Redbooks

ibm.com/redbooks

IBM z/OS V2R2: Storage Management and Utilities

Keith Winnard

Natalia Barros

Jose Gilberto Biondo Jr

Jaqueline Mourão

Alvaro Salla

Ewerton Waki

z Systems

IBM

Redbooks

International Technical Support Organization

IBM z/OS V2R2: Storage Management

December 2015

SG24-8289-00

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

First Edition (December 2015)

This edition applies to Version 2, Release 2 of z/OS 2.2 (product number 5650 ZOS).

Contents

Notices vii
Trademarks viii

IBM Redbooks promotions ix

Preface xi
Authors xi
Now you can become a published author, too! xiii
Comments welcome xiii
Stay connected to IBM Redbooks xiv

Chapter 1. DFSMSdfp 1
1.1 DFSMSdfp overview 2
1.2 DFSMSdfp enhancements 2
1.3 SMS space constraint relief enhancements 3
1.3.1 Space constraint relief for guaranteed space allocations 3
1.3.2 Space constraint relief for secondary allocation 4
1.3.3 New IGD17296I message 4
1.4 User-defined ACS read-only variable 5
1.5 SMS RAS enhancements 6
1.5.1 New information console message 6
1.5.2 Enhance SELECT/DESELECT command 6
1.5.3 Including data set name and storage name in ACS messages 7
1.5.4 Display Storage Groups in IGD17800I Message 7
1.5.5 Externalizing message IGD17364I to hardcopy log 7
1.5.6 Storage Group Space alert messages 8
1.6 SMF VTOC audit log 10
1.6.1 **Changes in z/OS V2R2** **10**
1.7 Open/Close/End of Volume 12
1.7.1 Tape performance and recovery 12
1.7.2 Converting tape installation exits 12
1.7.3 DEVSUPxx enhancements 14
1.7.4 Adding JOBID and SYSPLEXID SMF 14 and 15 enhancements 15
1.7.5 Including above the line CSECTs in OCE in-core module table 15
1.7.6 Eliminating A13-18 while ensuring correct mount 15

Chapter 2. DFSMShsm 17
2.1 DFSMShsm overview 18
2.1.1 **Space management** **18**
2.1.2 **Availability management** **18**
2.1.3 DFSMShsm enhancements 18
2.2 Classic Migration, Storage Tiers, and Data Migration 19
2.2.1 Classic Migration 19
2.2.2 Storage tiers 19
2.2.3 Data migration 20
2.3 ARCMDEXT support for transition 21
2.4 Fast Replication Patch non-DFSMShsm backup utility 22
2.5 Fast replication messaging in a consolidated log 22
2.6 DFSMShsm common dump queue 25

2.6.1 Sample implementation 27
2.6.2 **Disconnecting from a Common Dump Queue 28**
2.6.3 **Changing the MS. 29**
2.6.4 **Shutdown considerations 30**
2.6.5 **MS loss of connectivity considerations 30**
2.6.6 **Holding, releasing, and canceling Dump considerations 30**
2.6.7 **Altering Dump priority 31**
2.6.8 Commands 31
2.7 Dump class definitions MINSTACK and MAXSTACK 33
2.8 Multiple copy pools onto the same tape 34
2.9 New UPDTCDS command 34

Chapter 3. DFSMSrmm 37
3.1 DFMSMSrmm overview 38
3.2 WHILECATALOG attribute and EXPDT retention method 38
3.3 RAS DFSMSrmm enhancements 41
3.3.1 UXTABLE RETPD change 41
3.3.2 SEARCHVOLUME and SEARCHDATASET 41
3.3.3 CHANGEVOLUME and CHANGEDATASET 41
3.3.4 Use of expiration time 42

Chapter 4. DFSORT 43
4.1 DFSORT overview 44
4.2 Functional enhancements 44
4.2.1 AGE function 44
4.2.2 WEEKNUM function 45
4.3 Use of zHPF 46
4.4 Message ICE099A update 47
4.5 Joinkey indicators in SMF data 47

Chapter 5. Virtual Storage Access Method 49
5.1 VSAM overview 50
5.2 VSAM performance enhancements 50
5.2.1 Faster search for Access Method Block List 50
5.2.2 Chained I/O for spanned records 51
5.3 VSAM RAS enhancement 53
5.3.1 Support for DB2 greater than 2 M opened clusters. 53
5.3.2 Message on all auto dumps 54
5.3.3 Improved recovery for EOV with non-SMS clusters 55
5.3.4 Dynamic Buffer Addition function 55
5.4 VSAM RLS enhancements 56
5.4.1 Index Record Locking 56

