ===> 60
  Space Sec.    ===> 6                    Extents     ===> 1
  Free Space    ===> 38604800             Percent used===> 0.26
```

Figure 4-10 RMM CDS details

Next we present the output of DATATEST jobs for each scenario described. All jobs were executed on a z/OS V1.12 system that is a member of a sysplex with a z/OS V1.11 system with the coexistence PTFs applied. Refer to 4.2.9, “Migration and coexistence considerations” on page 49 for details about CA reclaim coexistence.

The initial basic characteristics of the DFSMSrmm control data set were listed using the LISTCAT online command. Figure 4-11 shows the CA-RECLAIM field set to YES, which will allow CA Reclaim for this data set.

```
CLUSTER ------- RMM.CONTROL.DSET
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2009.309
       RELEASE----------------2     EXPIRATION------0000.000
       CA-RECLAIM---------(YES)
       EATTR-------------(NULL)
       BWO STATUS--------(NULL)     BWO TIMESTAMP-----(NULL)
       BWO---------------(NULL)
     PROTECTION-PSWD-----(NULL)     RACF----------------(NO)
```

Figure 4-11 RMM CDS LISTCAT

Figure 4-12 on page 52 shows the result of the IDCAMS EXAMINE DATASET JCL shown in Example 4-2 on page 50. This was before any changes were made.

```
INDEXTEST BEGINS
            535 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KEY LENGTH:    12.3
             49 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KEY LENGTH:    10.4
 CURRENT INDEX CISIZE IS  2048, RECOMMENDED MINIMUM INDEX CISIZE IS  1024
 INDEXTEST COMPLETE - NO ERRORS DETECTED
 DATATEST BEGINS
 FOUND 25 EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED
 DATATEST COMPLETE - NO ERRORS DETECTED
 535 CONTROL INTERVALS ENCOUNTERED
 DATA COMPONENT CONTAINS 19779 RECORDS
 DATA COMPONENT CONTAINS 25 DELETED CONTROL INTERVALS
 MAXIMUM LENGTH DATA RECORD CONTAINS 1036 BYTES
 98 PERCENT FREE SPACE
FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure 4-12 IDCAMS examine before records insertion

For test purposes thousands of records were inserted into the RMM CDS and a new IDCAMS EXAMINE DATASET command was issued. Figure 4-13 shows the new statistics for the RMM CDS data set.

```
 INDEXTEST BEGINS
            826 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KEY LENGTH:    15.0
             64 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KEY LENGTH:    11.1
 CURRENT INDEX CISIZE IS  2048, RECOMMENDED MINIMUM INDEX CISIZE IS  1024
 INDEXTEST COMPLETE - NO ERRORS DETECTED
 DATATEST BEGINS
 FOUND 25 EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED
 DATATEST COMPLETE - NO ERRORS DETECTED
 826 CONTROL INTERVALS ENCOUNTERED
 DATA COMPONENT CONTAINS 28777 RECORDS
 DATA COMPONENT CONTAINS 25 DELETED CONTROL INTERVALS
 MAXIMUM LENGTH DATA RECORD CONTAINS 1036 BYTES
 97 PERCENT FREE SPACE
FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure 4-13 IDCAMS Examine SYSOUT after records insertion on RMM CDS

From the information in Figure 4-13 you can see there was an increase in the number of records; however, the number of empty control areas remained the same. This happens because those empty control areas existed before the CA reclaim feature was activated.

To analyze how the new empty CAs will be handled with CA Reclaim activated, we deleted some records to create more empty CAs. The difference is that those newly empty CAs are reclaimable, since they were created after the feature activation. Note the new statistics on empty control areas in Figure 4-14.

```
 INDEXTEST BEGINS
 DATA COMPONENT CA NOT KNOWN TO SEQUENCE SET
            649 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KEY LENGTH:    17.8
            162 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KEY LENGTH:    16.9
              2 KEYS PROCESSED ON INDEX LEVEL   3, AVERAGE KEY LENGTH:    25.5
 CURRENT INDEX CISIZE IS  2048, RECOMMENDED MINIMUM INDEX CISIZE IS  1024
 INDEX CONTROL INTERVAL COUNT ERROR
 165 CONTROL INTERVALS ENCOUNTERED
 HIGH-USED RBA/CI IS 74
 CONTROL INTERVAL SIZE IS 2048
 HIGH-USED INDEX RBA DOES NOT EQUAL HIGHEST RBA FOUND
 HIGH-USED RBA/CI IS 74
 RBA/CI IS 164
 MINOR ERRORS FOUND BY INDEXTEST
 DATATEST BEGINS
 FOUND 137 EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED
 DATATEST COMPLETE - NO ERRORS DETECTED
 649 CONTROL INTERVALS ENCOUNTERED
 DATA COMPONENT CONTAINS 19779 RECORDS
 DATA COMPONENT CONTAINS 137 DELETED CONTROL INTERVALS
MAXIMUM LENGTH DATA RECORD CONTAINS 1036 BYTES
 98 PERCENT FREE SPACE
 ERRORS MAY BE DUE TO CONCURRENT ACCESS
FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 4

IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 4
```

Figure 4-14 IDCAMS Examine SYSOUT after records deletion on a RMM CDS

As you can see in Figure 4-14, after the record deletion the number of keys decreased and there are 137 empty control areas; all but the previously identified 25 are eligible to be reclaimed using the CA reclaim function.

For test purposes, new records were added. With the VSAM data set opening and the record addition, the empty CAs will be reclaimed. Figure 4-15 on page 54 shows how this will look.

```
 INDEXTEST BEGINS
 DATA COMPONENT CA NOT KNOWN TO SEQUENCE SET
            3838 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KEY LENGTH:    18.8
             205 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KEY LENGTH:    17.1
               4 KEYS PROCESSED ON INDEX LEVEL   3, AVERAGE KEY LENGTH:    26.5
 CURRENT INDEX CISIZE IS  2048, RECOMMENDED MINIMUM INDEX CISIZE IS  1024
 INDEX CONTROL INTERVAL COUNT ERROR
 210 CONTROL INTERVALS ENCOUNTERED
 HIGH-USED RBA/CI IS 74
 CONTROL INTERVAL SIZE IS 2048
 HIGH-USED INDEX RBA DOES NOT EQUAL HIGHEST RBA FOUND
 HIGH-USED RBA/CI IS 74
 RBA/CI IS 209
 MINOR ERRORS FOUND BY INDEXTEST
 DATATEST BEGINS
 FOUND 25 EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED
 DATATEST COMPLETE - NO ERRORS DETECTED
 3838 CONTROL INTERVALS ENCOUNTERED
 DATA COMPONENT CONTAINS 131201 RECORDS
 DATA COMPONENT CONTAINS 25 DELETED CONTROL INTERVALS
DATA COMPONENT CONTAINS 25 DELETED CONTROL INTERVALS
 MAXIMUM LENGTH DATA RECORD CONTAINS 1036 BYTES
 85 PERCENT FREE SPACE
 ERRORS MAY BE DUE TO CONCURRENT ACCESS
FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 4

IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 4
```

Figure 4-15 IDCAMS Examine SYSOUT after record insertion in RMM CDS with CA reclaim enabled

As Figure 4-15 shows, the number of empty control areas is again 25. These 25 CAs that were not reclaimed existed before the CA Reclaim function was enabled.

To reclaim those specific empty CA records created before the CA reclaim function was enabled, a reorganization must done for this data set. After the reorganization all the control areas will be eligible for reclaim, since the CA reclaim is enabled on a system level.

Important: If a KSDS has a large number of empty control areas, then after CA reclaim is enabled on the system, the data set should be reorganized.

4.3 Disk striping support for VSAM record-level sharing

VSAM disk and data striping is not a new feature for VSAM data sets in z/OS V1.12; it became available in z/OS V1.3.

The enhancement in z/OS V1.12 is the ability to use VSAM record-level sharing (RLS) to access VSAM striped data sets. The previous restriction on accessing striped data sets from VSAM RLS is removed. All existing VSAM striping rules now apply to VSAM RLS.

4.3.1 VSAM RLS overview

VSAM record-level sharing is an access option for VSAM data sets that allows transactional applications (such as CICS), and non-transactional applications to concurrently access data. This option provides multisystem sharing of VSAM data sets across a z/OS Parallel Sysplex®. VSAM RLS exploits the data sharing technology of the coupling facility (CF), including a CF-based lock manager and a CF cache manager. VSAM RLS uses the CF-based lock manager and the CF cache manager in its implementation of record-level sharing.

> **Note:** VSAM RLS requires that the data sets be System Managed Storage (SMS) data sets. To be eligible for RLS, a data set that is not already SMS-managed must be converted to SMS.

RLS is an access option interpreted at open time. Select the option by specifying a JCL parameter (RLS) or by specifying MACRF=RLS in the ACB. The RLS MACRF option is mutually exclusive with the MACRF NSR (nonshared resources), LSR (local shared resources), and GSR (global shared resources) options.

Access method services do not use RLS when performing an IDCAMS EXPORT, IMPORT, PRINT, or REPRO command. If the RLS keyword is specified in the DD statement of a data set to be opened by access method services, the keyword is ignored and the data set is opened and accessed in non-RLS mode.

RLS access is supported for KSDS, ESDS, RRDS, and VRRDS data sets, and for VSAM alternate indexes. The VSAM RLS functions are provided by the SMSVSAM server. This server resides in a system address space. The address space is created and the server is started at IPL time.

For RLS access use no change to the format of the data in the VSAM data sets is required. The data sets are compatible for non-RLS access. A data set can be open for non-RLS write access or open for RLS access.

4.3.2 Comparing RLS access and non-RLS access

This section provides an overview of the differences between RLS access and non-RLS access.

Share options

For non-RLS access, VSAM uses the share options settings to determine the type of sharing permitted. If you set the cross-region share option to 2, a non-RLS open for input is permitted while the data set is already open for RLS access. VSAM provides full read and write integrity for the RLS users, but does not provide read integrity for the non-RLS user. A non-RLS open for output is not permitted when already opened for RLS access.

VSAM RLS provides full read and write sharing for multiple users; it does not use share options settings to determine levels of sharing. When an RLS open is requested and the data set is already open for non-RLS input, VSAM checks the cross-region setting. If it is 2, then the RLS open is permitted. The open fails for any other share option or if the data set has been opened for non-RLS output.

Locking

Non-RLS provides local locking (within the scope of a single buffer pool) of the VSAM control interval. Locking contention can result in an “exclusive control conflict” error response to a VSAM record management request. VSAM RLS uses a DFSMS lock manager to provide a system-managed duplexing rebuild process. The locking granularity is at the VSAM record level. When contention occurs on a VSAM record, the request that encountered the contention waits for the contention to be removed. The DFSMS lock manager provides deadlock detection. When a lock request is in deadlock, VSAM rejects the request. This results in the VSAM record management request completing with a deadlock error response. When you request a user-managed rebuild for a lock structure, the validity check function determines if there is enough space for the rebuild process to complete. If there is not enough space, the system rejects the request and displays an informational message. When you request an alter operation for a lock structure, the validity check function determines if there is enough space for the alter process to complete. If there is not enough space, the system displays a warning message that includes the size recommendation.

VSAM RLS uses share and exclusive record locks to control access to the shared data. An exclusive lock is used to ensure that a single user is updating a specific record. The exclusive lock causes any read-with-integrity request for the record by another user (CICS transaction or non-CICS application) to wait until the update is finished and the lock released.

RLS general requirements

To have a VSAM RLS access there are some rules that must be followed:

- The VSAM data set must be SMS managed.
- LOG parameter (NONE, UNDO or ALL) must be specified on the DEFINE CLUSTER or the ALTER CLUSTER command for the sphere. Or you can select one of the data classes containing such parameter.
- Option RLS in MACRF keyword in ACB must be set. Another way of doing this is using the keyword RLS= (NRI / CR) in the DD card defining the sphere.
- In general, you need to have the RLS system configurations set on the system.

For more details about VSAM RLS requirements and implementation, refer to *VSAM Demystified*, SG24-6105 and *CICS and VSAM Record Level Sharing: Implementation Guide*, SG24-4766.

4.3.3 VSAM data and disk striping

Usually, in a multi-extent, multi-volume VSAM data sets are processed in sequential access; processing does not allow for any type of parallelism for I/O operations among the volumes. This means that when an I/O operation is executed for an extent in a volume, no other I/O activity from the same task or same data set is scheduled to the other volumes. In a situation where I/O is the major bottleneck, and there are available resources in the channel subsystem and controllers, it is a waste of these resources.

Data striping addresses this sequential access performance problem by allowing parallel I/O operations for sequential stripes (CIs) on different volumes. The records are organized in stripes instead of key ranges along the volumes, which allows the concurrent access.

Performance is improved as a result, with data transfer into the application occurring at a rate greater than any single I/O path.

To have a VSAM data set allocated with striped data, the data set has to be SMS managed. In addition there are required Data class and Storage class parameters.

The DATA CLASS must have the DATA SET NAME TYPE field specified as EXTENDED REQUIRED, as shown in Figure 4-16.

```
 DATA CLASS LIST
Command ===>                                                  Scroll ===> CSR
                                                              Entries 1-1 of 1
                                                              View in Use
CDS Name : ACTIVE

Enter Line Operators below:

     LINE       DATACLAS
     OPERATOR   NAME     DATA SET NAME TYPE
    ---(1)----  --(2)--- -------(26)-------
                DCCAYES  EXTENDED REQUIRED
    ----------  -------- ------  BOTTOM  OF  DATA  ------  --------  ----------
```

Figure 4-16 DATA CLASS mandatory field for VSAM striping

The STORAGE CLASS must have a value specified for the field SUSTAINED DATA RATE as shown in Figure 4-17. The value specified is based on your system requirements.

```
 STORAGE CLASS LIST
Command ===>                                                  Scroll ===> HALF
                                                              Entries 1-1 of 1
                                                              View in Use
CDS Name : ACTIVE

Enter Line Operators below:

     LINE       STORCLAS SUSTAINED DATA
     OPERATOR   NAME     RATE (MB/SEC)
    ---(1)----  --(2)--- -----(15)-----
                WSTRIP2            230
    ----------  -------- ------  BOTTOM  OF  DATA  ------  --------  ----------
```

Figure 4-17 STORAGE CLASS mandatory field for VSAM striping

For more details about VSAM data sets refer to *DFSMS Using Data Sets,* SC26-7410.

4.3.4 Migration and coexistence considerations

There is no coexistence requirement or migration action to use this enhancement.

4.3.5 Practical scenario

Some tests were performed with VSAM RLS access to VSAM data striped and VSAM non-striped data sets.

An ASSEMBLER job was used to access the VSAMs using RLS parameters.

The job submitted on a z/OS V1.11 system failed to run due to the RLS restriction for VSAM striped data. The same job run on a z/OS V1.12 system completed successfully.

Figure 4-17 shows the VSAM data set RLS fields displayed using a LISTCAT command after the job was executed successfully.

```
CLUSTER ------- MHLRES4.SMS.CAREYES.RLS01
     IN-CAT --- UCAT.VSBOX01
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2010.286
RELEASE----------------2     EXPIRATION------0000.000
      SMSDATA
        STORAGECLASS ----WSTRIP2     MANAGEMENTCLASS---MCDB22
        DATACLASS -------DCCAYES     LBACKUP ---0000.000.0000
        CA-RECLAIM---------(YES)
        EATTR-------------(NULL)
        BWO STATUS------00000000     BWO TIMESTAMP---00000 00:00:00.0
        BWO---------------(NULL)
      RLSDATA
        LOG ------------------NONE   RECOVERY REQUIRED --(NO)     FRLOG
--------
----(NULL)
        VSAM QUIESCED -------(NO)    RLS IN USE --------(YES)
.
.
.
STRIPE-COUNT-----------3
        SHROPTNS(3,4)   RECOVERY     UNIQUE           NOERASE     INDEXED
NOWRITECHK     NOIMBED       NOREPLICAT
        UNORDERED        NOREUSE     NONSPANNED      EXTENDED     EXT-ADDR
```

Figure 4-18 LISTCAT of VSAM KSDS striped after be accessed using VSAM RLS

As shown in Figure 4-18, the RLS IN USE field now is shown as YES. That means that the last access to that VSAM data set was done in RLS mode.

The following VSAM characteristics are also shown:

- STORAGECLASS existence, which makes the data set System Managed Storage (SMS)
- LOG parameter defined as NONE, a requirement for RLS use
- STRIPE-COUNT, which is the confirmation about a VSAM striped data set

A striped data set has tracks that spread across multiple devices, as is the case for sequential access method or the CIs for VSAM. This format allows a single application request for records in multiple tracks or CIs to be satisfied by concurrent I/O requests to multiple volumes. The result is improved performance for sequential data access by achieving data transfer into the application at a rate greater than any single I/O path. The scheduling of I/O to multiple devices to satisfy a single application request is referred to as an I/O packet.

VSAM data striping applies only to data sets that are defined with more than one stripe. Any data set listed with one stripe is in extended format and is not considered to be a striped data set.

4.4 VSAM dynamic trace enhancements

In previous releases, the only way to enable VSAM record management trace was specifying the trace parameter in the JCL DD card.

```
AMP=('TRACE=(subparameters)')
```

Using the JCL AMP parameter it is not possible to include data sets that are allocated dynamically, such as catalog data sets. It also requires that you take down the application and open the data set before the trace can be enabled.

With this dynamic trace enhancement, it is possible to start a trace recording without needing to reopen the VSAM data set with specific JCL code. You set the parameters that you want using a member in PARMLIB, and just enable that specific trace dynamically.

Attention: The trace functions are typically enabled for a specific problem or performance problem for a component or data set. Dynamic trace should only be used in cases where a better analysis is needed. It must be turned off when not in use.

4.4.1 Dynamic trace description

Dynamic trace improves VSAM serviceability, extending the usability of VSAM record management trace supporting data sets that are allocated dynamically. New trace functions also were added to provide more information in trace results.

To support this dynamic trace enhancement two new members were created:

- An entry on PARMLIB - the IDAVDTxx member. The trace details must be coded in the IDAVDTxx member, providing the data set that is going to be traced, the job name, and trace options to be used.
- An entry on PROCLIB - the IDAVDT member, to allow the start of dynamic trace.

Also a LINKLIB entry with the member IDAVDTRA is provided. The LINKLIB entry works with the PROCLIB entry to enable the started task.

Before any trace activation you must verify that the Generalized Trace Facility (GTF) task is active on the system and that you have the buffer set correctly for the trace to be collected. To check the GTF task status issue the command TRACE STATUS on the system console. The results are shown in Figure 4-19.

```
RESPONSE=SC64
 IEE839I ST=(ON,0001M,00006M) AS=ON  BR=OFF EX=ON  MO=OFF MT=(ON,064K)
         ISSUE DISPLAY TRACE CMD FOR SYSTEM AND COMPONENT TRACE STATUS
         ISSUE DISPLAY TRACE,TT CMD FOR TRANSACTION TRACE STATUS
```

Figure 4-19 Trace status command output

You can verify from this output that the trace option is active on the system (ST=ON).

It is recommended that you have the reserved common storage buffer size set to at least 1024 K. The default is 40 K, so that must be changed by your system support.

More details about GTF can be found in *MVS System Commands,* SA22-7627.

The trace verifications we have discussed are default for any kind of VSAM trace activation. For dynamic trace use, also ensure that the PROCLIB member IDAVDT and PARMLIB member IDAVDT00 are updated in your PARMLIB and PROCLIB concatenation.

4.4.2 PARMLIB MEMBER (IDAVDTxx)

In the IDAVDTxx PARMLIB member a maximum of eight VSAM record management trace entries can be defined.

Each trace entry has to have the target data set name, the target job name, and the usual trace parameters such as Hook, PARM1, PARM2, and so on.

The trace parameters are the same as for batch trace.

For IDAVDTxx PARMLIB member the following parameters must be coded:

VTRACE	Specifies the current entry is to define a VSAM record management trace entry.
DSNAME	**(data_set_name)** - Specifies which data set the user would like to enable VSAM record management trace.
JOBNAME	**(job_name)** - Specifies the job where the targeted data set was opened or will be opened.
HOOK	**(hook_id [,hook_id])** - Specifies the list of VSAM predefined Hook points where the trace should execute during VSAM record management processing. This statement consists of 1 to 3 other statements (SLIP, DUMP or ABEND). This is an optional parameter, the default is HOOK=(1).
PARM1	**(trace_parm_1)** - Specifies which control blocks VSAM record management trace should capture.
PARM2	**(trace_parm_2**) - Specifies which control blocks VSAM record management trace should capture when tracing the AIX®.This is an optional parameter.
ECODE	(**ANY** | **rsncode)** - Specifies that tracing should occur only if an error code is being returned to the caller. **ANY** - Tracing is performed for any nonzero return code. **rsncode** - Tracing is preformed only if the RPLFDBK code matches the rsncode.
KEYVALUE	(target-key-value) | KEYVALUE(x'target_key_value_hex')].
END	Specifies the end of trace statement.

For descriptions and more details about IDAVDTxx member parameters refer to *DFSMSdfp Diagnosis*, GY27-7618.

Note: The default member name for Dynamic trace in PARMLIB is IDAVDT00.

4.4.3 IDAVDTxx PARMLIB member example

Example 4-4 shows how the IDAVDTxx member will look after customization.

Example 4-4 IDAVDTxx PARMLIB member

```
VTRACE  DSNAME(xxxxxxx.xxxx.xxxxx)
        JOBNAME(XXXXXXX)
        HOOK(0,1)
        PARM1(100101000000)
        KEYVALUE(C1C2C3C4C5C6)
END
```

4.4.4 IDAVDTxx PROCLIB member

The IDAVDT PROCLIB member is provided with z/OS base code. This PROCLIB member is associated with the IDAVDT task.

An example of IDAVDT in SYS1.PROCLIB is shown in Example 4-5.

Example 4-5 IDAVDT PROCLIB member

```
// IDAVDT PROC
// VSAMTRAC EXEC PGM=IDAVDTRA
```

4.4.5 Operation commands for trace

After the trace entries are defined, you can invoke the new started task, IDAVDT, through the START console command to enable VSAM record management trace. Once you start IDAVDT, you can interact with IDAVDT through the MODIFY command to invoke various functions.

Each trace entry will consist of the target data set name, the target job name, and the usual trace parameters. After you start IDAVDT, you can interact with IDAVDT using the MODIFY command to perform the following functions for VSAM dynamic trace parameters:

READIN Parse the trace entries in the specified IDAVDTxx parmlib member and insert it into the dynamic trace save area located in common storage.

ENABLE Enable a specific trace entry that is stored in the dynamic trace area.

DISABLE Disable a specific trace entry that is stored in the dynamic trace area.

DISPLAY Display a specific trace entry that is stored in the dynamic trace area.

VALIDATE Validate a specific trace entry, or all trace entries, that are stored in the dynamic trace area.

INVALIDATE Invalidate a specific trace entry, or all trace entries, that are stored in the dynamic trace area.

Note: You can predefine up to the maximum of eight trace entries with different trace parameters. Each trace entry will consist of the target data set name, the target job name, and the usual trace parameters. When the trace entries are defined, you can use the START console command to call IDAVDT to enable VSAM record management trace.

4.4.6 Dynamic Trace activation

The trace can be started per batch job or by issuing the console command:

```
START IDAVDT
```

After the initialization of Dynamic Trace the system will receive the WTO message shown in Figure 4-20 confirming that dynamic trace initialization was successful.

```
IDAT0001I VSAM DYNAMIC TRACE WAS INITIALIZED SUCCESSFULLY
```

Figure 4-20 WTO message after Dynamic Trace initialized

Once the initialization completes, the user can begin to issue the modify commands to invoke various VSAM Dynamic Trace functions.

Catalog trace

For catalogs already in use in the Catalog address space, you must issue the CLOSE catalog command shown in Figure 4-21 to allow the trace to start.

```
F CATALOG,CLOSE(catalogname)
```

Figure 4-21 CLOSE command for Catalogs

The CLOSE command will close the catalog. When the catalog is re-opened the trace will be active.

Dynamic Trace configuration activation

After Dynamic Trace is initialized you can use the F IDAVDT,READIN command to interact with IDAVDT (Figure 4-22). The READIN parameter causes the trace entries in the IDAVDT00 member of parmlib to be inserted into the dynamic trace save area located in common storage. Each trace entry consists of the target data set name, the target job name, and the usual trace parameters.

The READIN parameter parses the trace entries in the specified IDAVDTxx parmlib member andinsert them into the dynamic trace save area located in common storage.

```
F IDAVDT,READIN
IDAT0002I THE FOLLOWING TRACE ENTRIES WERE STORED SUCCESSFULLY
1 MHLRES4.SMS.CAREYES.TESTT                    MHLRES4T V
```

Figure 4-22 Display of acceptance of Modify READIN command

4.4.7 Dynamic trace deactivation

The tracing of a data set is terminated when the data set is closed. If you need to stop the dynamic trace function before the data set is closed, this can be done for all dynamic traces or just for a specific trace.

To stop a specific trace, you must issue the command F IDAVDT,DISABLE=n, where n is the trace entry for the specific data set you do not want to trace anymore.

To stop the IDAVDT started task issue the STOP IDAVDT command. It is suggested that you issue the command every time you end a trace to avoid keeping that task in memory.

4.4.8 Dynamic trace quick checklist to start the function

Use the following steps to exploit the dynamic trace function:

1. Verify the Trace Master task is active on the system and the buffer size is adequate for your requirements.
2. Verify the existence of the IDAVDT PROCLIB member and the IDAVDT00 PARMLIB member on your system PROCLIB and PARMLIB concatenations.
3. Customize the IDAVDTxx PARMLIB member with the desired trace details.
4. Include these new PROCLIB and PARMLIB members in your working library.
5. Copy the IDAVDTxx parmlib member from the data set pointed to by your parmlib DD definition to the system's working parmlib data set.
6. Copy the IDAVDT PROCLIB member from the data set pointed to by your PROCLIB DD definition to the system's working PROCLIB data set.
7. Invoke the new started task IDAVDT using the START console command.

4.4.9 Migration and coexistence considerations

There is no coexistence requirement or migration action to use this enhancement.

5

DFSMSdss enhancements in z/OS V1.12

In this chapter we provide details about the enhancements to DFSMSdss provided in z/OS V1.12.

5.1 Using Basic Sequential Access Method I/O

Starting with z/OS V1R12, DFSMSdss now uses Basic Sequential Access Method (BSAM) instead of EXCP to read from and write to DFSMSdss dump data sets during DUMP, COPYDUMP, and RESTORE operations. DFSMSdss is designed to use larger blocks when possible for DUMP, COPYDUMP, and RESTORE operations, and to support Extended Format Sequential dump data sets on DASD for DUMP, RESTORE, and COPYDUMP. The use of larger block sizes improves performance for these operations, and using Extended Format dump data sets supports striping and compression. This allows DFSMSdss to support 256 K blocks when writing to and reading from a tape. Before z/OS V1R12, the maximum was 65,520 bytes.

5.1.1 Enhanced BLKSIZE DD

The DUMP and COPYDUMP commands require a DD to be passed using the OUTDDNAME keyword that identifies a basic or large format sequential data set. DFSMSdss uses this data set to write the output blocks using EXCP during the DUMP and COPYDUMP. Users are not required to specify the BLKSIZE parameter on this output DD, but when the BLKSIZE is specified it currently only supports a limited set of the overall valid values for a tape device.

Currently, users can specify a block size using the BLKSIZE parameter of the DD with either a value of zero or a value in the range of 7892 - 32760 bytes for both DASD and tape. DFSMSdss would ignore any block size over 32760 bytes and behaves as though no block size was specified.

OUTDDNAME specifying a DASD data set

The maximum supported block size for DASD is 32,760 bytes. DFSMSdss DUMP and COPYDUMP already support the entire range of valid block sizes when the output is DASD.

OUTDDNAME specifying a tape data set

When the output of a DUMP or COPYDUMP is directed to a tape device and either BLKSIZE=0 was coded on the output DD, or the BLKSIZE parameter was left off, then DFSMSdss currently chooses 65,520 bytes as the default block size. The DUMP and COPYDUMP command will increase the default block size from 65,520 bytes to a maximum of 262,144 bytes when the output is to a tape device. DFSMSdss will choose the optimum block size for the particular tape device if BLKSZLIM was not specified.

Currently a user could specify a BLKSIZE greater than 262,144 bytes and DFSMSdss would not fail the operation. Instead, DFSMSdss just chooses to write 65,520 byte blocks and create the backup. However, there is no indication that the block size specified was not honored. The message ADR030I will now provide the user with that information but DFSMSdss will not change this default behavior.

Messages

Example 5-1 shows messages generated if DFSMSdss has not honored the BLKSIZE specified.

Example 5-1 ADR030I message and examples

```
ADR030I DCB VALUES HAVE BEEN MODIFIED FOR ddname.  [BLKSIZE | LRECL] VALUE
         MODIFIED FROM nnnnnnnn1 TO nnnnnnnn2

ADR030I (001)-OPNCL(01), DCB VALUES HAVE BEEN MODIFIED FOR DASD. BLKSIZE VALUE
MODIFIED FROM 62145 TO 32760
```

```
ADR030I (001)-OPNCL(02), DCB VALUES HAVE BEEN MODIFIED FOR TAPE. BLKSIZE VALUE
MODIFIED FROM 262154 TO 262144
```

5.2 EXCP fall back support

The existing EXCP support to read and write data from and to a dump data set will be left in the product and will be able to be activated as a fall back method. DFSMSdss will add a new EXEC PARM value, 'USEEXCP=YES | NO' for JCL users and a new bit (UFOPEXCP) in the ADRUFO data area for storage administrators and application programs. The default is 'USEEXCP=NO' or UFOPEXCP=OFF. The new parameter will be supported for DUMP output, RESTORE input and COPYDUMP operations. If the new PARM is specified on a command other than DUMP, COPYDUMP or RESTORE it will be ignored.

When there are requests where DFSMSdss uses the old method of EXCP during DUMP, COPYDUMP, or RESTORE, and the dump data set is Extended Format, DFSMSdss will ABEND when trying to OPEN the DCB and the task will fail with a return code of 16. An existing OPEN error will be issued as shown in Example 5-2:

Example 5-2 IEC143I message example

```
IEC143I
213-B8,IFG0194D,MHLRES3A,STEP1,DDUMPOUT,047C,VT0127,MHLRES3.DUMP
```

The B8 return code means that an OPEN was attempted against an Extended Format data set with a DCB that specified EXCP. EXCP is not supported for Extended Format data sets.

5.2.1 Fall back via JCL

Using the USEEXCP parameter, you are able to specify whether you would like to fall back to using EXCP. USEEXCP specifies whether the access method used by DFSMSdss for DUMP output, RESTORE input, and COPYDUMP operations is to be EXCP. If the backup is to or from tape, the default is NO. If the backup is to or from DASD, the default is YES, unless the backup data set is in the Extended Format. Example 5-3 contains sample JCL for using the USEEXCP parameter.

When USEEXCP=YES is specified, DFSMSdss will fall back to the traditional method of using EXCP to write blocks to the output dump data set during DUMP commands, read blocks from the input dump data set during RESTORE commands, and read and write blocks from the input and output data set during COPYDUMP commands.

When PARM='USEEXCP=YES | NO' is specified, a new message ADR054I is issued to display the PARM value specified. Any value other than YES|NO will cause DFSMSdss to issue existing message ADR009T and the job ends with a return code 12.

When PARM='USEEXCP=NO' is specified, or defaulted, DFSMSdss will use BSAM to write blocks to the output dump data set during DUMP commands, read blocks from the input dump data set during RESTORE commands, and read and write blocks from the input and output dump data sets during COPYDUMP commands.

Example 5-3 Example of JCL to specify USEEXCP parameter

```
//STEP1    EXEC  PGM=ADRDSSU,PARM='USEEXCP=YES'
//SYSPRINT DD    SYSOUT=*
```

```
//DASD1    DD    UNIT=3390,VOL=(PRIVATE,SER=MLDE65),DISP=OLD
//TAPE     DD    UNIT=VT3590,
//  LABEL=(1,SL),DISP=(NEW,CATLG),DSNAME=USER.BACKUP1
//SYSIN    DD    *
     DUMP  OUTDDNAME(TAPE) -
        DS(INCLUDE(TEST.BIGSEQ))
/*
```

In the JOBLOG, message ADR054I will appear, stating that EXCP will be used as shown in Example 5-4.

Example 5-4 An example of message ADR054I

```
ADR054I (SCH)-PRIME(01), USE EXCP FOR DUMP OUTPUT, RESTORE INPUT AND COPYDUMP YES
REQUESTED
```

5.2.2 Fall back via the Options installation exit

The Options Installation Exit Routine (ADRUIXIT) is given control when DFSMSdss is invoked and also before the processing of each task. ADRUIXIT can override options, defaults, or values specified by the user JCL. UFOPARM contains bits that can be set to change defaults or override commands.

A new flag UFOUEXCP will be defined in the UFOPARM DSECT. UFOUEXCP=ON indicates DFSMSdss will use the existing EXCP support when writing to the output dump data set during a DUMP command, reading from the input dump data set during a RESTORE command, and reading and writing the input and output dump data sets during COPYDUMP.

When a storage administrator deactivates the BSAM I/O support by setting UFOUEXCP to ON in the User Options Installation Exit Notification Routine (ADRUIXIT), existing message ADR035I will be issued indicating the new PARM value being used.

Refer to *z/OS V1R12 DFSMS Installation Exits*, SA22-7593 in topic 8.5.1.2 ADRUFO Parameter List for further details.

5.2.3 Fall Back via application program

Application programs will also be able to deploy the fall back either through the use of a new bit (UFOPEXCP) in the ADRUFO data area during EXIT 13 (presenting ADRUFO record) or by passing the address of an area containing the new 'USEEXCP=YES|NO' EXEC PARM in the OPTPTR when attaching or linking DFSMSdss.

5.3 Stand-alone restore

Use the DFSMSdss BUILDSA function to create the Stand-Alone Services IPL-capable core image. If this migration action is not performed, users of the DFSMSdss stand-alone restore will not be able to restore backups on tape created with greater than 65,520 byte blocks. Message ADRY3530I SEQUENCE ERROR ON RESTORE TAPE is issued and the operation is terminated. Backups created with 65,520 byte blocks will restore as they did in z/OS V1R11.

Coexistence

Tapes with greater than 65,520 bytes BLKSIZE can be read by z/OS V1R11 and z/OS V1R10 systems, with the PTFs for APAR OA30822 installed.

5.4 Removal of IMBED, REPLICATE, and KEYRANGE

The creation of new VSAM data sets with IMBED, REPLICATE, and KEYRANGE attributes has been unsupported since z/OS V1.6. These attributes, originally introduced to improve performance on older DASD, typically act only to occupy additional space and slow performance on modern cached DASD. In z/OS V1.12, the system removes these attributes automatically from VSAM data sets logically dumped using DFSMSdss and migrated using DFSMShsm when DFSMSdss is used as the data mover during restore and recall processing. An informational message ADR507W will be issued to confirm that newly restored data sets no longer retain these attributes, as shown in Example 5-5.

Example 5-5 ADR507W message

```
ADR507W (ttt)-mmmmm(yy), DATA SET dsn WAS RESTORED WITHOUT THE IMBED OR REPLICATE
ATTRIBUTES
```

Note: Only DFSMSdss data set dump and restore processing will be changed to remove these attributes. DFSMSdss is not changing the copy function.

6

System Managed Storage enhancements

In this chapter we provide information about the SMS enhancements in z/OS V1.12, including:

- SMS Health Check
- Storage Group and volume selection
- Separate VOLSELMSG Trace parameters

6.1 z/OS V1R12 DFSMSdfp features

z/OS V1.12 enhances the z/OS Health Check function with two new Health Checks specific to the SMS subsystem. These new Health Checks will check two best practices for CDS allocation that will improve performance and availability when the suggested settings are implemented.

z/OS V1.12 DFSMS also provides an update for the Storage Group copy function, allowing volume definitions to be copied across a new Storage Group being defined. This enhancement provides the ability to easily copy Storage Groups.

VOLSELMSG and Trace parameters were modified to allow individual settings in z/OS 1.12 so that different trigger criteria can to be set for each of them.

6.2 DFSMS SMS health check enhancement

The objective of IBM Health Checker for z/OS is to identify potential problems before they affect system availability or, in worst cases, cause outages. It checks the current active z/OS and sysplex settings and definitions for a system and compares the values to those suggested by IBM or defined by you. It is not meant to be a diagnostic or monitoring tool, but rather a continuously running preventative that finds potential problems. IBM Health Checker for z/OS produces output in the form of detailed messages to let you know of both potential problems and suggested actions to take. Note that these messages do not mean that IBM Health Checker for z/OS has found problems that you need to report to IBM. IBM Health Checker for z/OS output messages simply inform you of potential problems so that you can take action on your installation.

For more information about Health Check MVS procedures refer to *IBM Health Checker for z/OS V1R12.0 User's Guide,* SA22-7994

6.2.1 SMS Health Check description

Prior to z/OS V1.12 there was no health check function for the SMS subsystem. The SMS Health Check performs routine check-ups for installation configurations, reports potential problems, and recommends solutions. It checks for, and provides, two important pieces of information about SMS control data sets. Specifically, SMS Health Check:

- Verifies that the ACDS and COMMDS are not allocated on the same volume.

  Having the ACDS and COMMDS on different volumes prevents a possible single point of failure and eases the complexity of recovery in case of a failure. If the ACDS and COMMDS are on the same volume, the Health Check will warn the system programmer about the discovery and recommend a reallocation of the data sets on different volumes.

- Verifies that both the ACDS and COMMDS have the REUSE option.

  Having the ACDS and COMMDS with the REUSE parameter coded prevents the SMS running into space problems (SMS reason code 6068) as a result of subsequent ACDS updates or IMPORT/EXPORT functions. If the ACDS or COMMDS does not have the REUSE attribute, the health check will warn the system programmer and recommend changing the data set.

6.2.2 SMS Health Check definitions

By default both SMS Health Check policies come with a severity setting of MEDIUM, but the system programmer can change that, if desired, to LOW or HIGH priority.

ACDS and COMMDS not allocated on the same volume

The format of the policy statement to enable the Health Check for the ACDS and COMMDS on the same volume is shown in Example 6-1. It has to be coded in a policy statement on HZSPRMxx parmlib or updated on the system with a MODIFY command.

Example 6-1 SMS ACDS and COMMDS on same volume

```
UPDATE CHECK(IBMSMS,SMS_CDS_SEPARATE_VOLUMES)
SEVERITY(MED)
DATE(20101018)
REASON('Prevent single point of failure for ACDS and COMMDS')
```

ACDS and COMMDS with the REUSE parameter

The format of the policy statement to enable the Health Check for the ACDS and COMMDS with the REUSE parameter is shown in Example 6-2. It has to be coded in a policy statement on HZSPRMxx parmlib or updated on system with a MODIFY command.

Example 6-2 ACDS and COMMDS coded

```
UPDATE CHECK(IBMSMS,SMS_CDS_REUSE_OPTION)
SEVERITY(MED)
DATE(20101018)
REASON('Avoid ACDS or COMMDS running into space problems.')
```

Issuing a MODIFY command for the Health Check task with the parameters shown, you can change the values in brackets with the desired parameters for Severity, Date, and Reason.

6.2.3 SMS Health Check function from SDSF

To verify whether the SMS policy is active on your system, from the SDSF menu select the CK option to list Health Check polices, as shown in Figure 6-1 on page 74.

```
HQX7770 -----------------  SDSF PRIMARY OPTION MENU  --------------------------
COMMAND INPUT ===> CK                                          SCROLL ===> CSR

DA    Active users                      INIT  Initiators
I     Input queue                       PR    Printers
O     Output queue                      PUN   Punches
H     Held output queue                 RDR   Readers
ST    Status of jobs                    LINE  Lines
                                        NODE  Nodes
LOG   System log                        SO    Spool offload
SR    System requests                   SP    Spool volumes
MAS   Members in the MAS
JC    Job classes                       RM    Resource monitor
SE    Scheduling environments           CK    Health checker
RES   WLM resources
ENC   Enclaves                          ULOG  User session log
PS    Processes

END   Exit SDSF
```

Figure 6-1 Example of a SDSF panel with the Health Checker

The SDSF HEALTH CHECKER DISPLAY is shown. In the NAME field locate the SMS policies:

- SMS_CDS_REUSE_OPTION
- SMS_CDS_SEPARATE_VOLUMES

Those two policies are delivered by default with Health Check code for z/OS V1.12. After you have the Health Check enabled for z/OS, the two default Health Checks will be available and in place. You can see the SMS Health Check options in the SDSF output shown in Figure 6-2.

```
SDSF HEALTH CHECKER DISPLAY  SC64                      LINE 93-110 (157)
COMMAND INPUT ===>                                            SCROLL ===>
PREFIX=*  DEST=(ALL)  OWNER=*  SYSNAME=
NP   NAME                           CheckOwner       State
     RSM_RSU                        IBMRSM           ACTIVE(ENABLED)
     RTM_IEAVTRML                   IBMRTM           ACTIVE(ENABLED)
S    SMS_CDS_REUSE_OPTION           IBMSMS           ACTIVE(ENABLED)
     SMS_CDS_SEPARATE_VOLUMES       IBMSMS           ACTIVE(ENABLED)
     SDSF_ISFPARMS_IN_USE           IBMSDSF          ACTIVE(ENABLED)
```

Figure 6-2 SDSF Health Check view for SMS Health Checks

To identify whether there is an exception to your SMS allocation, and consequently an action required, select the policy you want to check.

The next SDSF panel provides information regarding the SMS Health Check policy you selected.

Using the same volume

When the ACDS and COMMDS reside on the same volume, Health Checker issues the message shown in Figure 6-3.

```
HZS0002E CHECK(IBMSMS,SMS_CDS_SEPARATE_VOLUMES):  872
IGDH1001E CHECK(IBMSMS,SMS_CDS_SEPARATE_VOLUMES) DETECTED THE
ACDS (SYS1.STPPLEX.ACDS)
AND COMMDS (SYS1.STPPLEX.COMMDS)
ALLOCATED ON THE SAME VOLUME.

Explanation: As a best practice, an ACDS/COMMDS must reside on a volume,
accessible from all systems in the SMS complex. To ease recovery in case of
failure, the ACDS should reside on a different volume than the COMMDS. Also,
you should allocate a spare ACDS on a different volume. The control data set
(ACDS or COMMDS) must reside on a volume that is not reserved by other systems
for a long period of time because the control data set (ACDS or COMMDS) must be
available to access for SMS processing to continue.

System Programmer Response: Reallocate ACDS and COMMDS on different
  volumes.
```

Figure 6-3 Health Checker message with CDSs on the same volume

The message provides an explanation about the error and shows what has to be done to fix the problem in the System Programmer Response paragraph.

In this case, the recommended response is to allocate the ACDS and COMMDS on separate volumes.

REUSE option

When the ACDS and COMMDS are not defined with the REUSE option Health Checker issues the message shown in Figure 6-4. The message accessed through the SDSF panels shows the problem with a brief explanation and also explains what has to be done to correct the error in the System Programmer Response paragraph.

In this case, an ALTER is required to enable the REUSE option for the SMS data sets.

```
HZS0002E CHECK(IBMSMS,SMS_CDS_REUSE_OPTION):  873
IGDH1011E CHECK(IBMSMS,SMS_CDS_REUSE_OPTION) DETECTED
ACDS (SYS1.STPPLEX.ACDS) AND COMMDS (SYS1.STPPLEX.COMMDS)
NOT DEFINED WITH THE REUSE OPTION.

Explanation: As a best practice, defining ACDS/COMMDS with the REUSE option
helps to avoid running into space problems (SMS reason code 6068) as result of
subsequent ACDS updates, or IMPORT/EXPORT functions.

System Programmer Response: Use ALTER command in IDCAMS to specify
  the REUSE option for the control dataset.  See DFSMS Access Method
  Services for Catalogs for further details.
```

Figure 6-4 SDSF panel of Health Check message for CDS defined without the REUSE option

For more information about SMS control data sets, how to proceed with the changes documented by the Health Check, and best practices for allocation, refer to the *DFSMSdfp Storage Administration Reference,* SC26-7402.

SMS Health Check examples

Figure 6-5 shows an IDCAMS listing of the ACDS and COMMDS not defined with the REUSE option and defined on the same volume.

```
CLUSTER ------- SYS1.STPPLEX.ACDS
     IN-CAT --- MCAT.BH5CAT
ASSOCIATIONS
  CLUSTER--SYS1.STPPLEX.ACDS
ATTRIBUTES
  KEYLEN-----------------0     AVGLRECL---------------0     BUFSPACE-------
SHROPTNS(3,3)   RECOVERY     UNIQUE           NOERASE     LINEAR        N
  UNORDERED        NOREUSE     NONSPANNED
VOLUME
  VOLSER------------STPSY2     PHYREC-SIZE---------4096     HI-A-RBA-------

CLUSTER ------- SYS1.STPPLEX.COMMDS
     IN-CAT --- MCAT.BH5CAT
   ASSOCIATIONS
     CLUSTER--SYS1.STPPLEX.COMMDS
   ATTRIBUTES
     KEYLEN-----------------0     AVGLRECL---------------0     BUFSPACE-------
SHROPTNS(3,3)   RECOVERY     UNIQUE           NOERASE     LINEAR        N
     UNORDERED        NOREUSE     NONSPANNED
VOLUME
  VOLSER------------STPSY2     PHYREC-SIZE---------4096     HI-A-RBA-------
```

Figure 6-5 Listing of IDCAMS ACDS and COMMDS

You can see in Figure 6-5 that both data sets are defined on volume STPSY2, which is not good practice, and they were defined with the NOREUSE parameter. To have a more reliable SMS configuration it is important to change these two attributes.

6.2.4 SMS Health Check commands

If the system does not have the Health Checker active at IPL time, and a verification is required for those SMS policies, a start Health Checker command should be issued. Use the following commands to manage health checks:

- To start the Health Check address space (if it is already configured on the system):

  ```
  S HZSPROC
  ```

- To run all SMS-related Health Checks:

  ```
  F hzsproc,RUN,CHECK=(ibmsms,*)
  ```

- To run SMS REUSE OPTION Health Check:

  ```
  F hzsproc,RUN,CHECK=(ibmsms,sms_cds_reuse_option)
  ```

- To run SMS SEPARATE VOLUMES Health Check:

  ```
  F hzsproc,RUN,CHECK=(ibmsms,sms_cds_separate_volumes)
  ```

6.2.5 Migration and coexistence considerations

There is no coexistence requirement or migration action to use this SMS Health Check enhancement.

6.3 Copy storage group volumes

To simplify the creation of new SMS classes, storage groups, aggregate groups, and SCDSs, you can copy existing ones and modify them.

Before z/OS V1.12 storage administrators could copy storage group definitions from one control data set to another, but the volumes defined in the storage groups could not be copied. The storage administrator would have to manually add volumes to the storage groups.

With z/OS V1.12, to simplify this process, a new option on ISMF under the Copy function for Pool Type Storage Groups is provided. This new option lets you indicate whether the volume list of the source storage group should also be copied or added to the target storage group.

This new feature is particularly useful when you need to copy the Storage Group and volume definitions from one CDS to another CDS. When doing the copy for the same CDS, the message about duplicate volumes will be shown and only the Storage Group definition will be copied. This is the same way that the older z/OS versions work.

6.3.1 Copying the Storage Group with volume information

When copying a Pool Storage Group definition from one CDS to another, you can now specify that the volume list in the source Storage Group be copied into the target Storage Group. A new “Copy Storage Group Volumes” field is added to the ISMF Copy Entry panel, under the Construct Name heading. If you select this field, SMS copies the volume list to the new Storage Group name and adds the volumes to the target CDS.

To perform a Storage Group copy, select option 6 Storage Group from the ISMF Primary Option Menu shown on Figure 6-6 on page 77.

```
ISMF PRIMARY OPTION MENU - z/OS DFSMS V1 R12
Selection or Command ===>  6

0  ISMF Profile              - Specify ISMF User Profile
1  Data Set                  - Perform Functions Against Data Sets
2  Volume                    - Perform Functions Against Volumes
3  Management Class          - Specify Data Set Backup and Migration Criteria
4  Data Class                - Specify Data Set Allocation Parameters
5  Storage Class             - Specify Data Set Performance and Availability
6  Storage Group             - Specify Volume Names and Free Space Thresholds
7  Automatic Class Selection - Specify ACS Routines and Test Criteria
8  Control Data Set          - Specify System Names and Default Criteria
9  Aggregate Group           - Specify Data Set Recovery Parameters
10 Library Management        - Specify Library and Drive Configurations
11 Enhanced ACS Management   - Perform Enhanced Test/Configuration Management
C  Data Collection           - Process Data Collection Function
G  Report Generation         - Create Storage Management Reports
L  List                      - Perform Functions Against Saved ISMF Lists
P  Copy Pool                 - Specify Pool Storage Groups for Copies
R  Removable Media Manager   - Perform Functions Against Removable Media
X  Exit                      - Terminate ISMF
```

Figure 6-6 ISMF initial panel

The next panel is the Storage Group Application Selection. Enter the source CDS name, for the Storage Group that will be copied, into the CDS Name field (Figure 6-7).

```
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    DBOA             (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;
```

Figure 6-7 Storage Group selection panel with specific Storage Group selected

In Figure 6-7, the name of the source Storage Group to be copied is entered. This is optional; you could also set the Storage Group Name as * (asterisk) to load all the Storage Groups and choose the one you want to copy. To reduce the range of Storage Groups displayed, you could enter the Storage Group Type as “pool” because no other Storage Group is allowed to copy the volume information (Figure 6-8).

```
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    *                (For Storage Group List, fully or
                                         partially specified or * for all)
  Storage Group Type    pool             (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;
```

Figure 6-8 Storage Group selection panel to display all pool Storage Groups in that CDS

To perform the copy, on the Storage Group List panel, enter COPY next to the Storage Group that will be copied to the new construct, as shown in Figure 6-9.

```
STORAGE GROUP LIST
Command ===>                                                Scroll ===> HALF
                                                        Entries 1-10 of 70
                                                        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)---
    COPY        DB0A     POOL              -------  ----  NO
                DB0ATC   POOL              -------  ----  NO
                DB0BARCH POOL              -------  ----  NO
                DB0BDATA POOL              -------  ----  NO
                DB0BIMAG POOL              -------  ----  NO
                DB0BLOG1 POOL              -------  ----  NO
                DB0BLOG2 POOL              -------  ----  NO
                DB0BMISC POOL              -------  ----  NO
                DB0BTARG POOL              -------  ----  NO
                DB8ADATA POOL              -------  ----  NO
```

Figure 6-9 Storage Group list panel

The Copy Entry panel is displayed as shown in Figure 6-10. In this panel choose to copy the Storage Group into the same CDS or to a different CDS. After entering the required information press Enter to continue.

It is possible to provide a new name for the new Storage Group that will be created in the specified CDS. Figure 6-10 shows a copy is being done to a different CDS using a new name for the Storage Group to be created.

```
 COPY ENTRY PANEL
Command ===>

Construct will be Copied from:

  Data Set Name . . : SYS1.SMS.SCDS
  Construct Name  . : DBOA
  Construct Type  . : STORAGE GROUP

Specify "Copy To" Construct:

  Data Set Name . . . 'MHLRES4.SMS.SCDS1'
                                (1 to 46 Characters)

  Construct Name  . . DBOAC2
                      (If Copy Pool - 1 to 23 characters, fully specified)
                      (Others   - 1 to 8 characters, fully specified)
Enter "/" to select option      Replace like-named Construct
                                Perform Alter
                              /  Copy Storage Group Volumes (Pool SG only)
Use ENTER to Perform COPY;
```

Figure 6-10 Copy Storage Group final panel

To copy the volumes inside the Storage Group you must specify a slash in front of the Copy Storage Group Volumes field. The default for this copy is NO.

Note: The “Copy Storage Group Volumes” default value is NO. It must be selected to have the volumes processed during the copy.

If duplicate volume names are found, SMS returns reason code CSRDUPV (6707) and lists the volume named in the ISMF log data, without adding it to the target Storage Group. The copy process continues for each volume in the list until the end of the list is reached. For more information about this new copy option for pool type Storage Groups, see *z/OS DFSMSdfp Storage Administration,* SG26-7402.

6.3.2 Copy Storage Group into the same CDS

If the Storage Group copy is into the same CDS, there will be the occurrence of duplicate volumes because they belong to the same CDS. For this situation, if the Copy Storage Group Volumes option was selected, an error will be issued to let you know that volumes were not copied.

Figure 6-11 shows that the Storage Group copy is into the same CDS.

.

```
 COPY ENTRY PANEL         SMS RETCODE:         4
Command ===>

Construct will be Copied from:

  Data Set Name . . : SYS1.SMS.SCDS
  Construct Name  . : DBOA
  Construct Type  . : STORAGE GROUP

Specify "Copy To" Construct:

  Data Set Name . . . 'SYS1.SMS.SCDS'
                                (1 to 46 Characters)

  Construct Name  . . DBOATC
                      (If Copy Pool - 1 to 23 characters, fully specified)
                      (Others   - 1 to 8 characters, fully specified)
Enter "/" to select option      Replace like-named Construct
                                Perform Alter
                            /   Copy Storage Group Volumes (Pool SG only)
Use ENTER to Perform COPY;
```

Figure 6-11 Selection CDS and Storage Group name target panel with duplicate volumes

Figure 6-12 shows the message generated when volumes are not copied.

```
 COPY ENTRY PANEL         SMS RETCODE:         4
 Command ===>

 Construct will be Copied from:

   Data Set Name . . : SYS1.SMS.SCDS
   Construct Name  . : DBOA
   Construct Type  . : STORAGE GROUP

 Specify "Copy To" Construct:

   Data Set Name . . . 'SYS1.SMS.SCDS'
                                (1 to 46 Characters)

   Construct Name  . . DBOATC
                       (If Copy Pool - 1 to 23 characters, fully specified)
EsssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssN
e RSNCODE:      6707 CSECT:IGDCSC50 SUBRETCODE:         0 SUBRSNCODE: e
e 0                                                                   e
DssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssssM y)
 Use ENTER to Perform COPY;
```

Figure 6-12 Message indicating an error 6707 in Copy Storage Group

When duplicate volume names are found, SMS returns reason code CSRDUPV(6707) and lists the volume names in the ISMF log data, without adding them to the target Storage Group, as shown in Figure 6-12.

The reason code 6707 appears under Configuration Services reason code in the *DFSMSdfp Diagnosis*, GY27-7618 manual. The message provides information about the Duplicate volume found in the target CDS. When this message is received, the Storage Group structure is copied, but the volume is not.

6.3.3 Copy of non-POOL Storage Groups

For Storage Group types other than POOL, the copy will work, but the volume entries are not copied. You will get an error message because of the different type of Storage Group, as shown on Figure 6-13; however, the new Storage Group structure will be copied with no volume specification.

```
-------------------------------------------------------------------------------
                              COPY ENTRY PANEL                INVALID SG TYPE
Command ===>

Construct will be Copied from:

  Data Set Name . . : SYS1.SMS.SCDS
  Construct Name  . : TOGRP1
  Construct Type  . : STORAGE GROUP

Specify "Copy To" Construct:

  Data Set Name . . . 'MHLRES4.SMS.SCDS1'
                               (1 to 46 Characters)

  Construct Name  . . TCOPT
                      (If Copy Pool - 1 to 23 characters, fully specified)
                      (Others   - 1 to 8 characters, fully specified)
Enter "/" to select option      Replace like-named Construct
                                Perform Alter
                             /  Copy Storage Group Volumes (Pool SG only)
Use ENTER to Perform COPY;
```

Figure 6-13 Copy Storage Group with different types of POOL type error

Pressing PF1 shows the related error messages, indicating that the Storage Group must be POOL as shown in Figure 6-14 on page 83.

```
 COPY ENTRY PANEL                INVALID SG TYPE
Command ===>

Construct will be Copied from:

  Data Set Name . . : SYS1.SMS.SCDS
  Construct Name  . : TOGRP1
  Construct Type  . : STORAGE GROUP

Specify "Copy To" Construct:

  Data Set Name . . . 'MHLRES4.SMS.SCDS1'
                               (1 to 46 Characters)

  Construct Name  . . TCOPT
                      (If Copy Pool - 1 to 23 characters, fully specified)
                      (Others   - 1 to 8 characters, fully specified)
Enter "/" to select  EssssssssssssssssssssssssssssssssssssN uct
                     e Storage Group Type must be POOL e
                     DssssssssssssssssssssssssssssssssssssM s (Pool SG only)
Use ENTER to Perform COPY;
```

Figure 6-14 Copy Storage Group error

Figure 6-15 shows the copy of Storage Group with a Storage Group type of POOL.

```
 STORAGE GROUP LIST
 Command ===>                                                Scroll ===> HALF
                                                        Entries 1-10 of 93
                                                        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)---
                 CPBSG0   COPY POOL BACKUP  -------  ----  --------
                 CPBSG1   COPY POOL BACKUP  -------  ----  --------
                 CPBSG2   COPY POOL BACKUP  -------  ----  --------
     COPY        DB0A     POOL              -------  ----  NO
                 DB0ACPB  COPY POOL BACKUP  -------  ----  --------
                 DB0BARCH POOL              -------  ----  NO
                 DB0BCPB  COPY POOL BACKUP  -------  ----  --------
                 DB0BDATA POOL              -------  ----  NO
                 DB0BIMAG POOL              -------  ----  NO
                 DB0BLOG1 POOL              -------  ----  NO
```

Figure 6-15 Storage Group List

The COPY Entry Panel is shown in Figure 6-16 on page 84. Enter a / (forward slash) to do the copy and specify a new Construct Name, DB0ATC, as shown in the figure.