Chapter 6. Catalogs and IDCAMS 59
6.1 Overview 60
6.2 Catalog RAS enhancements 60
6.2.1 Catalog modify command security enhancement 60
6.2.2 Catalog health check 60
6.2.3 LISTCAT statistics for catalogs 61
6.2.4 Restoring a catalog to Any Volume 62
6.2.5 Connecting multi-line Write To Operator (WTO) 62
6.2.6 New catalog search interface fields 62
6.3 IDCAMS enhancements 63
6.3.1 DIAGNOSE 63

6.3.2 PARM(TEST) command 63
6.3.3 REPRO MERGECAT command 63
6.3.4 TSO LISTCAT command 64
6.3.5 PRINT and REPRO commands 64
6.3.6 VERIFY RECOVER command 65
6.4 Generation Data Group enhancements 66
6.4.1 GDG extended 66
6.4.2 GDG SCRATCH Parmlib value 67
6.4.3 GDG PURGE keyword 68

Related publications 71
IBM Redbooks 71
Other publications 71
Help from IBM 72

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 grant 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites 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.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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

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

COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

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

AIX®
CICS®
DB2®
DS8000®
ECKD™
FICON®
FlashCopy®
GDPS®
IBM®
IBM z™
IBM z Systems™
MVS™
Parallel Sysplex®
RACF®
Redbooks®
Redpaper™
Redbooks (logo)  ®
RMF™
z Systems™
z/OS®

The following terms are trademarks of other companies:

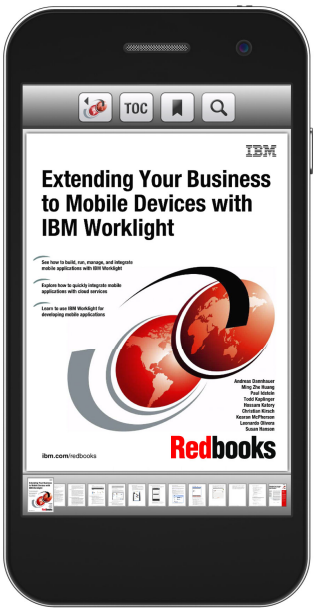
Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get up-to-the-minute Redbooks news and announcements
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication helps you to become familiar with the technical changes that were introduced into the Storage Management and Utilities areas with IBM z/OS V2R2.

This book is one of a series of IBM Redbooks publications that take a modular approach to providing information about the updates that are included with z/OS V2R2. This approach has the following goals:

- Provide modular content
- Group the technical changes into a topic
- Provide a more streamlined way of finding relevant information that is based on the topic

We hope you find this approach useful and we welcome your feedback.

Authors

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

Keith Winnard is the z/OS Project Leader at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and is keen to engage with customers to understand what they want from IBM Redbooks Publications. Before joining the ITSO in 2014, Keith worked for clients and Business Partners in the UK and Europe in various technical and account management roles. He is experienced with blending and integrating new technologies into the traditional landscape of mainframes.

Natalia Barros is a Systems Programmer at Banco do Brasil, a government bank in Brazil. She has seven years of mainframe experience and holds a degree in Computer Science. Her areas of expertise include storage architecture, mainframe hardware configuration, Data Facility Storage Management Subsystem (DFSMS), and remote copy solutions.

Jose Gilberto Biondo Jr is an IT Specialist in Integrated Technology Delivery, ServerSystems Operations/Storage Management in IBM Brazil. He has seven years of experience inz/OS, working with storage management since 2007. Jose works mainly with IBM storage products (DFSMSdfp, DFSMSdss, DFSMShsm, and DFSMSrmm), but he also works with OEM software products. Jose's areas of expertise include installing and maintaining storage products, and process automation.

Jaqueline Mourão is a Senior Systems Programmer at Banco do Brasil, a government bank in Brazil. She has 13 years of mainframe experience and holds a degree in Information Systems. Her areas of expertise include z/OS installation and maintenance, IBM Parallel Sysplex, and security.

Alvaro Salla is a Senior IT Consultant for the ITSO. He has more than 40 years of teaching and developing educational material experience that includes the z/OS mainframe platform and consulting services focusing on performance. Alvaro has co-authored Redbooks about IBM DS8000, DFSMS, WLM, SYSPLEX, ABCs, IBM RMF, and disaster recovery. He started with IBM in 1969.