```
COPY ENTRY PANEL
Command ===>

Construct will be Copied from:

  Data Set Name . . : SYS1.SMS.SCDS
  Construct Name  . : DBOA
  Construct Type  . : STORAGE GROUP

Specify "Copy To" Construct:

  Data Set Name . . . 'SYS1.SMS.SCDS'
                                (1 to 46 Characters)

  Construct Name  . . dbOatc
                      (If Copy Pool - 1 to 23 characters, fully specified)
                      (Others   - 1 to 8 characters, fully specified)
Enter "/" to select option      Replace like-named Construct
                                Perform Alter
                              /  Copy Storage Group Volumes (Pool SG only)
Use ENTER to Perform COPY;
```

Figure 6-16 Copy storage group panel with new option

6.3.4 Storage Group Copy Quick list

Follow these steps to access this new option:

1. From the ISMF Primary panel, select option 6.
2. Enter your CDS Name and select option 1. Generate a list of Storage Groups. (Names shown are in our system).
3. Select a STORGRP NAME and type COPY**.**
4. Provide the target CDS Name and the Storage Group new name.
5. Select the field “Copy Storage Group Volumes.”

After completing these steps a new Storage Group will be created in the CDS, with the exact same configuration and volume information as the source Storage Group.

6.3.5 Migration and coexistence considerations

There is no coexistence requirement or migration action to use this new enhancement.

6.4 VOLSELMSG and TRACE parameters

VOLSELMSG and TRACE are diagnostic facilities in SMS used to get statistics and details about SMS allocation errors and related events that need to be analyzed. The analysis messages are written to the hardcopy log and the job log.

The VOLSELMSG and TRACE parameters were used to share the settings of TYPE, ASID and JOBNAME, so in case you needed a VOLSELMSG setting you would end up having exactly the same setting for TRACE, and vice-versa.

The new support in z/OS V1.12 allows different values to be specified on the shared parameters for TRACE and VOLSELMSG facilities. This allows the operator to have better control over SMS TRACE and VOLSELMSG diagnostic tools. The changes in V1.12 are for the shared parameters TYPE, ASID, and JOBNAME.

TRACE parameter

The TRACE parameter should be enabled when needed to gather performance or troubleshooting information from SMS, to be used when working with the IBM Support Center on a problem. The IBM Support Center might request a copy of the SMS trace table. You can obtain this by dumping the contents of the SMS address space to a SYS1.DUMPnn data set, using the DUMP operator command at the time of the failure. There are several TRACE sub-parameters that can be specified related the level of information needed. Entries in the trace table can be of two types: ERROR or NONERROR.

VOLSELMSG parameter

The VOLSELMSG parameter produces information about volume selection with analysis messages issued when you create or extend an SMS-managed data set to a new volume. Those messages are useful to identify the point of the allocation problem or if any different selection criteria is in place.

When all volumes are to be included, volumes are listed by Storage Group. If only a subset of volumes is to be included, volumes are listed in volume selection preference order with no association to Storage Group.

To more detailed information about TRACE and VOLSELMSG, see *DFSMSdfp Diagnosis*, GY27-7618.

6.4.1 New configuration for TRACE and VOLSELMSG

If you want to apply any specific criteria to only one of those facilities, you must specify the facility name TRACE or VOLSELMSG, or the short form, T or V, as a sub-parameter in parentheses after the parameter value. For example, if you want to set a TYPE value of ERROR for the SMS TRACE facility, and a TYPE value of ALL for the VOLSELMSG facility, you could code the following in the IGDSMSxx PARMLIB member:
TYPE(ERROR(TRACE),ALL(VOLSELMSG))

The new options for the parameters TRACE and VOLSELMSG can be implemented or not, depending on the objective of the diagnosis. If you want the values of the TYPE, ASID, or JOBNAME parameters to apply to both VOLSELMSG and SMS TRACE, simply code the parameter value as in previous releases and that will have no impact.

6.4.2 Display TRACE and VOLSELMSG

To list the current system specifications related to VOLSERLMSG and TRACE, issue the console command D SMS,TRACE. That command will display the TRACE and VOLSELMSG settings as shown in Figure 6-17 on page 86.

```
IGD002I 20:48:52 DISPLAY SMS 767
TRACE    = ON     SIZE = 128K     TYPE = ERROR
  JOBNAME = *    ASID = *
  TRACING EVENTS:
    MODULE = ON   SMSSJF = ON   SMSSSI = ON   ACSINT = ON
    OPCMD  = ON   CONFC  = ON   CDSC   = ON   CONFS  = ON
    MSG    = ON   ERR    = ON   CONFR  = ON   CONFA  = ON
    ACSPRO = ON   IDAX   = ON   DISP   = ON   CATG   = ON
    VOLREF = ON   SCHEDP = ON   SCHEDS = ON   VTOCL  = ON
    VTOCD  = ON   VTOCR  = ON   VTOCC  = ON   VTOCA  = ON
    RCD    = ON   DCF    = ON   DPN    = ON   TVR    = ON
    DSTACK = ON   UAFF   = ON   DEBUG  = ON
VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = *
    ASID = *  STEPNAME = *
    DSNAME = *
```

Figure 6-17 D SMS,TRACE command display

Note that the command D SMS,TRACE brings you both specifications, for TRACE and for VOLSELMSG.

Another command that can be used, but with less detailed information for the TRACE parameter, is D SMS, VOLSELMSG. That command will display the VOLSELMSG configuration and also information about the TRACE settings as shown in Example 6-3.

Example 6-3 SMS,VOLSELMSG command display

```
IGD002I 13:30:46 DISPLAY SMS 879
 VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = *
     ASID = *  STEPNAME = *
     DSNAME = *
 TRACE    = ON     SIZE = 128K     TYPE = ERROR
   JOBNAME = MHLRES4S    ASID = *
```

The D SMS,VOLSELMSG command displays the settings of the parameters that are related to SMS volume selection analysis messages.

The configuration shown in Figure 6-17 is on a z/OS V1.12 system. There is no change to the SMS display related to this enhancement. The same information is displayed in earlier z/OS versions.

To find a detailed explanation of each sub-parameter of TRACE and VOLSELMSG, see *z/OS MVS Initialization and Tuning Reference,* SA22-7592 and *z/OS MVS SystemCommands,* SA22-7627.

6.4.3 Syntax

The syntax for these three shared parameters was enhanced, allowing them to be set with different values for TRACE and VOLSELMSG facilities. Each parameter TYPE, ASID, and JOBNAME can have a maximum of two sub-parameters, which identify whether it is related to the TRACE or VOLSELMSG facility.

The value for TYPE, ASID, or JOBNAME parameters can be set in PARMLIB member IGDSMSxx to be referred to at IPL time, or can be updated using the SETSMS command.

IGDSMSxx parmlib member

With the IGDSMSxx PARMLIB member, using the VOLSELMSG keyword allows you to control volume selection analysis messages issued when you create or extend an SMS-managed data set to a new volume. Example 6-4 shows an example of how this is coded in PARMLIB.

Example 6-4 IGDSMSxx PARMLIB information for TRACE and VOLSELMSG

```
SMS ACDS(SYS1.SMS.ACDS)
    COMMDS(SYS1.SMS.COMMDS)
    INTERVAL(15)
    DINTERVAL(150)
.
.
.
    TRACE(ON)
    SIZE(128K)
    TYPE(ERROR)
    JOBNAME(JOB111)
    ASID(20)
    SELECT(ALL)
```

Example 6-5 shows the specification of these shared parameters applies to both facilities simultaneously.

Example 6-5 Values being applied to TRACE and VOLSELMSG

```
TYPE(ERROR)
- Both TRACE and VOLSELMSG are limited to the ERROR event ASID(20)
- Both TRACE and VOLSELMSG are limited to ASID 20 JOBNAME(JOB111)
- Both TRACE and VOLSELMSG are limited to job name JOB111.
```

When none of the optional sub-parameters is specified, the value specified in the first sub-parameter applies to both TRACE and VOLSELMSG as in previous releases.

To have different values set for each parameter at IPL time, you must code them in PARMLIB using the syntax shown in Figure 6-18.

```
TYPE(ALL|ERROR[(T|V),[ALL|ERROR(T|V)]]
ASID(asid|*[(T|V),[asid|*(T|V)]]
JOBNAME(jobname|*[(T|V),[jobname|*(T|V)]]
```

Figure 6-18 PARMLIB syntax for different values for TRACE and VOLSELMSG

Each parameter is allowed to have up to two sets of sub-parameters; `T|TRACE` or `V|VOLSELMSG` are valid facility names.

The first optional sub-parameter, T | V, specifies whether the first set of sub-parameters applies to TRACE or VOLSELMSG.

SETSMS command

There are situations where the operator might need to use TRACE and VOLSELMSG on various events, ASID, and job names without affecting each other.

For those specific situations, the dynamic setting of the parameters would be a better solution than a static parameter in PARMLIB.

Use the SETSMS command when SMS is active to change a subset of SMS parameters from the console without changing the active IGDSMSxx member of SYS1.PARMLIB.

It is a good practice to display the current parameters before making any changes, so it will be possible to restore them after the end of analysis. To display them you just issue a D SMS,TRACE command as shown in Figure 6-17 on page 86.

You can use the SETSMS command to control how to issue the volume selection analysis messages (see Example 6-6). To issue detailed analysis messages for all volumes on failure allocations, enter the command:

```
SETSMS VOLSELMSG(ON,ALL) TYPE(ERROR)
```

Example 6-6 Using the SETSMS command with VOLSELMSG

```
SETSMS VOLSELMSG(ON,ALL) TYPE(ERROR)
IEE712I SETSMS   PROCESSING COMPLETE

DISPLAY SMS,VOLSELMSG
IGD002I 13:50:25 DISPLAY SMS 841
VOLSELMSG = (ON,ALL)    TYPE = ERROR    JOBNAME = *
    ASID = *  STEPNAME = *
    DSNAME = *
TRACE    = ON     SIZE = 128K     TYPE = ERROR
  JOBNAME = *    ASID = *
```

It is possible to code different criteria for both parameters in the same command. The second set of sub-parameters is optional and can be used to specify another value and facility after the first set of sub-parameters is specified. Following there are some options that can be coded. There is no positional demand between TRACE and VOLSELMSG, which means that you can code them in any order.

Consider this command:

```
JOBNAME(MHLRES4C(TRACE),MHLRES4D(V))
```

In this command the TRACE facility is limited to job name MHLRES4C and the VOLSELMSG facility is limited to job name MHLRES4D. Example 6-7 shows how those commands are accepted by SMS.

Example 6-7 Simultaneous different settings for TRACE and VOLSELMSG

```
SETSMS JOBNAME(MHLRES4C(T),MHLRES4D(V))
IEE712I SETSMS   PROCESSING COMPLETE

DISPLAY SMS,VOLSELMSG
 IGD002I 19:12:43 DISPLAY SMS 834
 VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = MHLRES4D
     ASID = 20 STEPNAME = *
     DSNAME = *
 TRACE    = ON     SIZE = 128K     TYPE = ERROR
   JOBNAME = MHLRES4C    ASID = *
```

6.4.4 Parameter options

The options for the TRACE and VOLSELMSG are the same as in previous versions. The difference is the possibility of coding different values for ASID, JOBNAME, and TYPE. Here we show those parameter options.

- `ASID(asid|*[(TRACE|T|VOLSELMSG|V)[,asid|*(TRACE|T|VOLSELMSG|V)]])`
  Specifies whether SMS is to limit tracing (TRACE(ON)) or issue volume selection messages (VOLSELMSG(ON)) to a specific address space (asid) or all address spaces *.

  asidl*- Limit tracing and volume selection analysis messages for a certain address space, or all address spaces (*). You can enter up to four digits for the ASID keyword. If you leave off the leading zeros, they are inserted. Default: *
- `JOBNAME(jobname|*[(TRACE|T|VOLSELMSG|V)[,jobname|*(TRACE|T|VOLSELMSG|V)]])`
  Specifies whether SMS is to limit tracing (TRACE) and volume selection analysis messages (VOLSELMSG(ON)) for a certain job (jobname), or permit tracing and volume selection analysis messages on all jobs.This keyword supports objects or tape libraries.

  jobnamel* - Limit tracing and volume selection analysis messages for a certain job, or all jobs (*). Specify "*" to issue for all jobs. Default: *
- `TYPE(ERROR|ALL[(TRACE|T|VOLSELMSG|V)[,ALL|ERROR(TRACE|T|VOLSELMSG|V)]])`
  Specifies how you want to trace events (TRACE) and issue volume selection analysis messages (VOLSELMSG).

  Specify ERROR to trace error events and issue volume selection analysis messages (VOLSELMSG(ON)) for allocations that have failed.

  Specify ALL to trace all events and issue volume selection analysis messages (VOLSELMSG(ON)) for all allocations.

6.4.5 TRACE and VOLSELMSG Examples

To illustrate the utilization of VOLSELMSG we provide examples of the commands and the expected SYSOUT presented on SYSLOG and JOBLOG.

Figure 6-19 shows the D SMS, TRACE command.

```
D SMS, TRACE
```

Figure 6-19 D SMS,TRACE command

Example 6-8 shows the result of the D SMS,TRACE command.

Example 6-8 D SMS,TRACE command output

```
RESPONSE=SC64
 IGD002I 19:40:54 DISPLAY SMS 915
 TRACE    = OFF    SIZE = 128K     TYPE = ALL
   JOBNAME = *    ASID = *
   TRACING EVENTS:
     MODULE = ON   SMSSJF = ON   SMSSSI = ON   ACSINT = ON
     OPCMD  = ON   CONFC  = ON   CDSC   = ON   CONFS  = ON
     MSG    = ON   ERR    = ON   CONFR  = ON   CONFA  = ON
     ACSPRO = ON   IDAX   = ON   DISP   = ON   CATG   = ON
     VOLREF = ON   SCHEDP = ON   SCHEDS = ON   VTOCL  = ON
     VTOCD  = ON   VTOCR  = ON   VTOCC  = ON   VTOCA  = ON
```

```
     RCD    = ON   DCF    = ON   DPN    = ON   TVR    = ON
     DSTACK = ON   UAFF   = ON   DEBUG  = ON
 VOLSELMSG = (OFF,0)    TYPE = ALL    JOBNAME = *
     ASID = *  STEPNAME = *
     DSNAME = *
```

As shown in Example 6-8, it is possible to verify that TRACE and VOLSELMSG are both disabled. This means that no information is being recorded for log purposes.

The following command shows the change to TRACE and VOLSELMSG to get additional information about the new SMS occurrences; it was issued to set different jobnames to be monitored per VOLSELMSG and TRACE:

```
SETSMS JOBNAME(MHLRES4C(T),MHLRES4D(V))
```

The SETSMS VOLSELMSG command is used to enable information collection. The following command turns on the VOLSELMSG parameter with a maximum of ten volumes being logged:

```
SETSMS VOLSELMSG(ON,10)
```

Figure 6-20 on page 90 show the information presented in SYSLOG for a job that met the criteria of jobname coded in SETSMS JOBNAME(MHLRES4C(T),MHLRES4D(V)).

```
IGD17385I =====SUMMARIZED ANALYSIS MESSAGES ON DEFINING DATA SET
MHLRES4.EXTDC.TEST15 =====
IGD17386I VOLSELMSG(ON,A) TYPE(ALL) JOBNAME( ) ASID(*) 625
STEPNAME(* ) DSNAME(* )
IGD17387I DS_TYPE(NONVSAM) SC(STANDARD) DC() GS(N) SPACE(498023KB)
BESTFIT(N) STRIPING(N)
IGD17290I THERE WERE 1 CANDIDATE STORAGE GROUPS OF WHICH THE FIRST 1
627
WERE ELIGIBLE FOR VOLUME SELECTION.
THE CANDIDATE STORAGE GROUPS WERE:SGMHL01
IGD17269I 2 VOLUMES WERE REJECTED BECAUSE THE SMS VOLUME STATUS WAS
DISABLED
IGD17269I 5 VOLUMES WERE REJECTED BECAUSE THEY WERE NOT ONLINE
IGD17269I 5 VOLUMES WERE REJECTED BECAUSE THE UCB WAS NOT AVAILABLE
IGD17385I =====DETAILED ANALYSIS MESSAGES ON DEFINING DATA SET
MHLRES4.EXTDC.TEST15 =====
.
.
.
```

Figure 6-20 VOLSELMSG displayed information about syslog

6.4.6 Migration and coexistence considerations

There is no coexistence requirement or migration action to use this new enhancement.

7

PDSE enhancements in DFSMS V1.12

This chapter describes the changes in PDSE introduced in DFSMS V1.12. Specifically, there are two new features: the ability to delete all members with the IDCAMS DELETE command and the ability to detect corrupted PDSEs in the LNKLST.

7.1 IDCAMS DELETE PDS and PDSE

Before z/OS V1.12, with the IDCAMS DELETE command, only one data set member could be deleted from a partitioned data set (PDS or PDSE) at a time. Wildcards were not allowed for the member names. With z/OS V1R12, DFSMSdfp access method services (IDCAMS) has added a new wildcard to the DELETE command, which lets you delete all members of a PDS or PDSE. The DELETE command is enhanced to allow users to specify a single asterisk (*) as the member name of a PDS or PDSE, as shown in Figure 7-1. This will delete all the members in that PDS or PDSE.

This new feature allows users to empty a partitioned data set with a single IDCAMS DELETE command, removing the restriction of deleting only one member at a time. This should save administrative work in managing data sets.

```
//STEP1 EXEC PGM=IDCAMS,REGION=4096K
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
  DELETE MHLRES3.PDS.DEL.TEST(*)
  DELETE MHLRES3.PDSE.DEL.TEST(*)
```

Figure 7-1 JCL showing DELETE commands for a PDS and PDSE

You will receive the new IDC0553I message as shown in Figure 7-2. This message has been added with this support.

```
 DELETE MHLRES3.PDS.DEL.TEST(*)
IDC0553I ALL MEMBERS IN DATA SET MHLRES3.PDS.DEL.TEST  DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

  DELETE MHLRES3.PDSE.DEL.TEST(*)
IDC0553I ALL MEMBERS IN DATA SET MHLRES3.PDSE.DEL.TEST  DELETED
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

Figure 7-2 IDC0553I message

7.2 Identify corrupt PDSE in LNKLST

In prior releases, when a corrupted PDSE was detected in the link list during IPL, the system often entered a wait state. In z/OS V1.12, the system is designed to issue a message identifying the corrupt PDSE and continue the IPL without placing that PDSE into the link list. This allows the user to attempt to restore the corrupt PDSE, and re-IPL the system if necessary, without taking a stand-alone dump to debug the problem.

When a PDSE is opened during the NIP, its directory structures will be validated. A new message IGW037I with the corrupt data set name will appear during IPL when a corrupted PDSE is found in the linklist (see Example 7-1). The IPL might continue without the data set, but the system could be compromised and may enter a disable wait state of 064 if the corrupted PDSE generates a program check. To avoid that possibility, you can attempt to restore a valid copy of the corrupted data set and re-IPL the system.

Example 7-1 IGW037I message

```
IGW037I PDSE ERROR DURING NIP PROCESSING, DSNAME: pdsedsname MODULE NAME:
                modulename RETURN CODE: retcd  REASON CODE: rsncd
```

The return codes and reason codes for the IGW037I can be found in the *DFSMSdfp Diagnosis*, GY27-7618 Section 15.0 PDSE Diagnostic Aids.

8

DFSMShsm enhancements in z/OS V1.12

In this chapter we provide details of the enhancements to DFSMShsm provided in z/OS V1.12. These include:

- Fast reverse restore
- Increased stack value for dump stacking
- Multitask recovery from dump
- Define cross memory invocation function
- DFSMShsm Fast Replication

8.1 Fast reverse restore

Fast reverse restore provides the capability to reverse the direction of an existing FlashCopy relationship and restore the source volume to the point-in-time state when it was last flashed to the target volume without waiting for the background copy to complete. Once a fast reverse restore has completed, the contents of the backup volume, which is the original FlashCopy target volume, become invalid. This allows users to recover much more quickly than they did in the past.

DFSMShsm supports fast reverse restore of a copy pool in both COPY and NOCOPY modes. However, fast reverse restore cannot be used in combination with the preserve mirror function.

For more in depth information, refer to *DFSMShsm Fast Replication Technical Guide*, SG24-7069.

8.2 Dump stacking

Dump stacking has always been limited to 99 volumes. This limit is now increased to 255 volumes. Tapes can be used more efficiently by stacking up to 255 volumes onto one single dump tape.

AUTODUMP processing will now allow up to 255 volumes to be stacked together. Any parameter that referenced a stack value can now specify up to 255 volumes for the stack value.

8.2.1 DEFINE DUMPCLASS command with new stack value

The DEFINE DUMPCLASS command can be used to define or update a dump class stack value. You can execute the DEFINE DUMPCLASS as shown in Example 8-1.

Example 8-1 DEFINE DUMPCLASS STACK(255)

```
DEFINE DUMPCLASS(DCNEW AUTOREUSE FREQUENCY(1) -
   RETENTIONPERIOD(3) -
   FRRECOV(AVAILABLEFORMOUNT(YES)) -
   FRDUMP(OPTIONAL)   -
   STACK(255) UNIT(VT3590G2) VTOCCOPIES(1))
```

The command in Example 8-1 also can be used in an HSEND TSO batch job or it can be added to an ARCCMDxx parmlib member. Once the dump class has been defined, you will receive the ARC0216I message as shown in Figure 8-1.

```
ARC0216I DUMPCLASS DEFINITION SUCCESSFUL, CLASS=DCNEW RC=0
```

Figure 8-1 ARC0216I message

DFSMShsm messages

All messages for dump functions will be written to the DFSMShsm dump activity log. The activity log is defined in the DFSMShsm STARTUP procedure. The amount of data sent to the log is control by the SETSYS ACTLOGMSGLVL command. It is recommended that the FULL setting be used. An example of this command is shown in Example 8-2.

Example 8-2 SETSYS ACTLOGMSGLVL command

```
SSETSYS ACTLOGMSGLVL (FULL)
```

Activity log naming convention

The data set naming convention for the activity logs differs by customer environments. However, it usually follows the standard:

```
mcvtactn.Hmcvthost.function.Dyyddd.Thhmmss
```

Where:

mcvtactn	The Activity log high-level qualifier
Hmcvthost	The Host Identifier for the DFSMShsm host that creates these activity logs
function	BAKLOG or CMDLOG
Dyyddd	Year and day of allocation
Thhmmss	Hour, minute, and second of allocation

For more information on the BACKUP and COMMAND ACTIVITY log, refer to the *DFSMShsm Implementation and Customization Guide,* SC35-0418.

8.2.2 BACKVOL DUMP with new stack value

If you want to override an existing stack value and not make the change a permanent change to the dumpclass, you can use the BACKVOL DUMP command as shown in Example 8-3.

Example 8-3 BACKVOL command

```
BACKVOL VOLUMES(PRIM01,PRIM02,NSMS01) -
    DUMP(DUMPCLASS(DCNEWFR) STACK(255))
```

The messages are written to the dump activity log as previously described in “DFSMShsm messages” on page 96. Figure 8-2 on page 98 shows the messages for the first volume being processed in the BACKVOL command as well as the ending messages of all of the volumes. Note that not all the volume messages are listed.

```
ARC0622I FULL VOLUME DUMP STARTING ON VOLUME PRIM01(NONSMS) AT 12:08:29 ON 2010/10/21,
SYSTEM 3090, TASK ID=ARCDVOL1
ARC0728I VTOC FOR VOLUME PRIM01 COPIED TO DATA SET DFHSM.DUMPVTOC.T290812.VPRIM01.D10294
ON VOLUME MIG103
ARC0640I ARCDVOL1 PAGE 0001 5695-DF175  DFSMSDSS V1R12.0 DATA SET SERVICES 2010.294
12:08
ARC0640I ARCDVOL1 ADR035I (SCH)-PRIME(06), INSTALLATION EXIT ALTERED BYPASS FAC CLASS
CHK DEFAULT TO YES
ARC0640I ARCDVOL1 ADR035I (SCH)-PRIME(03), INSTALLATION EXIT ALTERED WORKUNIT DEFAULT TO

ARC0640I ARCDVOL1 DUMP FULL INDDNAME(SYS00037)-
ARC0640I ARCDVOL1 OUTDDNAME(SYS00038) -
ARC0640I ARCDVOL1 ALLEXCP ALLDATA(*) OPTIMIZE(1) TOLERATE(IOERROR)
ARC0640I ARCDVOL1 ADR101I (R/I)-RI01 (01), TASKID 001 HAS BEEN ASSIGNED TO COMMAND 'DUMP
'
ARC0640I ARCDVOL1 ADR109I (R/I)-RI01 (01), 2010.294 12:08:30 INITIAL SCAN OF USER
CONTROL STATEMENTS COMPLETED
ARC0640I ARCDVOL1 ADR050I (001)-PRIME(02), DFSMSDSS INVOKED VIA CROSS MEMORY APPLICATION
INTERFACE
ARC0640I ARCDVOL1 ADR016I (001)-PRIME(01), RACF LOGGING OPTION IN EFFECT FOR THIS TASK
IEC501A M 0B97,PRIVAT,SL,COMP,DFHSM,JV,DFHSM.DMP.DCNEWFR.VPRIM01
IEC705I TAPE ON 0B97,A00014,SL,COMP,DFHSM,JV,DFHSM.DMP.DCNEWFR
ARC0640I ARCDVOL1 ADR006I (001)-STEND(01), 2010.294 12:08:43 EXECUTION BEGINS
ARC0120I DUMP VOLUME A00014 ADDED, RC= 0, REAS= 0
ARC0640I ARCDVOL1 ADR006I (001)-STEND(02), 2010.294 12:08:43 EXECUTION ENDS
ARC0640I ARCDVOL1 ADR013I (001)-CLTSK(01), 2010.294 12:08:43 TASK COMPLETED WITH RETURN
CODE 0000 ARC0640I ARCDVOL1 ADR012I (SCH)-DSSU (01), 2010.294 12:08:43 DFSMSDSS
PROCESSING COMPLETE. HIGHEST RETURN CODE IS 0000
ARC0637I DUMP COPY OF VOLUME PRIM01 COMPLETE, DCLASS=DCNEWFR, EXPDT=2010/10/24
ARC0623I FULL VOLUME DUMP OF VOLUME PRIM01 ENDING AT 12:08:43, PROCESSING SUCCESSFUL
.
.
.
IEC205I SYS00038,DFHSM,JV,FILESEQ=2, COMPLETE VOLUME LIST,  8
DSN=DFHSM.DMP.DCNEWFR.VPRIM02.D10294.T430812,VOLS=A00014,TOTALBLOCKS=123999
IEC205I SYS00038,DFHSM,JV,FILESEQ=3, COMPLETE VOLUME LIST,  9
DSN=DFHSM.DMP.DCNEWFR.VNSMS01.D10294.T430812,VOLS=A00014,TOTALBLOCKS=240345
IEF234E K 0B97,A00014,PVT,DFHSM,JV
ARC0637I DUMP COPY OF VOLUME NSMS01 COMPLETE, DCLASS=DCNEWFR, EXPDT=2010/10/24
ARC1000I VOL=PRIM01 BACKVOL PROCESSING ENDED
ARC1000I VOL=PRIM02 BACKVOL PROCESSING ENDED
ARC1000I VOL=NSMS01 BACKVOL PROCESSING ENDED
```

Figure 8-2 Sample BACKVOL Command output

8.3 Multitask recovery from dump

Prior to DFSMS V1.12, dump recovery processing would take longer because dump processed only one dump command at a time. Dump volume recovery is now multi-tasked. A user can now instruct dump recovery to run 64 concurrent dump recovery commands at one time. This feature substantially reduces the recovery window for both Fast Replication copy pools and other dump volume recoveries.

8.4 Cross memory invocation definitions by function

In an effort to reduce below the line storage constraints, in z/OS V1.7 DFSMShsm made use of the DFDSS cross memory interface. The original settings were designed so that all DFSMShsm functions invoked DFDSS via the cross memory interface or none of the functions did.

Note: DFSMShsm FRBACKUP and FRRECOV functions are the only functions that have always invoked DFDSS via the cross memory interface by default.

In z/OS DFSMShsm V1.9 through V1.11 users could specify whether or not DFSMShsm would invoke cross memory mode using the DSSXMMODE parameter. If none was specified, NO was the default and DFDSS was invoked without using cross memory.

In z/OS DFSMS V1.12, additional parameters were added to the DSSXMMODE command so that users could specify which functions will invoke DFDSS via cross memory mode and which will not. Previously, cross memory was invoked for all functions or none (with the exception of FRBACKUP). Each setting is either Y or N: Yes, invoke DSS in cross memory mode; No, do not invoke DSS in cross memory mode. If nothing is specified, the default is No as in previous releases.

Figure 8-3 shows an example of how to specify the new subparameters on the command in the ARCCMDxx SYS1.PARMLIB member.

```
SETSYS DSSXMMODE(BACKUP(Y)  /* DON'T XMEM BACKUP   */   -
       CDSBACKUP(Y)         /* RUN CDSBKUP IN XMEM */   -
       DUMP(Y)              /* RUN XMEM DUMP       */   -
       MIGRATION(Y)         /* RUN MIG IN XMEM     */   -
       RECOVERY(Y)          /* DON'T XMEM RECOV    */
```

Figure 8-3 DSSXMMODE parmlib member example

After DFSMShsm is started, QUERY SETSYS will display these options as part of the ARC0150I message as shown in Example 8-4.

Example 8-4 Message ARC0150I

```
ARC0150I (CONT.) DSSXMMODE=(BACKUP=YES, CDSBACKUP=YES,
ARC0150I (CONT.)  DUMP=YES, MIGRATION=YES,
ARC0150I (CONT.) RECOVERY=YES)
```

When any of the specified functions start, a new DFDSS address space will be attached. It can be seen under SDSF, DISPLAY ACTIVE. IEESYSAS is the jobname, with the new started address space identifiers as the step name as shown in Example 8-5.

Example 8-5 SDSF DISPLAY ACTIVE of cross memory tasks

```
IEESYSAS ARC3CDSB IEFPROC  STC00047 STC NS  FE  829   0.00   0.00
IEESYSAS ARC3BKUP IEFPROC  STC00048 STC NS  FE  812   0.00   0.00
IEESYSAS DSSFRB01 IEFPROC  STC22401 STC NS  F2  815   0.00   0.00
IEESYSAS DSSFRR01 IEFPROC  STC22398 STC NS  F2  815   0.00   0.00
IEESYSAS DSSFRDSR IEFPROC  STC22398 STC NS  F2  815   0.00   0.00
IEESYSAS ARC3DUMP IEFPROC  STC22466 STC NS  F2  805   0.00   0.00
IEESYSAS ARC3MIGR IEFPROC  STC22466 STC NS  F2  805   0.00   0.00
IEESYSAS ARC3RCVR IEFPROC  STC22466 STC NS  F2  805   0.00   0.00
```

```
IEESYSAS ARC3REST IEFPROC  STC22466 STC NS  F2  805   0.00   0.00
IEESYSAS ARC3RST3 IEFPROC  STC22466 STC NS  F2  805   0.00   0.00
```

These tasks are detached when DFSMShsm terminates. PDA tracing will show DFSMShsm invoking the cross memory task and its communication with the *server* address space. DFSMShsm and DFSMSdss cross memory task communicate through the UIM. This activity can be seen in the PDA trace. Most of the activity occurring in the DFSMSdss cross memory task address space is not traced.

8.5 DFSMShsm Fast Replication

This section provides an overview of the additional features in the DFSMShsm Fast Replication function since Phase 1 in z/OS V1.5 DFSMShsm. These enhancements allow DFSMShsm to fully exploit the latest FlashCopy features of the IBM System Storage DS8000 and complete the Phase 2 implementation. In this section, we review the concept of Fast Replication, FlashCopy, and the enhancements as of z/OS V1.12 DFSMShsm.

We start with a sample Fast Replication configuration to provide a quick overview of DFSMShsm Fast Replication.

8.5.1 DFSMShsm Fast Replication sample configuration

Figure 8-4 shows a sample FRBACKUP based target configuration. The purpose here is to provide a first look at what this section is about.

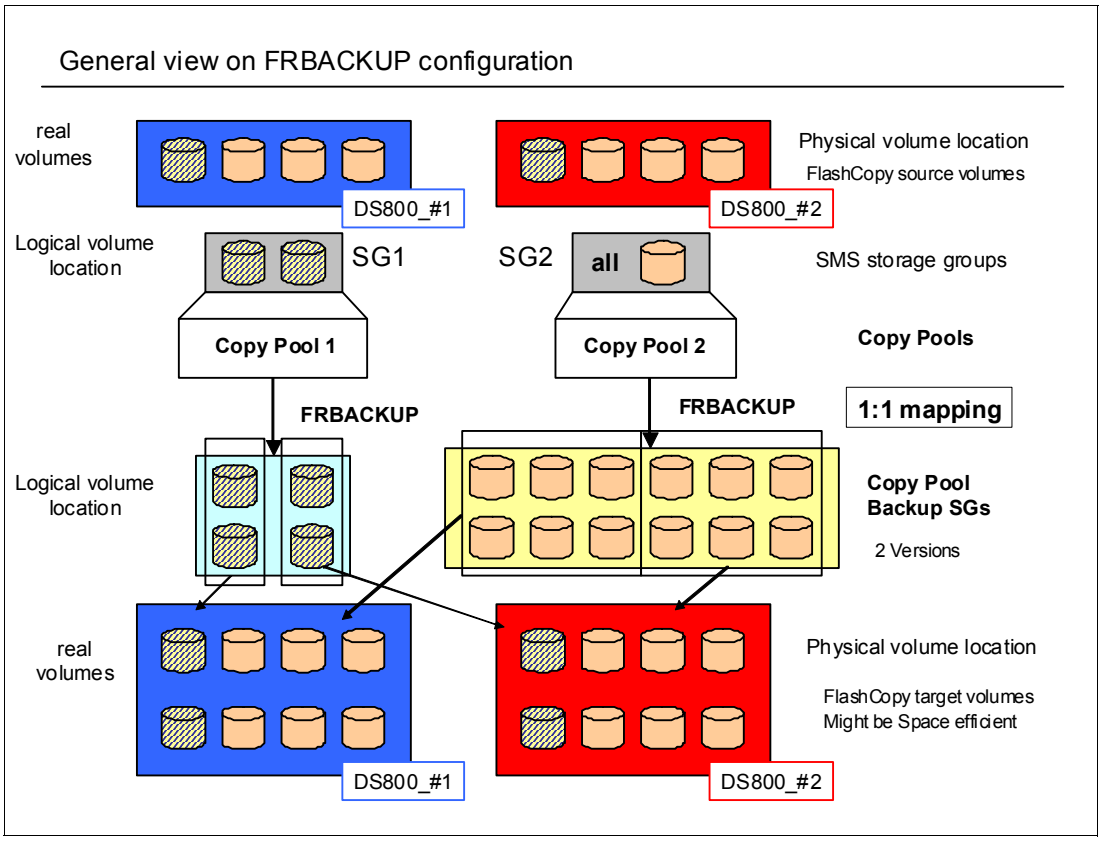

Figure 8-4 Potential Fast Replication target configuration

The remainder of this section provides information on what FRBACKUP is all about and why IBM is providing this solution. DFSMShsm Fast Replication builds exclusively on the FlashCopy functionality in the DS8000.

8.5.2 FlashCopy function overview

In a z/OS and system-managed storage (SMS) environment physical full volume copies are generally not very useful. FlashCopy, as the DS8000 implements the function, only provides physical copies from a source volume to a target volume within the same DS8000. With DFSMShsm Fast Replication management this changed. DFSMShsm manages through enhanced SMS constructs and configuration. Figure 8-4 is a sample Fast Replication configuration. It provides an overview of a complete FRBACKUP configuration and points out what and where the real physical volumes reside. It also shows the logical constructs which are wrapped around the physical volumes and storage subsystems. In this section we focus on the DS8000 FlashCopy function.

SMS already incorporates the logical concept of Storage Groups to logically group volumes. Another tier of logical grouping is now Copy Pools, which map to SMS Storage Groups and can have up to 256 SMS Storage Groups mapped to a pool. This is the source site of a Fast Replication configuration. The target or backup site has another SMS construct, a Copy Pool Backup Storage Group (CPBSG). A CPBSG has all the potential FlashCopy target volumes and contains only FlashCopy target volumes.

Figure 8-4 shows such a Fast Replication configuration as a general purpose fast backup and fast recovery solution. DFSMShsm Fast Replication is mainly geared towards DB2 but can also be utilized for other applications when an instant backup copy of any number of application volumes through FlashCopy is required. What Figure 8-4 does not show are the possible automatic dumps to tape from the CPBSG volumes.

8.6 Fast Replication review

Fast Replication uses FlashCopy, which is a point-in-time copy of a volume. A point-in-time copy gives the appearance of an almost instantaneous volume copy.

Fast data replication occurs so fast because it builds a map, with pointers, to the source volume tracks or extents. There is no longer a need to wait for the physical copy to complete before applications can resume the access to the data. Both the source and target data are available for read/write access almost immediately, while the copy process continues in the background. This process guarantees that the contents of the target volume are an exact duplicate of the source volume at that point in time. You can back up, recover to, and resume processing from that point in time.

Most 24x7 database operations require a mirror solution which has no or little impact on application performance, and provides immediate access to a copy of the mirrored data. This access to a copy of mirrored data, which can be obtained by various techniques with a minimal impact on mirroring, is one of the main returns on investment that many companies utilize.

In the past, volume-level point-in-time copies have been used almost exclusively as disaster recovery backups. With the new software services that are available, these copies, clones, or mirrors can be used for job restart, data mining, and application testing.

8.6.1 Point-in-time hardware support

The DFSMShsm Fast Replication line item supports the volume-level FlashCopy function of the IBM DS8000 disk storage subsystem and any other disk storage subsystem that supports FlashCopy APIs.

> **Note:** Appropriate microcode levels might be required to activate these features on the storage device.

The process of creating a point-in-time copy has two phases. The first phase completes in a few milliseconds by constructing a map to describe the source volume.

At the completion of the first phase, the data has been cloned and both the source and target volumes are available for read and write access. From the user's perspective, the source and target volumes' contents are an exact duplicate at this point in time, even though actual copying of data has not been initiated.

The second phase consists of copying the physical source volume data to the target volume. The point-in-time copy is maintained and preserved by an on-demand copy of the data to the target volume. When an update request has been issued for either a source or target volume data set that has not yet been copied to the target volume, the DS8000 firmware immediately copies the data from the source volume to the target volume before the update is applied. The possible situations are:

- Read request to the source volume

  The data is read from the source volume.

- Read request to the target volume

  If the data has already been copied from the source volume to the target volume, the data is read from the target volume. If the data has not yet been copied to the target volume, the data is read from the source volume.

- Write request to the source volume

  If the data has already been copied from the source volume to the target volume, this results in a normal write to the source volume. However, if the data has not yet been copied from the source volume to the target volume, the data will be copied from the source volume to the target volume before the source volume update occurs.

- Write request to the target volume

  If the data has already been copied from the source volume to the target volume, the write request results in a normal write to the target volume. However, if the data has not yet been copied from the source volume to the target volume, the data will be copied to the target volume before the target volume update occurs.

8.6.2 FlashCopy: Background copy

By default, FlashCopy performs a background copy. The background copy task makes a physical copy of all tracks from the source volume to the target volume. De-staging algorithms are used to efficiently manage the background DS8000 copy process. The background copy task runs at a lower priority than normal I/O on the DS8000, so as not to affect the normal application I/O processing.

The DS8000, using the metadata structure that was created during FlashCopy establish, keeps track of which data has been copied from the source to the target and manages the integrity of both copies. If an application wants to read some data from the target that has not

yet been copied, the data is read from the source; otherwise, the read can be satisfied from the target volume.

Before updating a not-yet-copied track on the source volume, the DS8000 performs an on-demand copy of the track to the target volume by what is called a *copy-on-write* function. Subsequent reads to this track on the target volume will be satisfied from the target volume.

Before updating a not-yet-copied track on the target volume, the DS8000 performs another copy-on-write of this track to the target volume. This on-demand activity is done asynchronously after the host has received I/O completion after the data is written in cache and non-volatile storage (NVS), so host I/O is not delayed.

After some time, when all tracks have been copied to the target volume, the FlashCopy relationship automatically ends unless the persistent FlashCopy option was specified.

As Figure 8-5 illustrates, a FlashCopy relationship goes through three stages:

- Establishing the relationship
- Copying the data
- Terminating the relationship

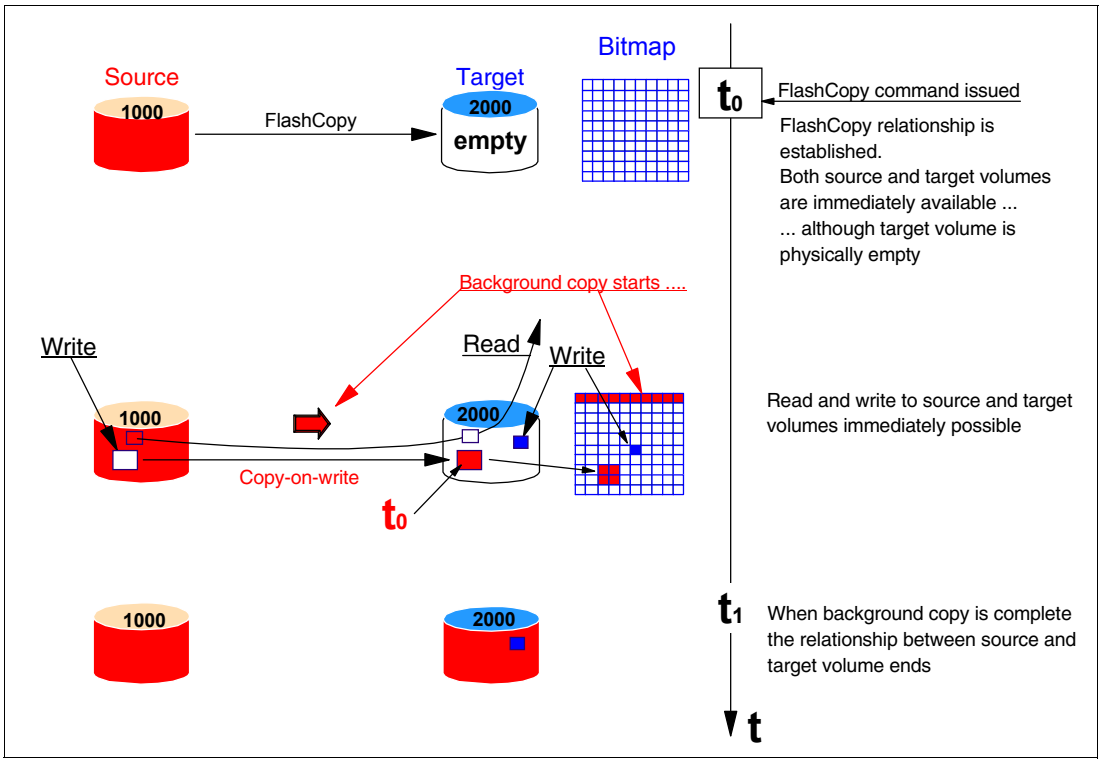

Figure 8-5 FlashCopy with background copy

Note that volume FlashCopy is a physical copy from the very first track to the very last track independent of what is on these tracks.

8.6.3 FlashCopy: No background copy

When selecting *not* to perform the background copy, the relationship is established but the background copy task—of all source volume tracks—is *not* initiated. Only the source tracks that receive application updates are copied to the target. Before updating a track on the source volume, the DS8000 performs an on-demand copy or *copy-on-write* of the track to the

target volume, thus preserving the T_0 copy. Similarly, before updating a track on the target volume, the DS8000 performs an on-demand copy of this track to the target volume.

A FlashCopy relationship that was established selecting no-background copy remains active until one of the following occurs:

- An explicit FlashCopy withdraw is done to terminate the relationship.
- All source device tracks have been copied on the target device because they were all updated.
- All target device tracks have been updated by user applications.

When a no-background copy FlashCopy relationship is terminated, the target volume is left in an indeterminate state. Some of the tracks on the volume might contain data from the source volume, while other tracks might contain residual data that was on the target volume before the copy. The volume should not be used in these conditions unless it is reformatted or used for another copy operation.

8.6.4 FlashCopy Version 2

This is the standard for DS8000.

FlashCopy Version 2 supports all of the FlashCopy V1 functions plus these enhancements:

- FlashCopy V2 can be used for data set copies as well as volume copies.
- The source and target of a FlashCopy can be on different LSSs.
- Up to 12 FlashCopy relationships are allowed.
- Incremental copies are possible.
- Inband FlashCopy commands can be sent over PPRC links to a remote site.
- FlashCopy consistency groups can be created.

In addition, there has been a reduction in the FlashCopy establish times. Figure 8-6 on page 105 compares the features of FlashCopy Version 1 and FlashCopy Version 2.

	FlashCopy V1 Original	FlashCopy V1 Updated	FlashCopy V2 Lic 2.2.0
Background Copy Mode	COPY or NOCOPY	COPY, NOCOPY NOCOPY->COPY	COPY, NOCOPY NOCOPY->COPY
Disposition	Ended	Ended / Persistent	Ended / Persistent
Data Movement	Full	Full	Full or Incremental
Relationship(s)	Single	Single	Multiple
Granularity	Volumes, Tracks	Volumes, Tracks	Volumes, Tracks, Data Set
Data Set	Single	Single	Multiple
Source/Target Boundries	Same LSS	Same LSS	Cross LSS Cross Cluster

Applies to ESS models F10, F20, and 800

Figure 8-6 FlashCopy V1 and FlashCopy V2 comparison for ESS

Figure 8-6 is only relevant for the ESS disk storage subsystem. Everything which is under the FlashCopy V2 column applies to the DS8000 FlashCopy functions, plus the capability to have a Metro Mirror primary volume as a FlashCopy target volume for full copy as well as for data set level FlashCopy. This is what we call *Remote Pair FlashCopy* or *Preserve Mirror* and is only available in DS8000.

8.6.5 DFSMShsm FRBACKUP new features

There are many ways to execute FRBACKUP. Each method is different depending on your installation requirements. The new features added to DFSMShsm increase the variety of ways you can back up your data.

Since z/OS DFSMS 1.5 the following FRBACKUP function has been added:

- Use of INCREMENTAL on copypools provided by z/OS DFSMS 1.8.0 PTF UA28767.
- Allowing use of an Alternate Copypool backup storage group provided by z/OS DFSMS 1.9.0 PTF UA36434.
- Allowing PPRC primary volumes to be selected as target volumes for FRBACKUP and FRRECOV function with new keywords provided in PTFs:
  - z/OS DFSMS 1.8.0 UA42371
  - z/OS DFSMS 1.9.0 UA42372
- Fast Replication Tape Support provided in base z/OS DFSMS V1.8.0.
- In base z/OS DFSMS 1.11:
  - Capture Catalog Information for recovery of deleted data sets provided in base z/OS DFSMS V1.11.

  - If an error occurs during FRBACKUP, ARC1803I lists the DFDSS failing message and volser in the backup activity log to help diagnose errors.
  - In a NOCOPY scenario, if an error occurs, a withdraw is done; now the target volume will be re initialized after withdraw.
- Copy pool recovery from dump tape provided in base z/OS DFSMS 1.12.0.
- Fast Reverse Restore provided in base z/OS DFSMS 1.12.0.
- FlashCopy to Space Efficient Volumes provided in base z/OS DFSMS 1.12.0.
- Additional items in z/OS DFSMS 1.12.0:
  - Up to 64 Volume recovery tasks
  - Up to 64 Data set recovery tasks
  - Up to 254 volume pairs to be backed up at one time

These features are available through new parameters in the FRBACKUP and FRRECOV commands.

9

DFSMSrmm enhancements

The DFSMSrmm enhancements in z/OS V1.12 DFSMS provide improvements in the areas of usability, EDGRMMnn parmlib commands and options, performance, and maintainability.

In this chapter we discuss the following enhancements:

- Ease of use and flexibility enhancements
  - Retention limit reporting
  - Ignore for duplicate volumes
  - Automation for WTORs in production and parallel running
  - Expiration override for volumes
  - ISPF Dialog CLIST option to avoid search results list
- Simplification
  - Display subsystem status information
- Storage reporting
  - Copy Export reporting
- Availability
  - STOP/CANCEL improved
  - PDA trace extended outside RMM address space
- Scalability and performance
  - IPV6 supports networking strategy
  - EAV supported for all data sets
  - EAS exploited for temporary data sets
  - XTIOT, DSAB above, and uncaptured UCB support

9.1 Overview

DFSMSrmm™ provides multiple enhancements in z/OS V1.12. The ACTIVITY file is updated to include the reason why a DFSMSrmm retention limit was reached. This function is also available now for z/OS V1.10 and z/OS V1.11 with the PTF for APAR OA30881. New reports created from the ACTIVITY and extract files are intended to help you see why retention limits were triggered. Also, OPENRULE ignore processing is available for duplicate tape volumes and support is available to allow you to set a volume hold attribute to prevent expiration and to search and report on volumes that have the hold attribute. It is also intended that the DFSMSrmm ISPF dialog search results can be bypassed when using the CLIST option.

- Additional z/OS V1.12 scalability and performance improvements:
  DFSMS now supports additional data set types, including basic and large format sequential data sets, partitioned (PDS/PDSE) data sets, direct (BDAM) data sets and catalogs in the extended addressing space (EAS) on extended address volumes (EAVs). Support is also included for generation data groups (GDGs) and VSAM volume data sets (VVDSs). Overall, EAV helps you relieve storage constraints as well as simplify storage management by providing the ability to manage fewer, large volumes as opposed to many small volumes.
- Another new EAV-related function is the support to make all data sets used by DFSMSrmm eligible for allocation in the extended addressing space of an EAV. This includes the DFSMSrmm journal and dynamically allocated temporary files.
- z/OS V1.12 DFSORT:
  Information about DFSMSrmm active and queued tasks is available via the DFSMSrmm API and via a TSO/E subcommand, enabling storage applications to monitor and act on the available information. In addition you can use the DFSMSrmm dialog to manage these tasks.
- Optimization improvements for z/OS V1.12 include:
  DFSMSrmm helps with reporting of data sets and logical volumes which are copy exported from a TS7700 virtualization engine.
- The Integrated Storage Management Facility (ISMF) includes a Data Collection application, DCOLLECT, which provides storage-related measurement data that can be used as input to the DFSMSrmm Report Generator to create customized reports or to feed other applications such as billing applications. In z/OS V1.12, DCOLLECT data class (DC) records are updated to include information about all data class attributes. Also, data set (D) records now include job names, and storage group (SG) records now include information about OAM Protect Retention and Protect Deletion settings.
- In z/OS 1.12, DFSMSrmm supports IPv6.

9.2 Retention limit reporting

For the best retention limit reporting for EXPDTDROP and VRSRETAIN you need the ACTIVITY file and the extended records from the report extract file (XREPTEXT or REPTEXT DD) from the same EDGHSKP run.

During inventory management processing if an ACTIVITY file is allocated, DFSMSrmm writes information about changes in the data set and volume information, such as matching vital record specification, vital record status, retention date and catalog status to the ACTIVITY file.

Attention: If VERIFY processing is being run, the changes are not actually made.

The ACTIVITY file is not intended to be a report, but to contain detailed information about changes made to data sets during DFSMSrmm processing.

Prior to V1R12 only header and data set ACTIVITY file records were created. With V1R12 volume ACTIVITY file records are created also.

Note: The new volume ACTIVITY file records are created by:

- VRSEL only for newly assigned volumes that are to be changed from not VRS retained to either VRS retained or set retained. This is limited to volumes that are retained *only* for volume VRS, and those which are retained because of RETAINBY(SET) and another volume in the set is VRS retained.
- EXPROC only for EXPDT retained volumes that are to be set pending release.

9.2.1 Parmlib member OPTION command

Use the EDGRMMnn PARMLIB member to tailor inventory management and expiration processing.

VRSRETAIN

To aid analysis of the results of the VRSRETAIN limit checking you can use the contents of the ACTIVITY file and extended records from the extract file. The EDGJACTP sample generates a detailed report and a summary report of newly assigned volumes showing how they are processed. Refer to the contents of the VRSRETN and VRSRETNS files.

VRSDROP

To aid analysis of the results of the VRSDROP limit checking you can use the contents of the ACTIVITY file. The EDGJACTP sample generates a detailed report and a summary report of data sets dropped from VRS retention showing the reasons why. Refer to the contents of the VRS and VRSS files.

EXPDTDROP

To aid analysis of the results of the EXPDTDROP limit checking you can use the contents of the ACTIVITY file and extended records from the extract file. The EDGJACTP sample generates a detailed report and a summary report of expiration date retained volumes showing why they are set to pending release. Refer to the contents of the EXPDROP and EXPDROPS files.

9.2.2 Using the inventory management ACTIVITY file

The ACTIVITY file contains detailed information about data set and volume related changes DFSMSrmm makes to the control data set during inventory management. The ACTIVITY file is a pre-allocated DASD data set with a record length set to the largest record created by DFSMSrmm.

Figure 9-1 shows sample DFSMSrmm housekeeping JCL for creating an activity report.

```
//STEP0001 EXEC PGM=EDGHSKP,
//         PARM=('VRSEL,EXPROC,REPTEXT,DATEFORM(J)')
//SYSPRINT DD   SYSOUT=*
//MESSAGE  DD   DSN=RMM.HSKP.MESSAGE,DISP=SHR
//REPTEXT  DD   DSN=RMM.HSKP.EXTRACT,DISP=SHR
//ACTIVITY DD   DSN=RMM.HSKP.ACTIVITY,DISP=SHR
//SYSIN    DD *
 RPTEXT RECORDS(X)
/*
```

Figure 9-1 Sample DFSMSrmm housekeeping JCL

Restriction: The MESSAGE, REPTEXT, and ACTIVITY file must be pre-allocated

In the activity log you now can see the new volume related records, as shown in Figure 9-2. On column 167 you can see that all four volumes have a non-retention reason of Expired.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----
H   2010/272  143525          YNYYYNNJN       2I   6         1INYNSC70    NSS
VX  RMM.CLEANUP                                GARBDSN1THS000        2009/224
VX  RMM.F00001                                 TEST9999VT0000        2009/224
VX  RMM.F00001                                 TEST9999VT0002        2009/224
VX  RMM.F13002.ON.TWO.VOLUMES.AND.CATLG        TEST9999VT0003        2009/240

8----+----9----+----0----+----1----+----2----+----3----+----4----+----5----+---
       0 10I         0 80I         0 10IGB                                2009/
 233450LIB2                                                   1NNN Y          S
 233450VTFM001                                            12999NNN Y
 233450VTFM001                                            13002YNN Y          S
 184046VTFM001                                                1YNN Y          S

-6----+----7----+----8----+----9----+----0----+----1----+----2----+----3----+--
309  181403
     NNNX        LIB2    2009/365
 0   NNNXVTFM001 VTFM001 2009/364  2009/241  2009/241
     NNNXVTFM001 VTFM001 2009/365
     NNNXVTFM001 VTFM001 2009/240
```

Figure 9-2 Activity log volume records

Where:

Column	Description
1	H - Header
	D - Data set
	V - Volume
2	R - VRSRETAIN
	D - VRSDROP
	X - EXPDTDROP
152 and 158	S - Return to scratch
153 and 159	R - Replace
154 and 160	O - Return to owner
155 and 161	I - Initialize

156 and 162	E - Erase
157 and 163	N - Notify
164	Y/N - Retain by set
165	Y/N - Old vital record status
166	Y/N - New vital record status
167	Expiration date: I - Ignored X - Expired

Refer to the *DFSMSrmm Reporting*, SC26-7406*, Appendix B. DFSMSrmm mapping macros,* for more information.

9.2.3 Creating VRSRETAIN and EXPDTDROP Reports

EDGJACTP sample JCL, shown in Example 9-1, provides reports from a combination of ACTIVITY file volume and data set records and extended records from the report extract file.

Example 9-1 EDGJACTP sample job control

```
//EDGJACTP JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM  SYSAFF=SC70
//*
//*
//*******************************************************************
//* z/OS DFSMSrmm V1R12                                             *
//*                                                                 *
//* PROPRIETARY V3 STATEMENT                                        *
//* LICENSED MATERIALS - PROPERTY OF IBM                            *
//* "RESTRICTED MATERIALS OF IBM"                                   *
//* 5647-A01                                                        *
//* COPYRIGHT 1993,2010  IBM CORP.                                  *
//* STATUS = HDZ1C10                                                *
//* END PROPRIETARY V3 STATEMENT                                    *
//*******************************************************************
//*
//* RMM report on ACTIVITY file contents
//* ------------------------------------
//*
//* INPUT: ACTIVITY DD Statement - current RMM activity file
//*        EXTRACT  DD Statement - current RMM report extract file
//*
//* NOTE:  The ACTIVITY and EXTRACT file must have the same date format.
//*        The allowed date formats are ISO and Julian.
//*
//* OUTPUT:
//*    RUNINFO  DD Statement - Controlling options and parameters
//*    VRS      DD Statement - data sets with changed V/R status
//*    VRSS     DD Statement - summary by status
//*    RETD     DD Statement - data sets with new retention date
//*    RETDS    DD Statement - summary by new date
//*    MATCHVRS DD Statement - data sets with new matching VRS
//*    MATCHVS  DD Statement - summary by new matching VRS
//*    SUBCHN   DD Statement - data sets retained by new part of chain
//*    SUBCHNS  DD Statement - summary by new retaining subchain
//*    EXPDROP  DD Statement - EXPDT retained volumes for EXPDTDROP
```