Ewerton Waki is an IT Analyst at Banco do Brasil, a government bank in Brazil. He has four years of experience in mainframe systems, including IBM GDPS, DFSMS, high-end storage systems, and remote copy solutions. He holds a Bachelor in Information Systems from Universidade Estadual de São Paulo. His area of expertise includes DFSMS and storage performance for mainframe environments.

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

Barbara McDonald (DFSMS Product Strategist, IBM z Systems Software, San Jose) for offering valuable advice and guidance on the new features of DFSMS and the related products that are covered in this IBM Redbooks publication.

Glenn Wilcock (Architect, DFSMShsm Development team, Tucson) for technical support, advice, and guidance on DFSMShsm throughout the creation of this IBM Redbooks publication.

James Cammarata (IBM) for technical support, advice, and guidance on DFSMShsm throughout the creation of this IBM Redbooks publication.

Alexsey Redko (IBM) for technical support, advice, and guidance on DFSMShsm throughout the creation of this IBM Redbooks publication.

Arseney Khobotkov (IBM) for technical support, advice, and guidance on DFSMSrmm throughout the creation of this IBM Redbooks publication.

Alberto Ferreira Aquino (IBM) for technical support, advice, and guidance on DFSMSrmm throughout the creation of this IBM Redbooks publication.

Sri Hari Kolusu (Advisory Software Engineer, DFSORT Development, San Jose) for technical support, advice, and guidance on DFSORT throughout the creation of this IBM Redbooks publication.

Tan Q Nguyen (Software Engineer, DFSMS Development at Almaden Research Center) for technical support, advice, and guidance on SMS and DFSMS throughout the creation of this IBM Redbooks publication.

Stephen Branch (ICF Catalog and IDCAMS Development, IBM) for technical support, advice, and guidance on IDCAMS and DFSMS throughout the creation of this IBM Redbooks publication.

Thanks to the following people for their environment and residency support contributions to this project:

Bob Haimowitz (DST, Poughkeepsie Center) for setting up and maintaining the systems, and providing valuable advice, guidance, and assistance throughout the creation of this IBM Redbooks publication.

Rich Conway (DST, Poughkeepsie Center) for setting up and maintaining the systems, and providing valuable advice, guidance, and assistance throughout the creation of this IBM Redbooks publication.

Peter Bertolozzi (Systems Management specialist, IBM Redbooks residency support, Poughkeepsie Center) for setting up and maintaining the environments in which residents worked.

John Gierloff (Operations, Poughkeepsie Center) for residency setup and support.

Don Brennan (DST, Poughkeepsie Center) for setting up and maintaining the systems hardware that is used in the creation of this IBM Redbooks publication.

Ella Buslovich (Graphics specialist, location) for providing guidance and specialist graphics for this IBM Redbooks publication.

Ann Lund (ITSO Administration, Poughkeepsie Center) for administrative support to enable the residency.

Cheryl Gera (ITSO Administration, Poughkeepsie Center) for managing the business operations for this IBM Redbooks publication.

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 by using leading-edge technologies. Your efforts help to increase product acceptance and customer satisfaction as you expand your network of technical contacts and relationships. Residencies run 2 - 6 weeks in length and you can participate 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 this website:

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

DFSMSdfp

This chapter describes the changes that were made to DFSMSdfp that was introduced with z/OS V2R2.

This chapter includes the following topics:

- 1.1, “DFSMSdfp overview” on page 2
- 1.2, “DFSMSdfp enhancements” on page 2
- 1.3, “SMS space constraint relief enhancements” on page 3
- 1.4, “User-defined ACS read-only variable” on page 5
- 1.5, “SMS RAS enhancements” on page 6
- 1.6, “SMF VTOC audit log” on page 10
- 1.7, “Open/Close/End of Volume” on page 12

1.1 DFSMSdfp overview

The DFSMSdfp is a Data Facility Storage Management Subsystem (DFSMS) element that provides the following storage, data, program, and device management:

- Interactive Storage management

  DFSMSdfp includes Interactive Storage Management Facility (ISMF), with which you define and maintain policies to manage your storage resources.

- Tape mount management

  Tape mount management is a methodology for improving tape usage and reducing tape costs.

- Data management

  DFSMSdfp helps store and catalog information on direct access storage devices (DASD) and optical and tape devices so that it can be quickly identified and retrieved from the system.