```
//*    EXPDROPS DD Statement - summary of EXPDT retained volumes
//*    VRSRETN  DD Statement - newly assigend volumes for VRSRETAIN
//*    VRSRETNS DD Statement - summary of newly assigend volumes
//*                            for VRSRETAIN
//*
//* DEPENDANCY: DFSORT R14, for SYMNAMES support
//*
//* Change History:
//* $S5=OW24798,130,961002,MWW: Created for Audit Phase 1 SPE
//* $K1=K140934,140,970408,BDG: Corrected BREAK for drop reason    §K1A
//* $S7=OW30969,140,971217,AE : VRS Enhancements                   §S7A
//* $LF=RMM210 ,210,990805,GW : RETAINBY and MOVEBY                §LFA
//* $LG=RMM210 ,210,990924,AH : DFSORT Symbols                     §LGA
//* $K2=K160798,210,991125,CHK: Correct the SYMNAMES library name  §K2A
//* $01=OW45053,210,000616,MWW: Clean up symbols/ comment SYMNOUT  §01A
//* $02=OW54189,210,020418,AL: MSGICE218A if short var put in VRST §02A
//* $MV=V1R10  ,1RA,070613,BRB: Support limits for Release/Scratch §MVA
//* $N3=RMMGDG ,1RB,080404,GW : VRSEL GDG Options                  §N3A
//* $NK=RMMAS1 ,1RC,090224,AP : 5.1 Cleanup VRSEL(OLD)             §NKA
//* $NN=RMMARC ,1RC,090402,AP : 8.1 Reporting for EXPDTDROP        §NNA
//* $03=OA28407,1R8,090410,ZB : Add NOCHALT to SORT options        §03A
//* $NT=RMMARC ,1RC,090605,MB : 8.2 Reporting for VRSRETAIN        §NTA
//* $00=RMMAS1 ,1RC,090925,WS : Volume HOLD attribute              §00A
//*******************************************************************
//CLEAN    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE 'RMM.REPORT.RUNINFO'
  DELETE 'RMM.REPORT.VRS'
  DELETE 'RMM.REPORT.VRSS'
  DELETE 'RMM.REPORT.RETDATE'
  DELETE 'RMM.REPORT.RETDS'
  DELETE 'RMM.REPORT.MATCHVRS'
  DELETE 'RMM.REPORT.MATCHVS'
  DELETE 'RMM.REPORT.SUBCHN'
  DELETE 'RMM.REPORT.SUBCHNS'
  DELETE 'RMM.REPORT.EXPDROP'
  DELETE 'RMM.REPORT.EXPDROPS'
  DELETE 'RMM.REPORT.VRSRETN'
  DELETE 'RMM.REPORT.VRSRETNS'
  SET MAXCC = 0
/*
//*****************************************************************§NNA***
//STEP1    EXEC PGM=ICETOOL,REGION=0M
//TOOLMSG  DD   SYSOUT=*        ICETOOL MESSAGES
//DFSMSG   DD   SYSOUT=*        DFSORT  MESSAGES
//*SYMNOUT  DD SYSOUT=*
//SYMNAMES DD DSN=SYS1.MACLIB(EDGACTSY),DISP=SHR    <=====
//TOOLIN   DD   *               CONTROL STATEMENTS
  COPY    FROM(ACTIVITY) USING(DATE) TO(DATEREC1,DATEREC2,DATEREC3)
*
//ACTIVITY DD DSN=RMM.HSKP.ACTIVITY,DISP=SHR        <=====
//DATEREC1 DD DSN=&&TEMPSY1,UNIT=SYSALLDA,SPACE=(TRK,1),DISP=(,PASS),
//            DCB=(RECFM=FB,LRECL=80)
//DATEREC2 DD DSN=&&TEMPSY2,UNIT=SYSALLDA,SPACE=(TRK,1),DISP=(,PASS),
```

```
//            DCB=(RECFM=FB,LRECL=80)
//DATEREC3 DD DSN=&&TEMPSY3,UNIT=SYSALLDA,SPACE=(TRK,1),DISP=(,PASS),
//            DCB=(RECFM=FB,LRECL=80)
//DATECNTL DD *
  OUTFIL FNAMES=DATEREC1,VTOF,
   INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_HDR,AND,
            ACTRC_HDR_VRS_LAST_RUNDATE,LE,
            C'          ',AND,
            ACTRC_HDR_DATEFORM,EQ,ACTRC_HDR_DATEFORM_JULIAN),
     BUILD=(C'TEMP_SYM_HSKP_RUN_DATE,''',
            ACTRC_HDR_RUN_DATE,C'''',80:X,/,
            C'TEMP_SYM_HSKP_RUN_TIME,''',
            ACTRC_HDR_RUN_TIME,C'''',80:X,/,
            C'TEMP_SYM_LAST_VRSEL_RUN_DATE,''',
            C'9999/999 ',C'''',80:X,/,
            C'TEMP_SYM_LAST_VRSEL_RUN_TIME,''',
            ACTRC_HDR_VRS_LAST_RUNTIME,C'''',80:X)
  OUTFIL FNAMES=DATEREC2,VTOF,
   INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_HDR,AND,
            ACTRC_HDR_VRS_LAST_RUNDATE,LE,
            C'          ',AND,
            ACTRC_HDR_DATEFORM,EQ,ACTRC_HDR_DATEFORM_ISO),
     BUILD=(C'TEMP_SYM_HSKP_RUN_DATE,''',
            ACTRC_HDR_RUN_DATE,C'''',80:X,/,
            C'TEMP_SYM_HSKP_RUN_TIME,''',
            ACTRC_HDR_RUN_TIME,C'''',80:X,/,
            C'TEMP_SYM_LAST_VRSEL_RUN_DATE,''',
            C'9999/99/99',C'''',80:X,/,
            C'TEMP_SYM_LAST_VRSEL_RUN_TIME,''',
            ACTRC_HDR_VRS_LAST_RUNTIME,C'''',80:X)
  OUTFIL FNAMES=DATEREC3,VTOF,
   INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_HDR,AND,
            ACTRC_HDR_VRS_LAST_RUNDATE,GT,
            C'          '),
     BUILD=(C'TEMP_SYM_HSKP_RUN_DATE,''',
            ACTRC_HDR_RUN_DATE,C'''',80:X,/,
            C'TEMP_SYM_HSKP_RUN_TIME,''',
            ACTRC_HDR_RUN_TIME,C'''',80:X,/,
            C'TEMP_SYM_LAST_VRSEL_RUN_DATE,''',
            ACTRC_HDR_VRS_LAST_RUNDATE,C'''',80:X,/,
            C'TEMP_SYM_LAST_VRSEL_RUN_TIME,''',
            ACTRC_HDR_VRS_LAST_RUNTIME,C'''',80:X)

//*******************************************************************
//STEP2    EXEC PGM=ICETOOL,REGION=5M
//TOOLMSG  DD   SYSOUT=*        ICETOOL MESSAGES
//DFSMSG   DD   SYSOUT=*        DFSORT  MESSAGES
//*SYMNOUT  DD SYSOUT=*                                          §01C
//SYMNAMES DD DSN=SYS1.MACLIB(EDGACTSY),DISP=SHR     <=====      §K2C
//         DD DSN=SYS1.MACLIB(EDGEXTSY),DISP=SHR     <=====
//         DD DSN=SYS1.MACLIB(EDGACXSY),DISP=SHR     <=====
//         DD DSN=&TEMPSY1,DISP=(SHR,PASS)
//         DD DSN=&TEMPSY2,DISP=(SHR,PASS)
//         DD DSN=&TEMPSY3,DISP=(SHR,PASS)
//TOOLIN   DD   *               CONTROL STATEMENTS
```

```
  COPY    FROM(ACTIVITY) USING(VRST)
  COPY    FROM(EXTRACT)  USING(REPT)
  SORT    FROM(VRST)     TO(SRTDVRST)  USING(SRTV)
  SORT    FROM(RETD)     TO(SRTDRETD)  USING(SRTD)
  SORT    FROM(MTCH)     TO(SRTDMTCH)  USING(SRTM)
  SORT    FROM(SUBC)     TO(SRTDSUBC)  USING(SRTC)
  SELECT  FROM(EDRP)     TO(SRTDEDR1)  USING(SRTE) -
          FIRST ON(ACXSYC_EXPDTDROP_VOL)
  SORT    FROM(SRTDEDR1) TO(SRTDEDR2)  USING(SRTF)
  SELECT  FROM(VRTNTMP2) TO(VRTNTMP1)  USING(SRT1) -
          FIRST ON(ACXSYC_VRSRETAIN_VSKEY)
  SORT    FROM(VRTNTMP1) TO(VRTNTMP3)  USING(SRT2)
*********************************************************
*  SPLICE TO COPY VOLUME RETENTION GROUP AND VOLUME VRS  *
*  FROM FIRST RECORD TO ALL RECORDS WITH SAME VOLSER     *
*********************************************************
  SPLICE  FROM(VRTNTMP3) TO(VRTNTMP4) -
          ON(ACXSYC_VRSRETAIN_VOL) -
          WITH(ACXSYC_VRSRETAIN_REC_TYPE1) -
          WITH(ACXSYC_VRSRETAIN_REC_TYPE2) -
          WITH(ACXSYC_VRSRETAIN_DSGRP) -
          WITH(ACXSYC_VRSRETAIN_DSVRSGRP) -
          WITH(ACXSYC_VRSRETAIN_NEW_REASON) -
          WITH(ACXSYC_VRSRETAIN_DSNNO) -
          WITH(ACXSYC_VRSRETAIN_NEW_INSET) -
          KEEPNODUPS KEEPBASE WITHALL
  COPY    FROM(VRTNTMP4) USING(SPL1)
  COPY    FROM(VRTNTMP5) USING(SPL2)
  COPY    FROM(VRTNTMP6) USING(SPL3)
  SORT    FROM(VRTNTMP7) TO(SRTDVRTN)  USING(SRT3)
  SELECT  FROM(VRTNTMP7) TO(SRTDVRTS) -
          FIRST ON(ACXSYC_VRSRETAIN_VOL)
*
*
*
  DISPLAY FROM(SRTDVRST) LIST(VRS) -
     TITLE('Data Sets Changed VRS Status') DATE TIME PAGE -
     BLANK -
     BTITLE('Status Change and Drop Reason:') -
     BREAK(ACTRC_OUTFIL_VITALANDDROP) -
     HEADER('DSNAME')       ON(ACTRC_DSN_DSNAME) -
     HEADER('JOBNAME')      ON(ACTRC_DSN_JOBNAME) -
     HEADER('VOLSER')       ON(ACTRC_DSN_VOL) -
     HEADER('O-ST')         ON(ACTRC_DSN_OLD_VITAL) -
     HEADER('N-ST')         ON(ACTRC_DSN_NEW_VITAL) -
     HEADER('RSN')          ON(ACTRC_DSN_DROP) -
     HEADER('PRIMARY VRS')  ON(ACTRC_DSN_NEW_MMASK) -
     HEADER('JOB MASK')     ON(ACTRC_DSN_NEW_MJOB) -
     HEADER('TYPE')         ON(ACTRC_DSN_NEW_MTYPE)
*
  OCCUR FROM(SRTDVRST) LIST(VRSS) -
     TITLE('Data Set VRS status change summary') -
     DATE TIME PAGE -
     BLANK -
     HEADER('Status Change') ON(ACTRC_OUTFIL_VITAL) -
```

```
     HEADER('Drop Reason')   ON(ACTRC_OUTFIL_DROP) -
     HEADER('COUNT') ON(VALCNT)
*
*
  DISPLAY FROM(SRTDRETD) LIST(RETDATE) -
     TITLE('Data Sets Changed Retention Date') DATE TIME PAGE -
     BLANK -
     BTITLE('New Retention Date:') BREAK(ACTRC_DSN_NEW_RETDATE) -
     HEADER('DSNAME')       ON(ACTRC_DSN_DSNAME) -
     HEADER('JOBNAME')      ON(ACTRC_DSN_JOBNAME) -
     HEADER('VOLSER')       ON(ACTRC_DSN_VOL) -
     HEADER('PREVIOUS')     ON(ACTRC_DSN_OLD_RETDATE) -
     HEADER('NEW DATE')     ON(ACTRC_DSN_NEW_RETDATE) -
     HEADER('PRIMARY VRS')  ON(ACTRC_DSN_NEW_MMASK) -
     HEADER('JOB MASK')     ON(ACTRC_DSN_NEW_MJOB) -
     HEADER('TYPE')         ON(ACTRC_DSN_NEW_MTYPE) -
     HEADER('SUBCHAIN')     ON(ACTRC_DSN_NEW_MNAME)
*
  OCCUR FROM(SRTDRETD) LIST(RETDS) -
     TITLE('Summary of new Data Set retention dates') -
     DATE TIME PAGE -
     BLANK -
     HEADER('New Retention Date') ON(ACTRC_DSN_NEW_RETDATE) -
     HEADER('COUNT') ON(VALCNT)
*
*
  DISPLAY FROM(SRTDMTCH) LIST(MATCHVRS) -
     TITLE('Data Sets Matching to different VRS') DATE TIME PAGE -
     BLANK -
     BTITLE('NEW PRIMARY VRS:') BREAK(ACTRC_DSN_NEW_VRSS) -
     HEADER('DSNAME')           ON(ACTRC_DSN_DSNAME) -
     HEADER('JOBNAME')          ON(ACTRC_DSN_JOBNAME) -
     HEADER('VOLSER')           ON(ACTRC_DSN_VOL) -
     HEADER('O-ST')             ON(ACTRC_DSN_OLD_VITAL) -
     HEADER('N-ST')             ON(ACTRC_DSN_NEW_VITAL) -
     HEADER('DROPRSN')          ON(ACTRC_DSN_DROP) -
     HEADER('OLD PRIMARY VRS')  ON(ACTRC_DSN_OLD_MMASK) -
     HEADER('JOB MASK')         ON(ACTRC_DSN_OLD_MJOB) -
     HEADER('TYPE')             ON(ACTRC_DSN_OLD_MTYPE) -
     HEADER('2nd. VRS')         ON(ACTRC_DSN_OLD_M2MASK) -
     HEADER('2nd. JOB')         ON(ACTRC_DSN_OLD_M2JOB)
*
  OCCUR FROM(SRTDMTCH) LIST(MATCHVS) -
     TITLE('Summary of new matching VRSs') -
     DATE TIME PAGE -
     BLANK -
     HEADER('New Primary VRS')  ON(ACTRC_DSN_NEW_MMASK) -
     HEADER('Jobname mask')     ON(ACTRC_DSN_NEW_MJOB) -
     HEADER('Match Type')       ON(ACTRC_DSN_NEW_MTYPE) -
     HEADER('COUNT') ON(VALCNT)
*
*
  DISPLAY FROM(SRTDSUBC) LIST(SUBCHN) -
     TITLE('Data Sets Changed VRS Subchain') DATE TIME PAGE -
     BLANK -
```

```
      BTITLE('NEW SUBCHAIN AND DATE:') BREAK(ACTRC_DSN_NEW_CHAINS) -
      HEADER('DSNAME')           ON(ACTRC_DSN_DSNAME) -
      HEADER('JOBNAME')          ON(ACTRC_DSN_JOBNAME) -
      HEADER('VOLSER')           ON(ACTRC_DSN_VOL) -
      HEADER('PRIMARY VRS')      ON(ACTRC_DSN_NEW_MMASK) -
      HEADER('JOB MASK')         ON(ACTRC_DSN_NEW_MJOB) -
      HEADER('TYPE')             ON(ACTRC_DSN_NEW_MTYPE) -
      HEADER('2nd.VRS JOB')      ON(ACTRC_DSN_NEW_M2MATCH) -
      HEADER('SUBCHAIN DATE     2nd.SUBC DATE') ON(ACTRC_DSN_OLD_CHAINS)
*
  OCCUR FROM(SRTDSUBC) LIST(SUBCHNS) -
      TITLE('Summary of new Data Set subchains') -
      DATE TIME PAGE -
      BLANK -
      HEADER('New Subchain') ON(ACTRC_DSN_NEW_CHAINS) -
      HEADER('COUNT') ON(VALCNT)
*
*
  DISPLAY FROM(SRTDEDR2) LIST(EXPDROP) -
      TITLE('EXPDT retained volumes subject to EXPDTDROP') -
      DATE TIME PAGE -
      BLANK -
      BTITLE('Status:')          BREAK(ACXSYC_PRE_TYPE) -
      BCOUNT('Volumes in this status:') EDBCOUNT(U15) -
      HEADER('VOLSER')           ON(ACXSYC_EXPDTDROP_VOL) -
      HEADER('VSEQ')             ON(ACXSYC_EXPDTDROP_VOLSEQ) -
      HEADER('DSNAME')           ON(ACXSYC_EXPDTDROP_DSNAME) -
      HEADER('JOBNAME')          ON(ACXSYC_EXPDTDROP_JOBNAME) -
      HEADER('EXPRSN')           ON(ACXSYC_EXPDTDROP_EXPRSN) -
      HEADER('ASSIGNED')         ON(ACXSYC_EXPDTDROP_ASSIGNED) -
      HEADER('EXPDT')            ON(ACXSYC_EXPDTDROP_EXPDT) -
      HEADER('SR')               ON(ACXSYC_EXPDTDROP_SR) -
      HEADER('RETDATE')          ON(ACXSYC_EXPDTDROP_RETDATE) -
      HEADER('ACTIONS')          ON(ACXSYC_EXPDTDROP_ACTIONS) -
      HEADER('LOCATION')         ON(ACXSYC_EXPDTDROP_LOCATION) -
      HEADER('HOME')             ON(ACXSYC_EXPDTDROP_HOME) -
      HEADER('DEST')             ON(ACXSYC_EXPDTDROP_DEST) -
      HEADER('RLS ACT')          ON(ACXSYC_EXPDTDROP_RLS_ACT) -
      HEADER('HOLD')             ON(ACXSYC_EXPDTDROP_HOLD)
*
  OCCUR FROM(SRTDEDR2) LIST(EXPDROPS) -
      TITLE('Summary of EXPDT retained volumes for EXPDTDROP') -
      DATE TIME PAGE -
      BLANK -
      HEADER('Status') ON(ACXSYC_PRE_TYPE) -
      HEADER('VOLUME COUNT') ON(VALCNT)
*
*
  DISPLAY FROM(SRTDVRTN) LIST(VRSRETN) -
      BETWEEN(1) -
      TITLE('Newly assigned volumes subject to VRSRETAIN') -
      DATE TIME PAGE -
      BLANK -
      BTITLE('Status:')         BREAK(ACXSYC_PRE_TYPE) -
      BCOUNT('data sets in this status:') EDBCOUNT(U15) -
```

```
     HEADER('D A T A',' ','VOLSER') -
                              ON(ACXSYC_VRSRETAIN_VOL) -
     HEADER('S E T',' ','FSEQ') -
                              ON(ACXSYC_VRSRETAIN_FSEQ) -
     HEADER(' ',' ','DSNAME') -
                              ON(ACXSYC_VRSRETAIN_DSNAME) -
     HEADER(' ',' ','JOBNAME') -
                              ON(ACXSYC_VRSRETAIN_JOBNAME) -
     HEADER('D A T A S',' ','RETAINED') -
                              ON(ACXSYC_VRSRETAIN_NEW_VITAL) -
     HEADER('E T  ','DROP','PRIM') -
                              ON(ACXSYC_VRSRETAIN_SUBCHAIN_DROP) -
     HEADER('V R S','REASON','2nd') -
                              ON(ACXSYC_VRSRETAIN_2SUBCHAIN_DROP) -
     HEADER(' ',' ','PRIMARY VRS') -
                              ON(ACXSYC_VRSRETAIN_NEW_MMASK) -
     HEADER(' ',' ','JOB MASK') -
                              ON(ACXSYC_VRSRETAIN_NEW_MJOB) -
     HEADER(' ','VRS','TYPE') -
                              ON(ACXSYC_VRSRETAIN_NEW_MTYPE) -
     HEADER('V O L U M E',' ','VRS     ') -
                              ON(ACXSYC_VRSRETAIN_VOLVRS) -
     HEADER(' ','RETAIN','REASON') -
                              ON(ACXSYC_VRSRETAIN_NEW_REASON) -
     HEADER(' ','FILE','COUNT') -
                              ON(ACXSYC_VRSRETAIN_DSNNO) -
     HEADER(' ','IN','SET') -
                              ON(ACXSYC_VRSRETAIN_NEW_INSET)
*
OCCUR FROM(SRTDVRTS) LIST(VRSRETNS) -
     TITLE('Summary of newly assigned volumes for VRSRETAIN') -
     DATE TIME PAGE -
     BLANK -
     HEADER('Status') ON(ACXSYC_PRE_TYPE) -
     HEADER('VOLUME COUNT') ON(VALCNT)
*
//ACTIVITY DD DSN=RMM.HSKP.ACTIVITY,DISP=SHR      <======
//EXTRACT  DD DSN=RMM.HSKP.EXTRACT,DISP=SHR       <======
//VRST     DD DSN=&&TEMPV1,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//RETD     DD DSN=&&TEMPD1,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//MTCH     DD DSN=&&TEMPM1,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//SUBC     DD DSN=&&TEMPC1,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//EDRP     DD DSN=&&TEMPE1,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA,
//         DISP=(MOD,PASS)
//VRTN     DD DSN=&&TEMPU1,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//VRTNTMP1 DD DSN=&&TEMPP1,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA,
//         DISP=(MOD,PASS)
//VRTNTMP2 DD DSN=&&TEMPP2,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA,
//         DISP=(MOD,PASS)
//VRTNTMP3 DD DSN=&&TEMPP3,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//VRTNTMP4 DD DSN=&&TEMPP4,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//VRTNTMP5 DD DSN=&&TEMPP5,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//VRTNTMP6 DD DSN=&&TEMPP6,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//VRTNTMP7 DD DSN=&&TEMPP7,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//SRTDVRST DD DSN=&&TEMPV2,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
```

```
//SRTDRETD DD DSN=&&TEMPD2,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//SRTDMTCH DD DSN=&&TEMPM2,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//SRTDSUBC DD DSN=&&TEMPC2,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//SRTDEDR1 DD DSN=&&TEMPE2,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//SRTDEDR2 DD DSN=&&TEMPE3,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//SRTDVRTN DD DSN=&&TEMPU2,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//SRTDVRTS DD DSN=&&TEMPU3,SPACE=(CYL,(10,10),RLSE),UNIT=SYSALLDA
//VRSTCNTL DD *
  OUTFIL FNAMES=RUNINFO,
   INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_HDR),
   HEADER2=(1:'Run Control Information',2X,DATE,2X,TIME,2X,PAGE,/,
           1:'Run Date',
          12:'Date Used',
          23:'Run Time',
          32:'Options',
          72:'DATEFORM',
          81:'VRSJOBNAME',
          92:'VRSCHANGE',
         102:'CATRETPD',
         111:'VRSMIN Count',
         124:'VRSMIN Action',
         138:'VRSEL',
         144:'UNCATALOG',
         154:'TPRACF',
         161:'SYSID',
         170:'CATSYSID',
         179:'RETAINBY',
         188:'MOVEBY',
         197:'VRSDROP Count',
         211:'VRSDROP Prcnt',
         225:'VRSDROP Action',
         240:'VRSRETAIN Count',
         256:'VRSRETAIN Prcnt',
         272:'VRSRETAIN Action',
         289:'EXPDTDROP Count',
         305:'EXPDTDROP Prcnt',
         321:'EXPDTDROP Action',
         338:'GDG CYCLEBY',
         350:'GDG DUPLICATE',/,
           1:10'-',
          12:10'-',
          23:8'-',
          32:33'-',
          72:8'-',
          81:10'-',
          92:9'-',
         102:8'-',
         111:12'-',
         124:13'-',
         138:5'-',
         144:9'-',
         154:6'-',
         161:8'-',
         170:8'-',
         179:8'-',
```

```
     188:8'-',
     197:13'-',
     211:13'-',
     225:14'-',
     240:15'-',
     256:15'-',
     272:16'-',
     289:15'-',
     305:15'-',
     321:16'-',
     338:11'-',
     350:13'-'),
OUTREC=(ACTRC_RDW,
      5:ACTRC_HDR_RUN_DATE,
     16:ACTRC_HDR_VERIFY_DATE,
     27:ACTRC_HDR_RUN_TIME,
     36:ACTRC_HDR_BACKUP,CHANGE=(6,ACTRC_YES,C'BACKUP',
                                 ACTRC_NO,C' '),X,
        ACTRC_HDR_DSTORE,CHANGE=(6,ACTRC_YES,C'DSTORE',
                                 ACTRC_NO,C' '),X,
        ACTRC_HDR_EXPROC,CHANGE=(6,ACTRC_YES,C'EXPROC',
                                 ACTRC_NO,C' '),X,
        ACTRC_HDR_RPTEXT,CHANGE=(6,ACTRC_YES,C'RPTEXT',
                                 ACTRC_NO,C' '),X,
        ACTRC_HDR_VRSEL,CHANGE=(5,ACTRC_YES,C'VRSEL',
                                 ACTRC_NO,C' '),X,
        ACTRC_HDR_DATE,CHANGE=(4,ACTRC_YES,C'DATE',
                                 ACTRC_NO,C' '),
     76:ACTRC_HDR_DATEFORM,CHANGE=(8,
           ACTRC_HDR_DATEFORM_AMERICAN,C'AMERICAN',
           ACTRC_HDR_DATEFORM_EUROPEAN,C'EUROPEAN',
           ACTRC_HDR_DATEFORM_ISO,C'ISO',
           ACTRC_HDR_DATEFORM_JULIAN,C'JULIAN'),X,
     85:ACTRC_HDR_VRSJOBNAME,
     96:ACTRC_HDR_VRSCHANGE,CHANGE=(6,
           ACTRC_HDR_VRSCHANGE_VERIFY,C'VERIFY',
           ACTRC_HDR_VRSCHANGE_INFO,C'INFO'),
    106:ACTRC_HDR_CATRETPD,
    115:ACTRC_HDR_VRSMIN_COUNT,
    128:ACTRC_HDR_VRSMIN_ACTION,CHANGE=(4,
           ACTRC_HDR_VRSMIN_ACTION_FAIL,C'FAIL',
           ACTRC_HDR_VRSMIN_ACTION_WARN,C'WARN',
           ACTRC_HDR_VRSMIN_ACTION_INFO,C'INFO',
           ACTRC_HDR_VRSMIN_ACTION_OFF,C'OFF '),
    142:ACTRC_HDR_OPT_VRSEL,CHANGE=(3,
           ACTRC_HDR_OPT_VRSEL_NEW,C'NEW',
           ACTRC_HDR_OPT_VRSEL_BLANK,C'NEW'),
    148:ACTRC_HDR_UNCATALOG,
    158:ACTRC_HDR_TPRACF,
    165:ACTRC_HDR_SYSID,
    174:ACTRC_HDR_CATSYSID,CHANGE=(8,
           ACTRC_HDR_CATSYSID_NOT_SET,C'NOT SET',
           ACTRC_HDR_CATSYSID_SET,C'SET',
           ACTRC_HDR_CATSYSID_SHARED,C'SHARED'),
    183:ACTRC_HDR_OPT_RETAINBY,CHANGE=(8,
```

```
            C' ',C' ',
            ACTRC_HDR_OPT_RETAINBY_VOLUME,C'VOLUME',
            ACTRC_HDR_OPT_RETAINBY_SET,C'SET'),
      192:ACTRC_HDR_OPT_MOVEBY,CHANGE=(8,
            C' ',C' ',
            ACTRC_HDR_OPT_MOVEBY_VOLUME,C'VOLUME',
            ACTRC_HDR_OPT_MOVEBY_SET,C'SET'),
      201:ACTRC_HDR_VRSDROP_COUNT,
      215:ACTRC_HDR_VRSDROP_PERCENT,
      229:ACTRC_HDR_VRSDROP_ACTION,CHANGE=(4,
            ACTRC_HDR_VRSDROP_ACTION_FAIL,C'FAIL',
            ACTRC_HDR_VRSDROP_ACTION_WARN,C'WARN',
            ACTRC_HDR_VRSDROP_ACTION_INFO,C'INFO',
            ACTRC_HDR_VRSDROP_ACTION_OFF,C'OFF '),
      244:ACTRC_HDR_VRSRETAIN_COUNT,
      260:ACTRC_HDR_VRSRETAIN_PERCENT,
      276:ACTRC_HDR_VRSRETAIN_ACTION,CHANGE=(4,
            ACTRC_HDR_VRSRETAIN_ACTION_FAIL,C'FAIL',
            ACTRC_HDR_VRSRETAIN_ACTION_WARN,C'WARN',
            ACTRC_HDR_VRSRETAIN_ACTION_INFO,C'INFO',
            ACTRC_HDR_VRSRETAIN_ACTION_OFF,C'OFF '),
      293:ACTRC_HDR_EXPDTDROP_COUNT,
      309:ACTRC_HDR_EXPDTDROP_PERCENT,
      325:ACTRC_HDR_EXPDTDROP_ACTION,CHANGE=(16,
            ACTRC_HDR_EXPDTDROP_ACTION_FAIL,C'FAIL',
            ACTRC_HDR_EXPDTDROP_ACTION_WARN,C'WARN',
            ACTRC_HDR_EXPDTDROP_ACTION_INFO,C'INFO',
            ACTRC_HDR_EXPDTDROP_ACTION_OFF,C'OFF '),
      342:ACTRC_HDR_GDGCYCLEBY,CHANGE=(10,
            ACTRC_HDR_GDGC_GENERATION,C'GENERATION',
            ACTRC_HDR_GDGC_CRDATE,C'CRDATE'),
      354:ACTRC_HDR_GDGDUPLICATE,CHANGE=(13,
            ACTRC_HDR_GDGD_BUMP,C'BUMP',
            ACTRC_HDR_GDGD_DROP,C'DROP',
            ACTRC_HDR_GDGD_KEEP,C'KEEP',
            ACTRC_HDR_GDGD_COUNT,C'COUNT'))

OUTFIL FNAMES=VRST,VLFILL=C' ',
  INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_DSN,AND,
           ACTRC_DSN_CHNG_VRS,EQ,ACTRC_YES),
  OUTREC=(ACTRC_RDW,
            ACTRC_PREFIX,
            ACTRC_DSN_DATA,
            ACTRC_DSN_VITAL,CHANGE=(9,
              ACTRC_DSN_VITAL_NY,ACTRC_DSN_VITAL_RETAIN,
              ACTRC_DSN_VITAL_YN,ACTRC_DSN_VITAL_DROPPED),
            ACTRC_DSN_DROP,CHANGE=(13,
              ACTRC_DSN_DROP_WHILECATALOG,C'WHILECATALOG',
              ACTRC_DSN_DROP_BLANK,C' ',
              ACTRC_DSN_DROP_UNTILEXPIRED,C'UNTILEXPIRED',
              ACTRC_DSN_DROP_CYCLES,C'CYCLES',
              ACTRC_DSN_DROP_DAYS,C'DAYS',
              ACTRC_DSN_DROP_LASTREF,C'LASTREFDAYS',
              ACTRC_DSN_DROP_EXTRADAYS,C'EXTRADAYS',
              ACTRC_DSN_DROP_BYDAYSCYCLE,C'BYDAYSCYCLE',
```

```
             ACTRC_DSN_DROP_NO_MATCH,C'NO MATCH',
             ACTRC_DSN_DROP_DUP_GDG,C'DUPL. GDG',
             ACTRC_DSN_DROP_VOL_RELEASED,C'VOL RELEASED'))

OUTFIL  FNAMES=RETD,
 INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_DSN,AND,
          ACTRC_DSN_CHNG_RETDATE,EQ,ACTRC_YES)

OUTFIL  FNAMES=MTCH,
 INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_DSN,AND,
          ACTRC_DSN_CHNG_MATCH,EQ,ACTRC_YES)

OUTFIL  FNAMES=SUBC,
 INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_DSN,AND,
          ACTRC_DSN_CHNG_SUBCHAIN,EQ,ACTRC_YES)

OUTFIL FNAMES=EDRP,VLFILL=C' ',
   INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_VOL,AND,
            ACTRC_PRE_RETENTION_GROUP,EQ,
            ACTRC_PRE_RETENTION_GROUP_EXPDTDROP),
   OUTREC=(ACXSYC_RDW,
         5:ACXSYC_PRE_TYPE_RELEASED,
        16:ACTRC_VOL_VOL,
        22:ACTRC_VOL_VSEQ,
        27:ACTRC_VOL_DSNAME,
        71:ACTRC_VOL_JOBNAME,
        79:ACTRC_VOL_DROP,
        80:ACTRC_VOL_ASDATE,
        90:ACTRC_VOL_EXPDT,
       100:ACTRC_VOL_RETAIN_BY_SET,
       101:ACTRC_VOL_NEW_RETDATE,
       111:ACTRC_VOL_ACTIONS_PENDING,
       117:ACTRC_VOL_LOC,
       125:ACTRC_VOL_HOME_LOC,
       133:ACTRC_VOL_DEST,
       141:ACTRC_VOL_ACTIONS_RELEASE,
       147:ACTRC_VOL_HOLD,
       148:C'  ')

 OUTFIL FNAMES=VRTNTMP1,VLFILL=C' ',
    INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_VOL,AND,
             ACTRC_PRE_RETENTION_GROUP,EQ,
             ACTRC_PRE_RETENTION_GROUP_VRSRETAIN),
    OUTREC=(ACXSYC_RDW,
         ACXSYC_VRSRETAIN_REC_TYPE1:C'V',
         ACXSYC_VRSRETAIN_REC_TYPE2:C'A',
         ACXSYC_VRSRETAIN_VOL:ACTRC_VOL_VOL,
         ACXSYC_VRSRETAIN_FSEQ:C'    1',
         ACXSYC_VRSRETAIN_DSNAME:ACTRC_VOL_DSNAME,
         ACXSYC_VRSRETAIN_JOBNAME:ACTRC_VOL_JOBNAME,
         ACXSYC_VRSRETAIN_VOLVRS:ACTRC_VOL_NEW_MMASK,
         ACXSYC_VRSRETAIN_DSNNO:ACTRC_VOL_DSNNO,
         ACXSYC_VRSRETAIN_NEW_INSET:ACTRC_VOL_INSET,
         ACXSYC_VRSRETAIN_VOL_RET:C'Y',
         ACXSYC_VRSRETAIN_VOL_ROBS:ACTRC_VOL_RETAIN_BY_SET)
```

```
   OUTFIL FNAMES=VRTNTMP2,VLFILL=C' ',
     INCLUDE=(ACTRC_PRE_TYPE,EQ,ACTRC_PRE_TYPE_DSN,AND,
              ACTRC_PRE_RETENTION_GROUP,EQ,
              ACTRC_PRE_RETENTION_GROUP_VRSRETAIN),
     OUTREC=(ACXSYC_RDW,
            ACXSYC_VRSRETAIN_REC_TYPE1:C'D',
            ACXSYC_VRSRETAIN_REC_TYPE2:C'A',
            ACXSYC_VRSRETAIN_VOL:ACTRC_DSN_VOL,
            ACXSYC_VRSRETAIN_FSEQ:ACTRC_DSN_FILESEQ,
            ACXSYC_VRSRETAIN_DSNAME:ACTRC_DSN_DSNAME,
            ACXSYC_VRSRETAIN_JOBNAME:ACTRC_DSN_JOBNAME,
            ACXSYC_VRSRETAIN_NEW_VITAL:ACTRC_DSN_NEW_VITAL,
            ACXSYC_VRSRETAIN_SUBCHAIN_DROP:ACTRC_DSN_SUBCHAIN_DROP,
                                           CHANGE=(1,C'N',C' '),
                       NOMATCH=(ACTRC_DSN_SUBCHAIN_DROP),
            ACXSYC_VRSRETAIN_2SUBCHAIN_DROP:ACTRC_DSN_2SUBCHAIN_DROP,
                                           CHANGE=(1,C'N',C' '),
                       NOMATCH=(ACTRC_DSN_2SUBCHAIN_DROP),
            ACXSYC_VRSRETAIN_NEW_MMASK:ACTRC_DSN_NEW_MMASK,
            ACXSYC_VRSRETAIN_NEW_MJOB:ACTRC_DSN_NEW_MJOB,
            ACXSYC_VRSRETAIN_NEW_MTYPE:ACTRC_DSN_NEW_MTYPE,
            ACXSYC_VRSRETAIN_DSNNO:ACTRC_DSN_VOL_DSNNO,
            ACXSYC_VRSRETAIN_NEW_INSET:ACTRC_DSN_VOL_INSET,
            ACXSYC_VRSRETAIN_VOL_RET:ACTRC_DSN_NEW_VITAL,
            ACXSYC_VRSRETAIN_VOL_ROBS:C'N')

  OPTION VLSHRT,NOCHALT
//REPTCNTL DD *
   OUTFIL FNAMES=EDRP,VLFILL=C' ',
     INCLUDE=(EXTRACT_TYPID,EQ,RXTYPEID,AND,
              (RXVOLTYPE,EQ,RXVOLTYPE_PHYSICAL,OR,
               RXVOLTYPE,EQ,RXVOLTYPE_LOGICAL),AND,
              (RXVSTATUS,EQ,RXVMST,OR,RXVSTATUS,EQ,RXVUSR),AND,
              RXDDSNSEQ,LE,C'1',AND,
              RXVPENDRS,EQ,RXVNO,AND,
              RXVVRS,EQ,RXVNO,AND,
              RXVRETDAT,LT,ACTRC_HDR_RUN_DATE,AND,
              ((RXVLCDATE,LT,TEMP_SYM_HSKP_RUN_DATE,OR,
                (RXVLCDATE,EQ,TEMP_SYM_HSKP_RUN_DATE,AND,
                 RXVLCTIME,LT,TEMP_SYM_HSKP_RUN_TIME)),OR,
               RXVDSNAM1,EQ,C' ',OR,
               RXVSTDATE,GE,ACTRC_HDR_RUN_DATE),AND,
              (RXVASDATE,LT,TEMP_SYM_LAST_VRSEL_RUN_DATE,OR,
               (RXVASDATE,EQ,TEMP_SYM_LAST_VRSEL_RUN_DATE,AND,
                RXVASTIME,LT,TEMP_SYM_LAST_VRSEL_RUN_TIME))),
     OUTREC=(ACXSYC_RDW,
              5:ACXSYC_PRE_TYPE_NOCHANGE,
             16:RXVOLSER,
             22:RXVVOLSEQ,
             27:RXVDSNAM1,
             71:RXVCRJOB,
             79:C' ',   ,
             80:RXVASDATE,
             90:RXVEXPDT,
```

```
      100:RXVRBYSET,
      101:RXVRETDAT,
      111:RXVACTSCR,CHANGE(1,
            RXVYES,C'S',
            RXVNO,C' '),
      112:RXVACTREP,CHANGE(1,
            RXVYES,C'R',
            RXVNO,C' '),
      113:RXVACTRET,CHANGE(1,
            RXVYES,C'O',
            RXVNO,C' '),
      114:RXVACTINI,CHANGE(1,
            RXVYES,C'I',
            RXVNO,C' '),
      115:RXVACTERA,CHANGE(1,
            RXVYES,C'E',
            RXVNO,C' '),
      116:RXVACTNOT,CHANGE(1,
            RXVYES,C'N',
            RXVNO,C' '),
      117:RXVSTORID,
      125:RXVHLOC,
      133:RXVDEST,
      141:RXVRETS,CHANGE=(1,
            RXVSCR,C'S',
            RXVOWN,C' ',
            C' ',C' '),
      142:RXVREPL,CHANGE=(1,
            RXVYES,C'R',
            RXVNO,C' '),
      143:RXVRETS,CHANGE=(1,
            RXVSCR,C' ',
            RXVOWN,C'O',
            C' ',C' '),
      144:RXVINIT,CHANGE=(1,
            RXVYES,C'I',
            RXVNO,C' '),
      145:RXVERASE,CHANGE=(1,
            RXVYES,C'E',
            RXVNO,C' '),
      146:RXVNTFY,CHANGE=(1,
            RXVYES,C'N',
            RXVNO,C' '),
      147:RXVHOLD,
      148:C'  ')

OUTFIL FNAMES=VRTNTMP2,VLFILL=C' ',
  INCLUDE=(EXTRACT_TYPID,EQ,RXTYPEID,AND,
           (RXVSTATUS,EQ,RXVMST,OR,RXVSTATUS,EQ,RXVUSR),AND,
           (RXVOLTYPE,EQ,RXVOLTYPE_PHYSICAL,OR,
            RXVOLTYPE,EQ,RXVOLTYPE_LOGICAL),AND,
            (RXVASDATE,GT,TEMP_SYM_LAST_VRSEL_RUN_DATE,OR,
            (RXVASDATE,EQ,TEMP_SYM_LAST_VRSEL_RUN_DATE,AND,
             RXVASTIME,GT,TEMP_SYM_LAST_VRSEL_RUN_TIME))),
   OUTREC=(ACXSYC_RDW,
```

```
            ACXSYC_VRSRETAIN_REC_TYPE1:C'D',
            ACXSYC_VRSRETAIN_REC_TYPE2:C'X',
            ACXSYC_VRSRETAIN_VOL:RXVOLSER,
            ACXSYC_VRSRETAIN_FSEQ:RXDDSNSEQ,
            ACXSYC_VRSRETAIN_DSNAME:RXDSNAME,
            ACXSYC_VRSRETAIN_JOBNAME:RXDCRTJBN,
            ACXSYC_VRSRETAIN_NEW_VITAL:RXDVRSR,
            ACXSYC_VRSRETAIN_NEW_MMASK:RXDVRSNAM,
            ACXSYC_VRSRETAIN_NEW_MJOB:RXDVRSJBN,
            ACXSYC_VRSRETAIN_NEW_MTYPE:RXDVRSTYP,
            ACXSYC_VRSRETAIN_DSNNO:RXVDSNNO,
            ACXSYC_VRSRETAIN_NEW_INSET:RXADJVOL,CHANGE=(1,
                                       C'            ',C'N'),
                                       NOMATCH=(C'Y'),
            ACXSYC_VRSRETAIN_VOL_RET:RXDVRSR,CHANGE=(1,
                                       C'Y',C'Y'),
                                       NOMATCH=(C'N'),
            ACXSYC_VRSRETAIN_VOL_ROBS:C'N')

  OPTION VLSHRT
//SPL1CNTL DD *
  OUTFIL FNAMES=VRTNTMP5,
  INCLUDE=(ACXSYC_VRSRETAIN_REC_TYPE1,EQ,C'D')
  OUTREC IFTHEN=(WHEN=((ACXSYC_VRSRETAIN_VOL_RET,EQ,C'Y'),AND,
                       (ACXSYC_VRSRETAIN_VOLVRS,CH,NE,C'     ')),
            OVERLAY=(ACXSYC_VRSRETAIN_NEW_REASON:C'VOLUME  ')),
         IFTHEN=(WHEN=((ACXSYC_VRSRETAIN_VOL_RET,EQ,C'Y'),AND,
                       (ACXSYC_VRSRETAIN_VOL_ROBS,CH,EQ,C'Y')),
            OVERLAY=(ACXSYC_VRSRETAIN_NEW_REASON:C'SET     ')),
         IFTHEN=(WHEN=((ACXSYC_VRSRETAIN_VOL_RET,EQ,C'Y'),AND,
                       (ACXSYC_VRSRETAIN_NEW_VITAL,CH,EQ,C'Y')),
            OVERLAY=(ACXSYC_VRSRETAIN_NEW_REASON:C'DATASET ')),
         IFTHEN=(WHEN=(ACXSYC_VRSRETAIN_VOL_RET,EQ,C'Y'),
            OVERLAY=(ACXSYC_VRSRETAIN_NEW_REASON:C'IMPLICIT'))

  OPTION VLSHRT
//SPL2CNTL DD *
  OUTFIL FNAMES=VRTNTMP6
  OUTREC IFTHEN=(WHEN=(ACXSYC_VRSRETAIN_FSEQ,NE,C' ',AND,
                       ACXSYC_VRSRETAIN_NEW_MMASK,EQ,C' '),
            OVERLAY=(ACXSYC_VRSRETAIN_NEW_VITAL:C' ')),
         IFTHEN=(WHEN=(ACXSYC_VRSRETAIN_FSEQ,EQ,C' '),
            OVERLAY=(ACXSYC_VRSRETAIN_FSEQ:C'    1'))

  OPTION VLSHRT ACXSYC_VRSRETAIN_FSEQ,NE,C' '           ),
//SPL3CNTL DD *
  OUTFIL FNAMES=VRTNTMP7
  OUTREC IFTHEN=(WHEN=(ACXSYC_VRSRETAIN_VOL_RET,EQ,C'Y'),
            OVERLAY=(ACXSYC_PRE_TYPE:C'RETAINED   ')),
         IFTHEN=(WHEN=(ACXSYC_VRSRETAIN_VOL_RET,EQ,C'N'),
            OVERLAY=(ACXSYC_PRE_TYPE:C'NOTRETAINED'))

  OPTION VLSHRT
//SRTVCNTL DD *
* Sort on status change fields
```

```
   SORT FIELDS=(ACTRC_DSN_OLD_VITAL,A,
                ACTRC_DSN_NEW_VITAL,A,
                ACTRC_DSN_DROP,A)
//SRTDCNTL DD *
* Sort on new retention date
  SORT FIELDS=(ACTRC_DSN_NEW_RETDATE,A)
//SRTMCNTL DD *          DFSORT STATEMENTS - SORT AND REFORMAT
* Sort on new matching VRS fields
  SORT FIELDS=(ACTRC_DSN_NEW_MTYPE,A,
               ACTRC_DSN_NEW_MMASK,A,
               ACTRC_DSN_NEW_MJOB,A)
//SRTCCNTL DD *          DFSORT STATEMENTS - SORT AND REFORMAT
* Sort on new subchain name and start date
  SORT FIELDS=(ACTRC_DSN_NEW_CHAINS,A)
//SRTECNTL DD *          DFSORT STATEMENTS - SORT AND REFORMAT
* Sort on volume name and status
  SORT FIELDS=(ACXSYC_EXPDTDROP_VOL,A,
               ACXSYC_PRE_TYPE,D)
//SRTFCNTL DD *         DFSORT STATEMENTS - SORT AND REFORMAT
* Sort on status
  SORT FIELDS=(ACXSYC_PRE_TYPE,D,
               ACXSYC_EXPDTDROP_VOL,A)
//SRT1CNTL DD *          DFSORT STATEMENTS - SORT AND REFORMAT
* Sort on VOLSER, FSEQ, RECORD TYPE (D,X)
  SORT FIELDS=(ACXSYC_VRSRETAIN_VOL,A,
               ACXSYC_VRSRETAIN_FSEQ,A,
               ACXSYC_VRSRETAIN_REC_TYPE2,A)
//SRT2CNTL DD *          DFSORT STATEMENTS - SORT AND REFORMAT
* Sort on VOLSER, RECORD TYPE (V,D) and VOLUME RETAINED (Y,N)
  SORT FIELDS=(ACXSYC_VRSRETAIN_VOL,A,
               ACXSYC_VRSRETAIN_REC_TYPE1,D,
               ACXSYC_VRSRETAIN_VOL_RET,D)
//SRT3CNTL DD *          DFSORT STATEMENTS - SORT AND REFORMAT
* Sort on Vol-Retained (Y/N), Volser and FSEQ
  SORT FIELDS=(ACXSYC_PRE_TYPE,D,
               ACXSYC_VRSRETAIN_VOL,A,
               ACXSYC_VRSRETAIN_FSEQ,A)
/*
//*
//RUNINFO  DD DSN=RMM.REPORT.RUNINFO,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(1,1),RLSE),UNIT=SYSALLDA
//VRS      DD DSN=RMM.REPORT.VRS,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//VRSS     DD DSN=RMM.REPORT.VRSS,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//RETDATE  DD DSN=RMM.REPORT.RETDATE,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//RETDS    DD DSN=RMM.REPORT.RETDS,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//MATCHVRS DD DSN=RMM.REPORT.MATCHVRS,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//MATCHVS  DD DSN=RMM.REPORT.MATCHVS,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//SUBCHN   DD DSN=RMM.REPORT.SUBCHN,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
```

```
//SUBCHNS  DD DSN=RMM.REPORT.SUBCHNS,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//EXPDROP  DD DSN=RMM.REPORT.EXPDROP,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//EXPDROPS DD DSN=RMM.REPORT.EXPDROPS,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//VRSRETN  DD DSN=RMM.REPORT.VRSRETN,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
//VRSRETNS DD DSN=RMM.REPORT.VRSRETNS,DISP=(,CATLG,DELETE),
//            SPACE=(CYL,(9,9),RLSE),UNIT=SYSALLDA
```

The new report files created for VRSRETAIN retention limit reporting are: VRSRETN and VRSRETNS.

Figure 9-3 shows the newly assigned volume subject VRS retention detail report.

```
1Newly assigned volumes subject to VRSRETAIN        09/30/10        13:40:25        - 1 -

 Status:  RETAINED

 D A T A S E T                                D A T A S E T   V R S                          V O L U M E
                                                        DROP REASON                          VRS             RETAIN   FILE  IN
 VOLSER  FSEQ  DSNAME                 JOBNAME  RETAINED  PRIM  2nd    PRIMARY VRS       JOB MASK TYPE VRS         REASON   COUNT SET
 ------- ----- ---------------------- -------- --------- ----- ------ ----------------- -------- ---- ----------- -------- ----- ---
 THS003      1 HSM.HMIGTAPE.DATASET   DFHSM63  Y                      HSM.**                     D                DATASET      1 N
 THS006      1 HSM.HMIGTAPE.DATASET   DFHSM70  Y                      HSM.**                     D                DATASET      1 N
 THS007      1 HSM.BACKTAPE.DATASET   DFHSM70  Y                      HSM.**                     D                DATASET      1 N
 THS014      1 HSM.BACKTAPE.DATASET   DFHSM70  Y                      HSM.**                     D                DATASET      1 N

 data sets in this status:              4
1Newly assigned volumes subject to VRSRETAIN        09/30/10        13:40:25        - 2 -

 Status:  NOTRETAINED

 D A T A S E T                                D A T A S E T   V R S                          V O L U M E
                                                        DROP REASON                          VRS             RETAIN   FILE  IN
 VOLSER  FSEQ  DSNAME                 JOBNAME  RETAINED  PRIM  2nd    PRIMARY VRS       JOB MASK TYPE VRS         REASON   COUNT SET
 ------- ----- ---------------------- -------- --------- ----- ------ ----------------- -------- ---- ----------- -------- ----- ---
 VT0001      1 HAIMO.VTFMTST1         VTFMTST1                                                                                  1 N
 VT0010      1 RMM.G1234567.B1234567. RES7GENR                                                                                  1 N

 data sets in this status:              2
```

Figure 9-3 VRS retention detail report

Note: The record layout is condensed to fit into the figure.

Figure 9-4 shows the newly assigned volume subject VRS retention summary report.

```
1Summary of newly assigned volumes for VRSRETAIN   09/30/10   13:40:25 - 1 -

 Status           VOLUME COUNT
 -----------   ---------------
 NOTRETAINED                 2
 RETAINED                    4
```

Figure 9-4 VRS retention summary report

The new report files created for EXPDTDROP retention limit reporting are: EXPDROP and EXPDROPS.

Figure 9-5 shows the newly assigned volume subject EXPDTDROP retention detail report.

```
1EXPDT retained volumes subject to EXPDTDROP        09/30/10        13:40:24        - 1 -

 Status:  RELEASED

 VOLSER VSEQ DSNAME         JOBNAME  EXPRSN ASSIGNED   EXPDT      SR RETDATE    ACTIONS LOCATION HOME     DEST RLS ACT HOLD
 ------ ---- -------------- -------- ------ ---------- ---------- -- ---------- ------- -------- -------- ---- ------- ----
 THS000    1 RMM.CLEANUP    GARBDSN1 X      2009/224   2009/365   N                      LIB2     LIB2          S       N
 VT0000    1 RMM.F00001     TEST9999 X      2009/224   2009/364   N  2009/241           VTFM001  VTFM001         0     N
 VT0002    1 RMM.F00001     TEST9999 X      2009/224   2009/365   N                      VTFM001  VTFM001       S       N
 VT0003    2 RMM.F13002.0.. TEST9999 X      2009/240   2009/240   N                      VTFM001  VTFM001       S       N

 Volumes in this status:          4
```

Figure 9-5 EXPDTDROP retention detail report

Note: The record layout is condensed to fit into the figure.

Figure 9-6 shows the newly assigned volume subject EXPDT retention summary report.

```
1Summary of EXPDT retained volumes for EXPDTDROP   09/30/10   13:40:25 - 1 -

 Status           VOLUME COUNT
 -----------   ---------------
 RELEASED                    4
```

Figure 9-6 EXPDTDROP retention summary report

9.3 Ignore for duplicate volumes

For specific volume requests, DFSMSrmm uses the requested volume to determine whether the volume is defined to DFSMSrmm, and uses the VOL1 label volser only to validate duplicate volumes. The requested volume is matched to the correct OPENRULE, and in this way DFSMSrmm correctly handles the identification and ignore action for duplicate volumes.

Figure 9-7 on page 128 compares OPENRULE processing when you are using DFSMSrmm z/OS V1.12 or later and the processing in the previous release.

OPENRULE IGNORE

- Ignore processing for specific volser request (read and write)
  - The external/vision volser is used to identify the mounted volume

Library	OPENRULE for		Result in	
	requested volser	VOL1 volser	V1R11	V1R12
Non-system managed (SHELF)	IGNORE	ACCEPT	requested volser ignored (at file validation)	requested volser ignored (at mount verification)
	ACCEPT	IGNORE	VOL1 volser ignored, LBL ERR VOL1	accepted
System managed (ATL)	IGNORE	ACCEPT	requested volser ignored (at file validation)	requested volser ignored (at mount verification)
	ACCEPT	IGNORE	VOL1 volser ignored, 613-1C requested volser	accepted

Figure 9-7 OPENRUL ignore processing

9.4 Automation for WTORs in production and parallel running

Use the parmlib member AUTORxx to activate auto-reply processing on a system. The member AUTOR00 contains the auto-reply policy suggested by IBM. You can modify the member (which is not recommended), or define another AUTORxx member to customize the auto-reply policy. The member AUTOR00 also contains comments that provide the message text for each WTOR and the rule used to select the WTOR.

9.4.1 Updating AUTORxx

The z/OS auto-reply (AUTOR) facility provides replies to write-to-operator with reply (WTOR) messages in cases where there is no automation or when the operator is unaware of the outstanding request or is taking too long trying to determine what the response should be. When AUTOR is active, z/OS will use the auto-reply policies provided for the subset of the DFSMSrmm WTORs that are included in the default AUTOR00 parmlib member provided by z/OS. AUTOR00 also contains comments that provide the message text for each WTOR and the rule used to select the WTOR. Update AUTORxx to authorize DFSMSrmm commands and utilities. If you use the TSO CSECT facility rather than IKJTSOxx, refer to *TSO/E Customization*, SA22-7783, for more information.

You can define your own AUTORxx member to customize the auto-reply policies for DFSMSrmm. You can override or supplement the policies in AUTOR00 to add more WTORs to be handled automatically and to change the replies for these WTORs already included in AUTOR00. DFSMSrmm provides these AUTORxx parmlib members, which you can use as-is or use as the base for your customization:

AUTORRM Includes, for all DFSMSrmm WTORs, an automated response suitable for production running.

AUTORRP Includes, for all DFSMSrmm WTORs, an automated response suitable for use when running DFSMSrmm in a mode other than OPMODE(PROTECT).

Parameter in IEASYSxx or supplied by the operator

Use the AUTOR=(xx,00) keyword in IEASYSxx to specify your selected parmlib member. If you are new to DFSMSrmm, you can use AUTOR=(RP,00) until you are ready for production. You can then use AUTOR=(RM,00) or your customized AUTORxx.

Syntax rules for AUTORxx

The following syntax rules apply to the AUTORxx parmlib member:

- Data is specified in columns 1-71.
- Comments can start in any column and can span lines, and must start with /* and end with */. Comments can appear in any place where a blank is accepted, except within quoted strings.
- Generally, syntax errors cause the auto-reply changes to be rejected, whereas in the following cases, syntax errors are ignored and allow the changes to become active:
  - The maximum number of message IDs is reached.
  - Duplicate message IDs are specified.
  - NOTIFYMSGS is specified in different members.

Syntax format of AUTORxx

Use the syntax shown in Figure 9-8 to add your DFSMSrmm related messages.

```
                 |—HC————————|
▶▶— NOTIFYMSGS(——————————————)————————————————————————————————◀◀
                 |           |
                 |——CONSOLE——|


                      |—NOAUTOREPLY————————————————————————————|
                      |                                        |
                      |            |—m—|            |———,————————|  |
                      |            |—s—|            ▼            |  |
                      | DELAY(—nnn———————)———————REPLY(——['']replytext[']——)—|
▶▶— MSGID('msgid')————————————————————————————————————————————————◀◀
```

Figure 9-8 AUTORxx command syntax

Note: The maximum number of message IDs that do not contain wildcards is limited to 10,413. The maximum number of message IDs that contain wildcards is limited to 1,500.

Updating Auto-reply Policy and WTORs Information

You can use the MVS SET AUTOR command shown in Figure 9-9 to change the current Auto-reply Policy information.

```
SET AUTOR=(RM,00)
or
SET AUTOR=00
```

Figure 9-9 Update AUTO-replay and WTORs information

The two alphanumeric characters indicate the AUTORxx members of the logical PARMLIB that contain the specified Auto-reply Policy information. AUTOR allows you to change the current Auto-reply Policy specified in one or more AUTORxx PARMLIB member. Do not use parentheses when only one PARMLIB member is specified.

After the new setting is in effect you should get the information shown in Figure 9-10.

```
SET AUTOR=(RM,00)
IEE252I MEMBER  AUTORRM FOUND IN SYS1.IBM.PARMLIB
IEE252I MEMBER  AUTOR00 FOUND IN SYS1.PARMLIB
CNZ2600I AUTO-REPLY POLICY MODIFIED.
```

Figure 9-10 Display AUTOR command output

Note: If you specify only one parmlib member with APPC=, ASCH=, AUTOR=, CEE=, CNGRP=, GRSRNL=, MPF=, OMVS=, PROG=, or SCH=, you do not need to enter the parentheses.

Displaying Auto-reply Policy and WTORs information

Use the DISPLAY AUTOR command shown in Figure 9-11 to display the current auto-reply policy.

```
DISPLAY AUTOR,POLICY
or
DISPLAY AUTOR,P
```

Figure 9-11 Display your current AUTOR settings

Example 9-2 shows the result of a DISPLAY AUTOR,POLICY command.

Example 9-2 DISPLAY AUTOR,POLICY command output

```
RESPONSE=SC70
 CNZ2603I 16.41.49 AUTOR POLICY 217
 POLICY ACTIVATED AT 16.28.35 ON 10/01/2010  NOTIFYMSGS(HC)
 FROM PARMLIB MEMBERS RM,00
 --MSG ID-- DELAY MEM ----REPLY TEXT----
...
 EDG0103D     1M   RM RETRY
 EDG0107A     1M   RM 00
 EDG0115D     1M   RM CANCEL
 EDG0117D     1M   RM CANCEL
 EDG0123D     5M   RM N
 EDG0127D     5M   RM CANCEL
 EDG0215D     2M   RM N
 EDG0228D     2M   RM CANCEL
 EDG0358D    10M   RM CANCEL
 EDG0361D    10M   RM RETRY
 EDG1107D     2M   RM RESTART
 EDG1200D    30S   RM CANCEL
 EDG1203D    30S   RM CANCEL
 EDG2103D    30S   RM L
 EDG2106D    30S   RM L
 EDG3213D    30S   RM RETRY
```

```
 EDG4000D     2M   RM RETRY
 EDG4001D     2M   RM RETRY
 EDG4010D     1M   RM RETRY
 EDG4012D     1M   RM RETRY
 EDG6605D    10M   RM RETRY
 EDG6607D    10M   RM RETRY
 EDG6609D    10M   RM RETRY
 EDG6626A    10M   RM END
 EDG6627A    10M   RM S
 EDG6629D     1M   RM CANCEL
 EDG6663D    10M   RM F
 EDG6671D    10M   RM F
 EDG8008D     1M   RM RETRY
 EDG8010D     1M   RM RETRY
 EDG8011D     1M   RM RETRY
 EDG8013D     1M   RM RETRY
 EDG8102D     1M   RM RETRY
 EDG8108D     1M   RM RETRY
 EDG8110D     1M   RM RETRY
 EDG8113D     1M   RM RETRY
 EDG8121D     1M   RM RETRY
 EDG8122D     1M   RM RETRY
 EDG8123D     1M   RM RETRY
....
```

The messages shown in this example are the default defined in the AUTORRM parmlib member and can by used if DFSMSrmm is running in protect mode.

The columns show the following detail:

Column 1 Message ID in the range of 1 to 10 characters.

Column 2 The amount of time, in minutes (M) or seconds (S), in the range 0 - 999, to wait, after the WTOR is issued and before auto-reply processing replies to the WTOR.

Column 3 The AUTORxx parmlib member suffixes that specified the policy.

Column 4 The reply that is used if auto-reply processing replies to the WTOR.

You can use the DISPLAY AUTOR command shown in Figure 9-12 to display the current WTORs being monitored by auto-reply processing.

```
DISPLAY AUTOR,WTOR
or
DISPLAY AUTOR,W
```

Figure 9-12 Display your current AUTOR processing

Example 9-3 shows the result of a DISPLAY AUTOR,POLICY command.

Example 9-3 DISPLAY AUTOR,WTOR command output with pending actions

```
D AUTOR,W
CNZ2604I 06.47.02 AUTOR WTORS
0009 STATUS=06.47.58 SYS=SY1
MSG=IEE800D CONFIRM VARY FORCE FOR 3D0 - REPLY NO OR YES
REPLY=NO
```

```
0008 STATUS=06.47.46 SYS=SY1
MSG=IXC371D CONFIRM REQUEST TO VARY SYSTEM SY1 OFFLINE. REPLY
SYSNAME=SY1 TO REMOVE SY1 OR C TO CANCEL.
REPLY=C
```

Example 9-4 shows the result of a DISPLAY AUTOR,WTOR command if there is no action pending.

Example 9-4 DISPLAY AUTOR,WTOR command output without pending actions

```
RESPONSE=SC70
 CNZ2604I 16.50.20 AUTOR WTORS 220
 NO PENDING AUTO-REPLIES TO WTORS
```

9.5 Expiration override for volumes

Before DFSMSrmm z/OS V1.12 there was a problem in preventing a volume from release. To solve this problem there is a new subcommand operand to prevent expiration. Existing policy and retention is unchanged.

9.5.1 Subcommand HOLD operand

Use the HOLD operand to set the volume HOLD attribute. Setting the HOLD attribute prevents automatic expiration and also prevents use of the RMM DELETEVOLUME subcommand with the RELEASE operand. The HOLD operand is only valid for non-scratch, non-pending release volumes. The RETAINBY(SET) option does not consider the HOLD attribute; each volume is subject to hold independently. HOLD and NOHOLD are mutually exclusive. Authorization requires either CONTROL access to RACF resource STGADMIN.EDG.MASTER, or UPDATE access to STGADMIN.EDG.CV.HOLD.*volser* as shown in Figure 9-13.

Define the resource	To Control the
STGADMIN.EDG.CV.HOLD.volser STGADMIN.EDG.CV.NOHOLD.volser	Use of the CHANGEVOLUME with HOLD/NOHOLD

When you define	With Access	Then
STGADMIN.EDG.CV.hold.volser Note: RACF profile must not contain generic characters prior to ‘hold.volser’	Entity not defined	CONTROL access to STGADMIN.EDG.MASTER is required
	NONE	No authority is granted to use CV HOLD/NOHOLD
	UPDATE	You are permitted to set and reset the volume HOLD attribute

Figure 9-13 Needed HOLD and NOHOLD authorization access

Set the HOLD attribute

Use the RMM TSO CHANGEVOLUME subcommand with the HOLD operand to prevent a volume from expiring. Figure 9-14 shows you volume VT0000 before the HOLD attribute is set.

```
 RMM LV VT0000
 Volume information:
 Volume = VT0000   VOL1  =          Rack  =           Owner   = MHLRES7
   Type = LOGICAL          Stacked count  = 0         Jobname = TEST9999
   Worldwide ID =                                     WORM    = N
 Creation:  Date = 2009/222    Time = 13:49:45  System ID = SC70
 Assign:    Date = 2009/224    Time = 23:34:50  System ID = SC70
 Expiration date = 2009/364   Original      =
 Retention date  = 2009/241   Set retained  = NO
 Data set name = RMM.F00001
 Volume status:             Hold = N
 Status = MASTER    Availability =                    Label = SL
...
 RMM SV VOLUME(VT*) OWNER(MHLRES7) LIMIT(*) NOHOLD
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
VT0000 MHLRES7         2009/224   2009/364   VTFM001  12999 M
VT0002 MHLRES7         2010/274   2010/274   VTFM001  13002 M
EDG3012I 2          ENTRIES LISTED
```

Figure 9-14 LISTVOLUME and SEARCHVOLUME before the HOLD attribute is set

Figure 9-15 shows the command we used to set the HOLD attribute for volume VT0000.

```
 RMM CV VT0000 HOLD
```

Figure 9-15 Setting the HOLD attribute for volume VT0000

Figure 9-16 on page 134 shows the details for volume VT0000 after the HOLD attribute is set.

```
 RMM LV VT0000
Volume information:
Volume = VT0000   VOL1  =          Rack  =           Owner   = MHLRES7
  Type = LOGICAL          Stacked count  = 0         Jobname = TEST9999
  Worldwide ID =                                     WORM    = N
Creation:  Date = 2009/222    Time = 13:49:45  System ID = SC70
Assign:    Date = 2009/224    Time = 23:34:50  System ID = SC70
Expiration date = 2009/364   Original      =
Retention date  = 2009/241   Set retained  = NO
Data set name = RMM.F00001
Volume status:             Hold = Y
Status = MASTER    Availability =                    Label = SL
...
 RMM SV VOLUME(VT*) OWNER(MHLRES7) LIMIT(*) HOLD
Volume Owner    Rack   Assigned   Expiration Location Dsets St Act   Dest.
                       date       date
------ -------- ------ ---------- ---------- -------- ----- -- ----- --------
VT0000 MHLRES7         2009/224   2009/364   VTFM001  12999 M
EDG3011I 1 ENTRY LISTED
```

Figure 9-16 LISTVOLUME and SEARCHVOLUME before the HOLD attribute is set

After the HOLD attribute is set for a volume the volume's expiration date can still be changed, but you cannot use the RMM DELETEVOLUME subcommand with the RELEASE operand as shown in Figure 9-17.

```
 RMM DV VT0000 RELEASE
EDG3361I HOLD ATTRIBUTE IS SET
```

Figure 9-17 Releasing a volume when the HOLD attribute is set

When EDG3361I is displayed the HOLD attribute is set, and the following is true:

- Explanation: If you issued the CHANGEVOLUME subcommand, you are attempting to set the hold attribute but the attribute is already set. If you issued the DELETEVOLUME subcommand, you are attempting to release a volume when the HOLD attribute is set.
- System action: The subcommand ends with return code 12 reason code 258.
- Application Programmer Response: Determine the correct setting and reissue the correct subcommand.
- Source: DFSMSrmm.
- Detecting module: EDGTSO.

Note: To reset the volume HOLD attribute you must use the TSO subcommand CHANGEVOLUME using the NOHOLD operand before you can releasing a volume have the HOLD attribute set.

Using the ISPF Dialog to set the HOLD/NOHOLD attribute

The HOLD or NOHOLD attribute can be set using the new **HY** and **HN** line commands in the DFSMSrmm ISPF SEARCHVOLUME result panel as shown in Figure 9-18.

```
                    DFSMSrmm Volumes (Page 1 of 2)                Volume changed
Command ===>                                                    Scroll ===> CSR

Enter HELP or PF1 for the list of available line commands
Use the RIGHT command to view other data columns
   Volume          Assigned   Expiration S                  Dest-    Tr- Data
S  serial Owner    date       date       R Status  Location ination  ans sets
-- ------ -------- ---------- ---------- - ------- -------- -------- --- -----
HY VT0000 MHLRES7  2009/224   2009/364     RELEASE VTFM001           N   12999
   VT0001          2010/274                SCRATCH VTFM001           N   1
   VT0002 MHLRES7  2010/274   2010/274     MASTER  VTFM001           N   13002
   VT0003          2010/274                SCRATCH VTFM001           N   1
   VT0004 STC      2009/252   1999/365     VRS     VTFM001           N   1
   VT0005 STC      2009/252   1999/365     VRS     VTFM001           N   1
   VT0006 STC      2009/252   1999/365     VRS     VTFM001           N   1
   VT0007 STC      2009/252   1999/365     VRS     VTFM001           N   1
```

Figure 9-18 ISPF DFSMSrmm SEARCHVOLUME result panel

Note: After the HOLD attribute for a volume is successfully changed you get the `Volume changed` message on the right corner at the top of the panel.

9.5.2 Report generator sample report EDGGAHLD

To list all volumes that have the HOLD attribute set you can use the EDGGAHLD report delivered as a sample in the report generator, as shown in Figure 9-19.

```
                         DFSMSrmm Report Definitions        Row 16 to 37 of 43
Command ===>                                                  Scroll ===> CSR

The following line commands are valid: A,D,G,H,J,L,M,N,S, and T

S Name     Report title                   Report type                User id
- -------- ------------------------------ --------------------------- -------
  ARCGS010 DFSMShsm Thrashing Report      DFSMShsm FSR-SMF Records    HSM
  EDGGAHLD Held Volumes by Volume Serial, Extended Extract Records    RMM
  EDGGAUD1 SMF Audit of Volumes by Volser SMF Records for Volumes     RMM
  EDGGAUD2 SMF Audit of Volume by Rack    SMF Records for Volumes     RMM
....
```

Figure 9-19 DFSMSrmm report generator reports

Figure 9-20 on page 136 displays a list of all data sets on volumes have the HOLD attribute set. If you want to list only the first data set on a volume you have to modify the sample EDGGAHLD report specifications.

```
1         Held Volumes by Volume Serial                 - 1 -        2010/10/03        19:22:19

 Extract file was created on 2010/276   at 190428

 Volume   Dataset name   Vol    DSN   Creating   DSN date   DSN time   Vol exp      Vol date   Vol date   Retentio   Volume   Location
                         seq    seq   jobname    created    created    date         read       write      date       Status
 ------   ------------   ----   ---   --------   --------   --------   ----------   --------   --------   --------   ------   --------
 VT0002   RMM.F00001       1     1    TEST9999   2009/224   233450     2010/274     2009/240   2009/240              MASTER   VTFM001
 VT0002   RMM.F00002       1     2    TEST9999   2009/224   233451     2010/274     2009/240   2009/240              MASTER   VTFM001
 VT0002   RMM.F00003       1     3    TEST9999   2009/224   233451     2010/274     2009/240   2009/240              MASTER   VTFM001
 VT0002   RMM.F00004       1     4    TEST9999   2009/224   233451     2010/274     2009/240   2009/240              MASTER   VTFM001

 Removable Media Manager
```

Figure 9-20 EDGGAHLD sample report

Note: The record layout was changed to fit in the available space.

9.6 ISPF Dialog CLIST option to avoid search results list

In all primary search panels there is a new selection field that can be used to create a data set of executable commands or to prepare an import list, for example. You can edit the data set to remove any volumes you do not want in the list. Then you can run the CLIST at your convenience.

```
EDGPT010                    DFSMSrmm Volume Search
Command ===>

                                                                  More:   - +
                            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
Hold  . . . . .             YES or NO
Clist . . . . . YES         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

Volume type . .             ( LOGICAL , PHYSICAL or STACKED )

Enter HELP or PF1 for the list of available line commands
```

Figure 9-21 DFSMSrmm primary SEARCHVOLUME panel

Note: You must scroll down to see the new Clist selection field.

If you choose YES in the panel in Figure 9-21, to create a data set, the pop-up panel shown in Figure 9-22 is displayed. If you choose View search results NO, which is the default, the

search result list is not displayed. In our example we created the RMM TSO CHANGEVOLUME subcommand to set the HOLD attribute for all matching volumes.

> **Note:** Add a blank after behind the RMM CV prefix and before the HOLD suffix of the command so that the subcommand is created correctly.

Use the Clist processing menu to specify the values used by the DFSMSrmm TSO subcommand CLIST operand. Clist processing creates a data set containing information about each resource that matches your search criteria. Typically this is a TSO command you would like to execute for each matching resource.

To control how the records in the Clist data set are created, you can specify a prefix string and a suffix string. You can also specify the name of a data set to be used instead of the default data set, userid.EXEC.RMM.CLIST. When you specify a data set name, you can give DFSMSrmm a hint about the possible size of the data set by specifying how many records you expect to be returned. When the search request is completed, you are provided with processing options including browse, edit, rename, and execute.

```
EDGP§CLS                  DFSMSrmm CLIST Processing
Command ===>

Enter optional prefix and suffix values

Prefix  . . . . . . . 'RMM CV '
Returned text depending on resource being searched
Suffix  . . . . . . . ' HOLD'

Enter optional fully qualified or partial data set information for CLIST

Data set name  . . . . 'MHLRES7.EXEC.RMM.HOLDX.CLIST'
Expected data set size 100         records

Extend existing CLIST  YES         YES, NO or blank

View search results    NO          YES, NO or blank



Press ENTER to CONTINUE, or END to RETURN.
```

Figure 9-22 DFSMSrmm CLIST option panel

Consider the following when selecting Clist processing options:

- **Prefix/Suffix**: Use the Prefix and Suffix fields to specify information to be included in the Clist data set along with the names of matching resources. DFSMSrmm returns the resource name for each resource. DFSMSrmm returns the resource name for each record if you do not specify a prefix or suffix value. You can add RMM TSO subcommands and operands to the records in the CLIST data set by specifying prefix and suffix strings. These text strings cannot exceed the panel width. Insert blanks in the prefix and suffix values to prevent DFSMSrmm from concatenating the strings with the data that DFSMSrmm returns. Put the prefix and suffix values into quotes, otherwise the leading and trailing blanks will not be used.

- **Data set**: The default data set name is the same as used by the RMM SEARCH subcommands. You only need to specify a data set name if you want to use a data set other than the default. A data set is allocated or created by dialog processing if you specify a data set name other than the SEARCH subcommand default.

  You can use any data set to receive the CLIST output; the dialog will allocate your data set using the RMMCLIST DD name. If you do not specify a data set name, DFSMSrmm allocates a data set called 'prefix.EXEC.RMM.CLIST'.

  Prefix can be:

  - A TSO user PROFILE PREFIX value, which is your TSO user ID if you do not change the PROFILE PREFIX.
  - The RACF user ID from the ACEE if you have specified PROFILE NOPREFIX and the RACF user ID is valid.

  When DFSMSrmm allocates this data set and determines that it is a partitioned data set, DFSMSrmm writes the output to the member TEMPNAME.

  The data set used for the CLIST output by default will have variable length records and a maximum logical record size of 255 unless you pre-allocate the data set and specify a different record format and LRECL. CLIST data set records can be either fixed or variable length and can optionally be blocked.

  When you use a variable record length the record length includes a 4-byte length field at the front of the record. DFSMSrmm honors the existing DCB attributes of the CLIST data set. When DFSMSrmm allocates a new CLIST data set or you do not specify DCB attributes, the defaults are variable length and an LRECL of 255. If the length of the output record exceeds the LRECL, DFSMSrmm splits the record into multiple records and adds a continuation character, +, to all but the last record.

  The LRECL must be at least long enough to contain the CLIST information for the record type you are searching. In addition, if a suffix or prefix is specified, the minimum LRECL is increased by 1 to allow for the continuation character. If the record is too short DFSMSrmm increases the record to the minimum size required to hold the output information. The maximum LRECL supported is 32760.

- **Expected size**: To help DFSMSrmm calculate the size of the CLIST output data set you can specify how many records you expect to be returned from your search request. When the dialog allocates the data set for you, the primary allocation amount is the value you specify and the secondary amount is half that value.

- **Extend**: To extend an existing CLIST data set, that is, to append new data to already collected data, specify YES. If you specify NO or leave the field blank, and the data set already exists, data will be replaced.

- **View results**: Specify YES to see the search results list. Otherwise specify NO or leave the field blank to skip the search results list. Skipping the search results list helps to reduce the use of system resources and the time taken to create the CLIST results. This is recommended when a large number of resources are found or when you do not need to see the results list.

If the search is finished the DFSMSrmm CLIST Menu panel is displayed as shown in Figure 9-23 on page 139.

```
                              DFSMSrmm CLIST Menu        Clist dataset created
Command ===>

  B - Browse data set                    D - Delete data set
  E - Edit data set                      K - Keep data set
  R - Rename data set                    S - Submit batch job
  X - Execute in foreground

Clist data have been saved in data set:

Data set name . . . : 'MHLRES7.EXEC.RMM.HOLDX.CLIST'

Specify new data set name, if "R" selected:

New data set name  . .

Job statement information:-
 ===> //RMMJOB   JOB ,RMM,NOTIFY=&SYSUID,
 ===> //         MSGCLASS=H,CLASS=A,MSGLEVEL=(1,1),REGION=6M
 ===> //*
 ===> //*
Press ENTER to process request, or END command to Exit,
```

Figure 9-23 DFSMSrmm CLIST menu panel

Select B to browse the data set or E to edit the data set created in this search. Figure 9-24 shows the RMM TSO CHANGEVOLUME subcommands we requested to create.

```
 BROWSE    MHLRES7.EXEC.RMM.HOLDX.CLIST              Line 00000000 Col 001 080
 Command ===>                                                  Scroll ===> PAGE
********************************* Top of Data ********************************
        RMM CV VT0000 HOLD
        RMM CV VT0001 HOLD
        RMM CV VT0002 HOLD
....
```

Figure 9-24 Executable RMM TSO subcommands created using the CLIST option

9.7 Display subsystem status information

The subsystem status is available via subcommands and API.

Now you can request information about DFSMSrmm subsystem address space status, tasks, and queued requests.

9.7.1 Subsystem status available via subcommands and API

In addition to the MODIFY operator command, you can now use the RMM LC STATUS subcommand to retrieve information about the DFSMSrmm subsystem, subsystem requests, and task status.

Use the MODIFY command shown in Figure 9-25 to produce a list of active tasks and other related status values.

```
F DFRMM,QUERY ACTIVE
```

Figure 9-25 Modify QUERY ACTIVE command

In this example `DFRMM` is the name of the DFSMSrmm started task.

Figure 9-26 shows the result of this command; as you can see there is only one function active at this time.

```
RESPONSE=SC70
 EDG1119I DFSMSrmm STATUS IS ACTIVE. JOURNAL ENABLED.
 EDG1120I Function  System  Task   Name  Started   Token  S IP Status
 EDG1113I HSKP              JOB=MHLRES7H 14:59:32 00701BE7      :  :
 EDG1114I LOCAL TASKS  10, ACTIVE   1, SERVER TASKS   0, ACTIVE   0
 EDG1122I                  HELD     0                    HELD     0
 EDG1118I      0 QUEUED REQUESTS, INCLUDING      0 NOWAIT      0 CATALOG
 EDG1121I DEBUG: DISABLED, PDA TRACE LEVEL: 1-2-3-4- RESERVE: 14:59:45
```

Figure 9-26 Modify QUERY ACTIVE result

9.7.2 Status information using the RMM LISTCONTROL subcommand

Use the RMM LISTCONTROL operand to request information about DFSMSrmm subsystem address space status, tasks, and queued requests. The information returned is similar to the results of the operator QUERY ACTIVE command.

This request is sent to the DFSMSrmm subsystem in the same way as all other subcommands: it is placed on a queue with other requests and processed in turn by the subsystem. The data returned reflects the status at the time the subsystem request is processed.

Use the RMM LISTCONTROL subcommand with the option STATUS as show in Figure 9-27 to get the DFSMSrmm subsystem address space status, tasks, and queued requests information.

```
RMM LISTCONTROL STATUS
```

Figure 9-27 Using the RMM LISTCONTROL STATUS subcommand

Figure 9-28 shows you the result of this command and as you can see there is only one function active at this time. The output of the LISTCONTROL STATUS command will always include at least one active task (LC) for itself.

```
DFSMSrmm status = ACTIVE    Journal = ENABLED   Server listener =
Local tasks     = 10        Server tasks  = 0
      Active    = 2                Active = 0
      Held      = 0                Held   = 0
Queued requests = 0         New requests  = NOTHELD
      Nowait    = 0
      Catalog   = 0
Last RESERVE    = 15:26:24 - DEQ
Debug Setting   = DISABLED
Trace Levels    =

Active requests:
Function  System  Task   Name  Started   Token   S IP Status
-------- -------- ------------ -------- -------- - ----------------
HSKP              JOB=MHLRES7H 15:26:16 00701BED
LC                TSU=MHLRES7  15:26:27 00900004
```

Figure 9-28 RMM LISTCONTROL STATUS subcommand result

9.7.3 Status information using the DFSMSrmm ISPF dialog

You can also use the DFSMSrmm ISPF dialog to request information about DFSMSrmm subsystem address space status, tasks, and queued requests. From the Primary DFSMSrmm Control Information Menu select Option 12 for Display status as shown in Figure 9-29.

```
                      DFSMSrmm Control Information Menu
Option ===> 12

0  OPTIONS      - Specify dialog options and defaults
1  CONTROL      - Display cds control record details
2  SYSTEM       - Display system options and defaults
3  SECURITY     - Display security classification rules
4  VLPOOLS      - Display volume pool definitions
5  MNTMSG       - Display mount message definitions
6  REJECT       - Display volumes to be rejected
7  ACTIONS      - Display volume moves and actions
8  LOCDEF       - Display location definitions
9  MEDINF       - Display media information definitions
10 PARTITION    - Display partition definitions
11 OPENRULE     - Display open rule definitions
12 STATUS       - Display status


Enter selected option or END command. For more info., enter HELP or PF1.
5694-A01  COPYRIGHT 1993,2010 IBM CORPORATION
```

Figure 9-29 DFSMSrmm Control Information Menu

Figure 9-30 shows the DFSMSrmm Status panel that will be displayed to show the requested information.

```
                              DFSMSrmm Status                 Row 1 to 2 of 2
Command ===>                                                 Scroll ===> CSR

DFSMSrmm status . : ACTIVE    Journal : ENABLED   Server listener :
                                                                  Task Commands
Local tasks . . . : 10        Active : 2       Held : 0         Active . .
Server Tasks  . . : 0         Active : 0       Held : 0         HSKP . . .
Queued requests . : 0         Nowait : 0       Catalog : 0      New  . . .
Debug . . . . . . : DISABLED  PDA trace level :
Last reserve  . . : 15:40:15  Outstanding : N                   New held : N

The following commands are valid: C,H and R or 'ENTER' to refresh

S Function System   Task Name     Started  Token    S IP Status
- -------- -------- ---- -------- -------- -------- - ----------------
  HSKP              JOB  MHLRES7H 15:40:08 00701BFA
  LC                TSU  MHLRES7  15:40:15 00900006
```

Figure 9-30 Display status result

Note: The output of the LISTCONTROL STATUS command will always include at least one active task (LC) for itself.

9.8 Copy Export reporting

EDGJCEXP provides a report on copies of logical volumes that have been exported from TS7700 Virtualization Engine. The report consolidates point-in-time information from the copy export status file, the library, and DFSMSrmm to help you identify tape data that has been copy exported.

Use the EDGJCEXP sample job to generate reports about copy exported data combining information from:

- TS7700 library
- Bulk Volume Information Retrieval Function (BVIR) or Export status file
- DFSMSrmm extended extract records

Restriction: BVIR code requirements

The BVIR function was introduced with the TS7700 Virtualization Engine. Although there are no library manager code changes to support the function, there are other functions of the TS7700 Virtualization Engine that require a compatible level of library manager code. To support the new BVIR function, the TS7700 code level required is 8.4.1.x or later with library manager code level 535.x or later. For the new recording formats of the TS1130 drive, the TS7700 code level required is 8.5.x.xx or later. There are no host software updates required for this function. For the new fields introduced with version 3 the TS7700 code level required is 8.6.x.xx or later. For the new fields introduced with version 4 the TS7700 code level required is 8.7.x.xx or later.

9.8.1 EDGJCEXP input and output

You can create the reports either from the export list file of up to three copy exports, or from the VOLUME MAP and PHYSICAL VOLUME STATUS POOL information created from the IBM Virtualization Engine TS7700 Series Bulk Volume Information Retrieval Function (BVIR). Information about stacked volumes and logical volume copies is retrieved from this input and merged with the information that the DFSMSrmm extract file X records has for the stacked volumes and the logical volumes.

A current report extract containing extended records (type X) is required. You can use any date format and time zone when you create the extract file.

Customize the EDGJCEXP sample JCL SET symbols to name the data sets to be used for input and output, and to select whether a copy export status file or BVIR output is used as input. The sample JCL contains an RMM Report Extract step at its beginning. The sample JCL is stored in the SAMPLIB. EDGJCEXP input and output that is defined next.

Input

The input SET symbols are:

CEXP	Set to 0 to use BVIR input, or to 1 to use copy export status file input.
BVIR	Set to 1 to use BVIR input, or to 0 to use copy export status file input.
EXTRACT	Set to the data set name to be used for the DFSMSrmm extract file. This data set must already exist.
MESSAGE	Set to the data set name to be used for the EDGHSKP MESSAGE DD. This data set must already exist.
BVOLMAP	If you set BVIR symbol to 1, set this symbol to the data set name of the BVIR Volume Map. If specified, this data set must already exist.
BVOLSTA	If you set BVIR symbol to 1, set this symbol to the data set name of the BVIR Volume Status. If specified, this data set must already exist.
COPEXP1	If you set CEXP symbol to 1, set to the data set name of the first Copy Export Status file. If specified, this data set must already exist.
COPEXP2	If you set CEXP symbol to 1, set to the data set name of the second Copy Export Status file or set to NULLFILE. If specified, this data set must already exist.
COPEXP3	If you set CEXP symbol to 1, set to the data set name of the third Copy Export Status file or set to NULLFILE. If specified, this data set must already exist.

Output

The output SET symbols are:

REPDSN	Set to the data set name to be used for the data set name report.
REPLVOL	Set to the data set name to be used for the logical volume report.
REPSVOL	Set to the data set name to be used for the stacked volume report.

9.8.2 Sample JCL

The sample JCL is stored in SYS1.SAMPLIB(EDGJCEXP) and must be tailored before it can be used.

9.8.3 Request a BVIR Report

To request and process requested data for the BVIR function you can use IEBGENER to write information to tape as shown in Figure 9-31.

Restriction: Do not use compaction when writing the request file.

```
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   DUMMY
//SYSUT2   DD   DSN=RMM.BVIR.CQUERY.VOLUME.MAP
//         UNIT=L3490J,LABEL=(,SL),
//         DISP=(NEW,CATLG),
//         DCB=(RECFM=F,BLKSIZE=80,LRECL=80,TRTCH=NOCOMP)
//SYSUT1   DD   *
VTS BULK VOLUME DATA REQUEST
VOLUME MAP
/*
```

Figure 9-31 Request a BVIR volume map report

In this example:

TRTCH=NOCOMP Specifies that the data set is not compressed.

RECFM=F Specifies that the data set is not blocked.

Attention: Using a logical scratch volume, create the request file with the two required records. To ensure that a scratch volume is allocated in the target library for the query operation, the ACS routines must have logic to allocate a tape drive in the target library. One way to accomplish this is to have a storage group unique to each VTS library providing a 1-to-1 relationship between storage group and library. The ACS routines would then need to key off of something unique in the DD statement (such as data set name, data class specification, or unit specification) to get the correct storage group and the right target library selected.

9.8.4 Copy the BVIR Report from Tape to DASD

To use the BVIR volume map report in a REXX exec you must copy it from tape to DASD using IEBGENER as shown in Figure 9-32.

```
//STEP1    EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   DUMMY
//SYSUT1   DD   DSN=RMM.BVIR.CQUERY.VOLUME.MAP,DISP=SHR
//SYSUT2   DD   DSN=RMM.BVIR.CQUERY.VOLUME.MAP-COPY,DISP=(,CATLG),
//         UNIT=SYSDA,SPACE=(TRK,(15,15),RLSE),
//         DCB=(LRECL=80,RECFM=FB,BLKSIZE=0)
```

Figure 9-32 Copying the BVIR volume map report

9.8.5 BVIR physical volume to logical volume mapping information

The TS7700 Virtualization Engine maintains the mapping between logical and physical volumes in a database on each cluster. It is possible that there are inconsistencies in the mapping information provided with this function. This results when a logical volume is being moved from one physical volume to another. For a period of time, the volume is shown on more than one physical volume. This can result in a small number of logical volumes reported as being on physical volumes which they were located on in the past, but are not presently located on.

Even with some inconsistencies, the mapping data is useful to customers that want to design jobs that recall data efficiently off of physical volumes. If the logical volumes reported on a physical volume are recalled together, the efficiency of the recalls will be increased. If a logical volume with an inconsistent mapping relationship is recalled, it will recall correctly, but an additional mount of a different physical volume might be required.

The physical volume to logical volume mapping associated with the physical volumes managed by the specific cluster the request volume is written to are returned in the response records. In a TS7700 Grid configuration, separate requests must be issued to each cluster to obtain the mapping for all physical volumes.

The response records are written in 80 byte fixed block format.

Note: The generation of the response might take several minutes to complete depending on the number of active logical volumes in the library and how busy the TS7700 cluster is at the time of the request.

Record 6

If this is a TS7720 configuration, the record shown in Table 9-1 is returned.

Table 9-1 Record 6 not supported

Bytes	Name	Description
1-57		'NOT SUPPORTED IN A DISK-ONLY TS7700 VIRTUALIZATION ENGINE'
58-80	Blanks	Blank character padding

This record provides a heading for the data records as shown in Figure 9-33.

```
----+----1----+----2----+----3----+----4----+----5----+----6----+----7----+----
PHYSICAL  LOGICAL P/B  ORDER   PART         SIZE
```

Figure 9-33 Record 6 heading

Table 9-2 provides a detailed description of this record type.

Table 9-2 Record 6 description

Bytes	**Name**	**Description**
1-8	PHYSICAL	Heading for the physical volume serial number column
9-10		Field Delimiter Blank characters
11-17	LOGICAL	Heading for the logical volume serial number column

Bytes	Name	Description
18		Field Delimiter Blank characters
19-21	P/B	Heading for the primary or backup pool indicator column
22-23		Field Delimiter Blank characters
24-28	ORDER	Heading for the logical volume order column
29-31		Field Delimiter Blank characters
32-35	PART	Heading for the logical volume spanning indicator column
36-44		Field Delimiter Blank characters
45-48	SIZE	Heading for the logical volume compresses size
49-80		Field Delimiter Blank characters

The primary and backup pool indicator column indicates whether the volume being reported resides on a primary pool volume or the secondary pool volume as defined using the Selective Dual Copy function that is part of the TS7700 Virtualization Engine's advanced policy management function.

The size field reports the number of MBs, rounded to two places after the decimal point, a logical volume occupies on the physical volume. This includes the effect of the compression performed on the data by the TS7700 Virtualization Engine, but does not include any effect of the compression performed by the physical drive.

Note: Only existing data created on prior B10/B18/B20 VTSs and migrated under the control of a TS7700 Virtualization Engine will indicate that it spans from one physical volume to another. Volumes created or recopied by the TS7700 Virtualization Engine do not span physical volumes.

9.8.6 BVIR physical volume status requests

A database is maintained on each individual TS7740 Cluster that contains information related to the management of the physical volumes on the cluster.

The physical volume status information returned represents the status of the volumes on the cluster the request volume is written on. In a TS7700 Grid configuration, separate requests must be issued to each cluster to obtain the physical volume status information for the individual clusters. A response record is written for each physical volume, selected based on the volume serial number mask or pool number specified in the request, that exists in the cluster.

A response record consists of the database fields defined in the following table. Fields are presented in the order defined in the table and are comma separated. The overall length of each record is 400 bytes, with blank padding after the last field as needed.

Figure 9-34 shows you the first few fields of the record returned for volser A03599.

```
A03599,2,FULL,READ-WRITE,2007-05-05-06.40.08.030061,2007-05-04-13.45.15.918473,..
```

Figure 9-34 Physical volume status example

Note: The generation of the response might take several minutes to complete depending on the number of volumes requested and how busy the TS7700 cluster is at the time of the request.

Record 6-N

If this is a TS7720 configuration, the record described in Table 9-3 is returned.

Table 9-3 Record 6-N not supported

Bytes	Name	Description
1-57		'NOT SUPPORTED IN A DISK-ONLY TS7700 VIRTUALIZATION ENGINE'
58-80	Blanks	Blank character padding

If the request is for a specific physical volume and it does not exist in the cluster's database or the request is for a range of physical volumes and none of the volumes exist in the cluster's database, or the request is for a pool and there are no physical volumes currently resident in the pool, the record shown in Table 9-4 is returned.

Table 9-4 Record 6-N no volume record exist

Bytes	Name	Description
1-53		'NO VOLUME RECORD(S) EXIST FOR REQUESTED VOLUME(S)'
54-80	Blanks	Blank character padding

For the requested volumes that do exist in the cluster's database, each of these records provides information for each physical volume as shown in Table 9-5.

Table 9-5 Record 6-N record description

Field Name	Description
VOLSER	6 character volume serial number
CURRENT_POOL	This field indicates the pool that the volume is currently assigned to. Pool 0 is the common scratch pool. Pools 1-32 are the specific data pools.
VOLUME_STATUS	This field indicates the volume's current capacity state. The following are the values that can be indicated: 'EMPTY' The volume contains no data and is available for use as a physical scratch volume. 'FILLING' The volume contains valid data, but is not yet full. It is available for additional data to be added to it. 'FULL' The volume contains valid data. As some point it was marked as full and additional data cannot be added to it. A volume can be marked full in some cases short of the volume capacity limit. 'UNKNOWN' The volume's capacity state is unknown.

Field Name	Description
VOLUME_ACCESS	This field indicates the volume's current accessibility state. The following are the values that can be indicated: 'READ-WRITE' The volume can be read from or written to. 'READ-ONLY' The volume contains valid data, but writes to it are not allowed. This might be because of an error that has been detected during a prior write operation to the volume, or a management operation, such as reclaim, is going to be performed against the volume. 'UNAVAILABLE' The volume is unavailable to the TS7740. 'DAMAGED' The volume has been damaged physically or logically such that it cannot be mounted and or read from. 'COPY_EXPORTED' The volume has been exported using the copy export function and is not currently resident in the library. 'CE_RECLAIM' The volume had a state of COPY_EXPORTED and a host console request was received requesting that the volume be reclaimed. It will remain in the CE_RECLAIM state during the reclaim process.
STARTED_EXPIRING_TIMESTAMP	Timestamp of when the data first started expiring on the physical volume. The format of the timestamp is: Year-Month-Day-Hour.Minute.Second.Microsecond for example: 2006-05-23-19.34.23.876129 This field is set to 1970-01-01-00.00.00.000000 when the volume becomes empty (returned to scratch) or is inserted into the library.
BECAME_EMPTY _TIMESTAMP	Timestamp of when all of the data on the volume last become invalid and the volume is empty. The format of the timestamp is: Year-Month-Day-Hour.Minute.Second.Microsecond for example: 2006-05-23-19.34.23.876129 When the volume is inserted into the library, this field is set to 1970-01-01- 00.00.00.000000.
BECAME_FULL _TIMESTAMP	Timestamp of when the volume is last marked as full. The format of the timestamp is: Year-Month-Day-Hour.Minute.Second.Microsecond for example: 2006-05-23-19.34.23.876129 When the volume is inserted into the library, this field is set to 1970-01-01-00.00.00.000000.
RECORDING_FORMAT	This field indicates the format that was used in recording the data on the volume. The following are the values that can be reported: '4' The volume is written in the 3592 J1A format. '5' The volume is written in the TS1120 E05 format. '6' The volume is written in the TS1120 E05E format (encrypted). '7' The volume is written in the TS1130 E06 format. '8' The volume is written in the TS1130 E06E format (encrypted). Values not defined are reserved.
PCT_UTILIZED	This field indicates the percent of the volume that contains active data. The percentage is determined using the active_size and the total_bytes_written fields. The number is reported in 1/10th of a percent, rounded up. This field is updated hourly (if needed). It is reset to 0 when the volume becomes empty (scratch).
TOTAL_BYTES_WRITTEN	This field indicates the number of bytes that were written to the volume when it was filled. It is reset to 0 when the volume becomes empty (scratch).
ACTIVE_SIZE	This field indicates the number of bytes of the active data on the volume. It is reset to 0 when the volume becomes empty (scratch).
ERROR_TYPE_FLAG	This field indicates whether the TS7700 has detected an error with the use of the volume. The error values are used by IBM service.

Field Name	Description
SCRATCH_COUNT	This field indicates the number of times the volume has been returned to scratch status since it was inserted into the library.
MOUNT_COUNT	This field indicates the number of times the volume has been mounted since it was inserted into the library.
MEDIA_TYPE	This field indicates the media type for the volume. The following are the values that can be reported: '5' IBM JA '6' IBM JJ '7' IBM JB Values not defined are reserved.
BATE_ROR_REASON	If the volume has a volume access value of 'Read-Only', this field provides additional information about the reason for the access value. A volume might be marked as read-only for reasons other than a media error. These values might be used for IBM Service.
DVM_ROR_REASON	If the volume has a volume access value of 'Read-Only', this field provides additional information about the reason for the access value. A volume might be marked as read-only for reasons other than a media error. These values might be used for IBM Service.
ERASE_FLAG	This field indicates whether or not the volume needs to be secure data erased.
MOUNT_FAILURE_COUNT	This field indicates the number of time a mount failure occurred with this volume since it was inserted into the library.
LAST_DEVICE_MOUNTED	This field contains the device number the volume was last mounted on when an error occurred. This field will be null if there has been no error mounting this volume.
LAST_WRITE_TIMESTAMP	Timestamp of when the volume was last written to. The format of the timestamp is: Year-Month-Day-Hour.Minute.Second.Microsecond for example: 2006-05-23-19.34.23.876129 When the volume is inserted into the library, this field is set to 1970-01-01- 00.00.00.000000.
LAST_READ_TIMESTAMP	Timestamp of when the volume was last read from. The format of the timestamp is: Year-Month-Day-Hour.Minute.Second.Microsecond for example: 2006-05-23-19.34.23.876129 When the volume is inserted into the library, this field is set to 1970-01-01-00.00.00.000000.
KEKLABEL1	If the volume has been encrypted, this field contains the first key label. A key label can be up to 64 characters in length. If the volume is not encrypted, this field is null. It is set to null when the volume becomes empty (scratch).
KEKLABEL2	If the volume has been encrypted, this field contains the second key label. A key label can be up to 64 characters in length. If the volume is not encrypted, this field is null. It is set to null when the volume becomes empty (scratch).
INSERT_TIMESTAMP	Timestamp of when the volume was inserted into the library. The format of the timestamp is: Year-Month-Day-Hour.Minute.Second.Microsecond for example: 2006-05-23-19.34.23.876129
ADSM_FORMAT_FLAG	This field indicates whether the physical volume is written using the pre-TS7700 (B10/B20 VTS) or TS7700 format. The following are the values that can be indicated: '0' The volume is written in the TS7700 format. '1' The volume is written in the pre-TS7700 format.
Blanks	As needed to pad record to 400 bytes

Figure 9-35 is an example of Physical Volume Status Volume output.

```
VTS BULK VOLUME DATA REQUEST
PHYSICAL VOLUME STATUS VOLUME ______
07/16/2008 17:18:49 VERSION 02
S/N: 00B0A LIB ID: 00000
A00063,2,FILLING,READ-WRITE,2008-07-01-23.59.49.000000,2008-07-01-10.12.12.000000,
1970-01-01-00.00.00.000000,5,828,341140026365,282600846878,0,0,30,5,0,0,N,0,NULL,
2008-07-14-05.24.03.000000,2008-07-14-05.23.59.000000,NULL,NULL,2008-06-24-18.47.44.256749,0
A00064,0,EMPTY,READ-WRITE,1970-01-01-00.00.00.000000,2008-07-16-00.44.12.000000,
2008-07-11-16.56.39.000000,6,0,0,0,0,2,13,5,0,0,N,0,NULL,2008-07-11-16.57.29.000
```

Figure 9-35 Physical Volume Status Volume output

Figure 9-36 is an example of Physical Volume Status Pool output.

```
VTS BULK VOLUME DATA REQUEST
PHYSICAL VOLUME STATUS POOL 12
10/30/2009 01:26:27 VERSION 02
S/N: 3484G  LIB ID: 3484A

A02075,12,EMPTY,COPY_EXPORTED,1970-01-01-00.00.00.000000,2009-10-29-16.50.05.000
A02076,12,EMPTY,COPY_EXPORTED,1970-01-01-00.00.00.000000,2009-10-29-16.50.05.000
A02077,12,FULL,COPY_EXPORTED,1970-01-01-00.00.00.000000,2009-10-28-16.50.06.0000
A02078,12,FILLING,READ-WRITE,1970-01-01-00.00.00.000000,2009-10-28-16.50.06.0000
A02079,12,EMPTY,READ-WRITE,1970-01-01-00.00.00.000000,2009-10-29-16.50.05.000000
A02080,12,EMPTY,READ-WRITE,1970-01-01-00.00.00.000000,2009-10-29-16.50.05.000000
A02081,12,EMPTY,READ-WRITE,1970-01-01-00.00.00.000000,2009-10-29-16.50.05.000000
```

Figure 9-36 Physical Volume Status Pool output

9.9 STOP/CANCEL improved

DFSMSrmm shutdown now issues an additional message to list the job names of the address spaces preventing shutdown. DFSMSrmm subsystem interface processing now correctly detects that DFRMM is or has been stopped.

Figure 9-38 shows the messages you get if there are some jobs preventing DFSMSrmm from shutting down.

```
EDG0154I SHUTDOWN OF DFSMSrmm DELAYED BY ANOTHER ADDRESS SPACE
EDG0155I ADDRESS SPACE LIST BY JOBNAME:
         MWWOOD   MHLRES7  SCHLUM   SIEGEL   MARYL
         NUMBER OF JOBNAMES DELAYING SHUTDOWN =    5
```

Figure 9-37 Shutdown delay messages

9.10 CA Reclaim considerations

If you have specified the CA RECLAIM in the PARMLIB and activated it as described in 4.2.2, “CA reclaim” on page 39, you should reorganize your DFSMSrmm a final time. To check your DFSMSrmm control data set run an IDCAMS EXAMINE as shown in Example 9-5.

Example 9-5 Sample JCL to run IDCAMS EXAMINE

```
//MHLRES7R JOB (999,POK),MSGLEVEL=1,NOTIFY=&SYSUID
/*JOBPARM S=SC70
//STEP1  EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN    DD    *
  EXAMINE  -
       NAME(RMM.CONTROL.DSET) -
       INDEXTEST         -
       DATATEST
/*
```

Figure 9-38 shows the output of the EXAMINE run.

```
IDC01700I INDEXTEST BEGINS
IDC11773I          10512 KEYS PROCESSED ON INDEX LEVEL   1, AVERAGE KEY LENGTH:    21.6
IDC11773I            522 KEYS PROCESSED ON INDEX LEVEL   2, AVERAGE KEY LENGTH:    19.3
IDC11773I             10 KEYS PROCESSED ON INDEX LEVEL   3, AVERAGE KEY LENGTH:    26.8
IDC11774I CURRENT INDEX CISIZE IS  2048, RECOMMENDED MINIMUM INDEX CISIZE IS  1024
IDC01724I INDEXTEST COMPLETE - NO ERRORS DETECTED
IDC01701I DATATEST BEGINS
IDC01728I FOUND 25 EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED
IDC01709I DATATEST COMPLETE - NO ERRORS DETECTED
IDC01708I 10512 CONTROL INTERVALS ENCOUNTERED
IDC01710I DATA COMPONENT CONTAINS 331221 RECORDS
IDC01711I DATA COMPONENT CONTAINS 25 DELETED CONTROL INTERVALS
IDC01712I MAXIMUM LENGTH DATA RECORD CONTAINS 1036 BYTES
IDC01722I 63 PERCENT FREE SPACE
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0

IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
```

Figure 9-38 IDCAMS EXAMINE output

The details regarding `IDC01728I FOUND` *nnnnnn* `EMPTY CONTROL AREAS THAT HAVE NOT BEEN RECLAIMED` are as follows:

- **Explanation:** The VSAM KSDS contains unused control areas (CAs). Large storage fragmentation might result in poor performance and “out of space” conditions. If the data set was defined with CA reclaim disabled or the function not available, and the data set is now CA reclaim-enabled, the number of unreclaimed CAs is the total of irreclaimable empty CAs before the enablement and a few that are left unreclaimed for data integrity after CA reclaim was enabled. These empty CAs might be reclaimed by subsequent VSAM ERASE requests on the reinserted records belonging to these CAs. Note that CA reclaim is not supported and is not performed for KSDSs defined with IMBED.

  In the message text, *nnnnnn* is the number of control areas found to be empty.
- **System action:** There is no error. Processing continues.
- **Application programmer response:** No action is required. Options to eliminate fragmentation can involve reinserting the ERASEd records or reorganizing the KSDS. To avoid further fragmentation by applications that delete and insert records of different keys, activate CA reclaim. CA reclaim will reclaim empty CAs created after CA reclaim has been activated, but will not reclaim preexisting ones. For data integrity, CA reclaim might temporarily leave some of the newly empty CAs unreclaimed; they can be reclaimed by later ERASE requests or reused by reinserting records belonging to the empty CAs. The CA with RBA 0 and the CA with the highest key of the KSDS are never reclaimed.

- **Source:** DFSMSdfp
- **Detecting Module:** IDCXM03, IDCXM05

Note: In this case a final reorganization need not be done immediately, but could be planned later.

9.11 PDA trace extended outside RMM address space

With DFSMSrmm on z/OS V1.12 and later releases the ASID/JOB value is automatically added to all trace records created outside of the DFRMM subsystem address space; this enables the ASID and jobname to be stored to the trace record.

Figure 9-39 shows the new record layout when the FORMAT option is used to print a trace entry.

```
TIME   USECS   ID AS/TCB         MOD   LOGIC CALLER  ARCPRPDO LEVEL=        -OW1
96319-----------------------------------------------------------------------------
121050.389472  01 D18E15         OCEOV ENTR   OCEXT
   +R13ADDR=      +0  06447000                          |................|
   ASID/JOB=      +0  0240C4C6 D9D4D4F1 C1              |..DFRMM1........|
   DATA=          +0  D3D6C3D2 60                       |LOCK-...........|
```

Figure 9-39 Formatted PDA trace output

The record entries are as follows:

- **AS/TCB**: Indicates it is a trace record from:

  D1 - a JOB

  E3 - a TSU address space

  E2 - a STC

  x'00' - for RMM Subsystem address space
- **ASID/JOB**: Shows the 2 byte ASID and the 8 character job name.

9.12 IPV6 supports networking strategy

The information in this section is important if you are planning to implement DFSMSrmm client and server systems or you have implemented it.

DFSMSrmm on z/OS V1.12 and later releases is an IPv6-enabled application that supports both IPv4 and IPv6 sockets. You can continue to use IPv4 on all systems or you can run a mixed environment with one or more V1.12 systems using IPv6 and other systems using IPv4. After all systems are V1R12, you have the option of moving all systems to IPv6. In a mixed environment, dual-mode IP stacks are required. You can use IP addresses that are compliant to either IPv4 or IPv6.

Note: DFSMSrmm client/server processing is dependent on Internet Protocol V4.

Figure 9-40 shows an example of an IPv6 address.

```
1080:0:0:0:8:800:200C:417A
```

Figure 9-40 IPv6 address

9.13 EAV supported for all data sets

All data sets used by DFSMSrmm can be eligible for allocation in the extended addressing space (EAS) of an extended address volume (EAV). This includes the DFSMSrmm journal and any dynamically allocated temporary files.

When not shared with a z/OS release before z/OS V1.12, the DFSMSrmm journal can be placed in an extended addressing space on an EAV.

Restriction: The 'prefix.EXEC.RMM.CLIST' data set when created automatically by SEARCH subcommand processing cannot be an extended addressing space (EAS) data set.

9.13.1 EAS exploited for temporary data sets

Temporary sort files created by DFSMSrmm for use by DFSORT are always large format EAS-eligible data sets when inventory management or EDGUTIL is used. EAS-eligible data sets can be placed in an EAS on an EAV. DFSMSrmm specifies EATTR=OPT on the dynamic allocation that creates the temporary data set.

All DFSMSrmm utilities and programs support NON_VSAM_XTIOT=YES options in the DEVSUPxx parmlib member. Any of the data sets used by or created by DFSMSrmm processing can be in EAS.

9.13.2 XTIOT, DSAB above, and uncaptured UCB support

All DFSMSrmm utilities and programs support NON_VSAM_XTIOT=YES options in the DEVSUPxx parmlib member. Any of the data sets used by or created by DFSMSrmm processing can be in EAS.

10

OAM enhancements in DFSMS V1.12

The Object Access Method (OAM) uses the concepts of system-managed storage to manage, maintain, and verify tape volumes and tape libraries within a tape storage environment. This chapter describes the OAM enhancements introduced in DFSMS V1.12.

10.1 Storage Group enablement in a non-OAMplex environment

OAM removes the restriction that a Storage Group can only be enabled to a single system in a non-OAMplex environment. OAM object processing now allows object and object backup storage groups enabled to more than one system in a non-OAMplex environment to be defined within OAM when the new MULTISYSENABLE keyword is specified in the CBROAMxx PARMLIB SETOPT statement.

To enable Storage Groups to be defined to OAM for more than one system in a non-OAMplex environment, in the CBROAMxx PARMLIB member, add the MULTISYSENABLE(YES) keyword to the SETOPT statement (Example 10-1). This keyword specifies that when a storage group has been encountered in a non-OAMplex environment that is enabled to more than one system, the storage group will be defined.

Example 10-1 MULTISYSENABLE keyword in CBROAMxx parmlib

```
SETOPT  MULTISYSENABLE(YES)
```

The SETOPT MULTISYSENABLE specification cannot be modified using the MODIFY OAM,UPDATE command. The SETOPT MULTISYENABLE keyword can only be specified at the global level, and does not apply to specific storage groups.

The SETOPT MULTISYSENABLE specification cannot be modified using the MODIFY OAM,UPDATE operator command, the MODIFY OAM,RESTART operator command, or an OAM restart resulting from an SCDS activation.

10.1.1 Displaying settings of the MULTISYSENABLE keyword

You can use the MODIFY OAM,DISPLAY,SETOPT,GLOBAL operator command to display the values of the new MULTISYSENABLE keyword. Example 10-2 is an example of the display message.

Example 10-2 MODIFY OAM command to display Global parameters

```
MODIFY OAM,DISPLAY,SETOPT,GLOBAL
CBR1075I GLOBAL value for OPREINIT is GROUP
CBR1075I GLOBAL value for OPDISDLY is 0
CBR1075I GLOBAL value for MWT is 5
CBR1075I GLOBAL value for UNLOADD is 1
CBR1075I GLOBAL value for UNLOADT is 9999
CBR1075I GLOBAL value for ABUNREAD is INACTIVE
CBR1075I GLOBAL value for ABOFFLIN is INACTIVE
CBR1075I GLOBAL value for ABNOTOPE is INACTIVE
CBR1075I GLOBAL value for ABDB2ERR is INACTIVE
CBR1075I GLOBAL value for ABLOST is INACTIVE
CBR1075I GLOBAL value for MULTISYS is Y
```

10.2 Message to show Immediate Backup and Recall to DB2

The output resulting from the MODIFY OAM,D,OSMC command is updated to include Immediate Backup and Recall to DB2 queue statistics to more accurately characterize the workload in OSMC. As shown in Example 10-3, a new message, CBR9364I, shows the name of the task, type of task, the number of Immediate Backup tasks that are currently active and the number of Immediate Backup tasks that are currently queued to process, followed by the number of Recall to DB2 tasks that are currently active and the number of Recall to DB2 tasks that are currently queued to process.

Example 10-3 CBR9364I message

```
CBR9364I OSMC Summary Status 2: 279
           TASK    TASKS    TASKS
ACTIVITY   TYPE    ACTIVE   QUEUED
IMBKUP     I            1        0
RCLDB2     B            0        0
End of Display Summary
```

10.3 Additional reminders when DB2 is stopped

OAM issues additional reminders in the event that DB2 is stopped, instead of a single, one-time message. OAM now issues CBR7530E whenever an F OAM command has been issued, to indicate that DB2 is not available and that OAM requires DB2 to be started for OAM to resume processing. Previously this message appeared once and then could be rolled off. This release will reissue CBR7530E after every subsequent OAM operator command entered until DB2 is brought back up or OAM is cancelled (Example 10-4).

Example 10-4 CBR7530E when DB2 stopped

```
CBR7530E OAM degraded. DB2 is not available. Start DB2.
```

10.4 Improved performance for volume recovery

The Volume Recovery utility is modified to improve the performance of the portion of the utility that extracts information from DB2 tables to determine which objects need to be recovered.

The amount of reduction varies greatly based on the following factors:

- The type of volume being recovered. The improvement for recovery of a backup volume will in most cases be much greater than for recovery of a primary volume.
- The number of volumes from which the objects to be recovered must be copied. In general, the higher the number of volumes from which objects need to be copied, the greater the percentage improvement will be.
- The number of collections containing the objects to be recovered on each volume from which the objects will be copied. The larger the number of unique "copy from" volume/collection combinations there are, the greater the percentage improvement will be.
- The total number of volumes, collections, and objects defined for each storage group containing objects that must be recovered. The greater the number of volumes, collections, and objects there are in a storage group, the greater the improvement will be.

- Whether the volume being recovered is tape or optical. The percentage improvement will be slightly greater for tape volumes than for optical volumes.

10.4.1 Changed message

The volume recovery CBR9863I message has been changed to include the total number of objects that were not recovered, as shown in Example 10-5.

Example 10-5 CBR9863I message

```
CBR9863I - Volume Recovery Utility status for volumes <volser> and <volser2>.
Total: <total>, Attempted:<attempted>, Successful: <successful>, Unsuccessful:
<unsuccessful>, Remaining: <remaining>.
```

10.5 Reduced CPU time for OSR

The OAM CICS Threadsafe feature enables the OSR application programming interface of OAM to function in a CICS threadsafe environment. Converting OSR to run in the CICS threadsafe environment reduces the CPU time used by OSR and application code that uses OSR.

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 159. Note that some of the documents referenced here might be available in softcopy only.

- *DFSMShsm Fast Replication,* SG24-7069

Other publications

These publications are also relevant as further information sources:

- *z/OS DFSMS Using Data Sets,* SC26-7410
- *DFSMS Access Method Services for Catalogs,* SC26-73942
- *z/OS MVS System Management Facilities (SMF),* SA22-7630
- *MVS System Command Manual,* SA22-7627
- *DFSMSdfp Diagnosis*, GY27-7618
- *IBM Health Checker for z/OS V1R12.0 User's Guide,* SA22-7994
- *TSO/E Customization*, SA22-7783
- *z/OS MVS Initialization and Tuning Reference,* SA22-7592

Online resources

These Web sites are also relevant as further information sources:

- z/OS V1.12 DFSMS online publications

  `http://www-03.ibm.com/systems/z/os/zos/bkserv/r12pdf/`

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 Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

E

F

G

H

IBM Redbooks

z/OS V1.12 DFSMS Technical Update

(0.2”spine)
0.17”<->0.473”
90<->249 pages

z/OS V1.12 DFSMS Technical Update

Introduces features and functions of z/OS V1.12 DFSMS

Includes implementation hints and tips

Provides working code and JCL 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 V1R12 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 enabling PTFs that have been integrated into z/OS V1R12 DFSMS.

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

SG24-7895-00 ISBN 0738435937