- Device management

  DFSMSdfp can be used when you define your input and output (I/O) devices to the system and in controlling those devices in the z/OS environment.

- Distributed data access

  Distributed data access allows all authorized systems and users in a network to use system-managed storage or automated storage management.

- Advanced copy services

  Advanced Copy Services includes remote and point-in-time copy functions that provide backup and recovery of data.

- Object access method

  Object Access Method (OAM) provides storage, retrieval, and storage hierarchy management for objects.

1.2 DFSMSdfp enhancements

The following DFSMSdfp enhancements are available in z/OS V2R2:

- SMS space constraint relief
- New user-defined Automatic Class Selection (ACS) read-only variable in SMS PARMLIB
- SMS RAS:
  - New information console message
  - Enhance SELECT/DESELECT command
  - Include data set name and storage name in ACS messages
  - Display storage groups in IGD17800I
  - Externalize IGD17364I to hardcopy log
- Storage group space alert messages
- SMF Volume Table of Content (VTOC) audit log
- DFSMSdfp open/close/end volume
  - Tape performance and recovery
  - Conversion of tape installation exits

- DEVSUPxx
- SMF 14/15
- Include above the line CSECTs
- Eliminate abend A13-18

These enhancements are described in this chapter.

1.3 SMS space constraint relief enhancements

The space constraint relief enhancements that are described in this section improve data set allocation.

1.3.1 Space constraint relief for guaranteed space allocations

During allocation, there might be insufficient space on a volume to meet the requested space. SMS volume selection might solve the problem for non-guaranteed space allocations by trying all candidate volumes before failing the allocation. Another option is to use the Space Constraint Relief and Reduce Space Up To (%) attributes to request that an allocation be retried if it fails because of space constraints.

In z/OS V2R2, this functionality was extended to guaranteed space allocations to help reduce these types of failures. The SCR space reduction function was enhanced to allocate the largest possible space that satisfies the value (percentage) for guaranteed space and non-guaranteed space requests.

The new subparameter, Guaranteed Space Reduction (Y | N), is added to the Data Class Panel to specify whether space reduction on guaranteed space allocation is permitted. The new subparameter is shown in Example 1-1.

Example 1-1 New subparameter in Data Class Panel

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

CDS Name  . . . . . : SYS1.STPPLEX.SCDS
Data Class Name . . : EXTADD

Data Set Name Type  . . . . . : EXTENDED
  If Extended . . . . . . . . : REQUIRED
  Extended Addressability . . : YES
  Record Access Bias  . . . . : USER
  RMODE31 . . . . . . . . . . :
Space Constraint Relief . . . : NO
  Reduce Space Up To (%)  . . :
  Guaranteed Space Reduction  : NO
  Dynamic Volume Count  . . . :

Compaction  . . . . . . . . . :
Spanned / Nonspanned  . . . . :
```

Note: Guaranteed space reduction affects VSAM and non-VSAM data sets and remains unsupported for striping allocation.

1.3.2 Space constraint relief for secondary allocation

Before z/OS V2R2, the space constraint relief (SCR) applies only in the Primary Space allocation; that is, SCR was not used if the data set was extended by using the Secondary Space amount of allocation.

A situation can occur in which large amounts of free space that is less than what the user specified for secondary allocation amount is left unused on the current volume, or the allocation can fail when there are no more volumes to be used.

With z/OS V2R2, SCR also applies to secondary allocation.

Another improvement in SCR is to use the largest available extent on volume. Consider a scenario that includes the following parameters:

- Secondary Space allocation of 2000 cylinders
- Reduce Space Up To(%) in Data Class of 90%
- Largest extent available on volume is 1000 cylinders

In this case, when the data set needs another extent, the allocated amount is 1000 cylinders, which represents a 50% of reduction.

To enable space reduction, the Data Class must include the following settings:

- Space Constraint Relief = Y
- Reduce Space Up To (%) > 0

The following new SMF fields were added to track the use of this function:

- SMF64SSR: Indicates whether secondary space reduction was used
- SMF64NTA: Indicates the size of extent that is returned in tracks

This enhancement supports VSAM/VSAM RLS/PDSE/BAM/SAM; however, some VSAM KSDS allocations result in minimum space that is allowed because CATALOG and DADSM cannot find space to allocate INDEX after the largest extent available was used for DATA. To avoid the failure in this case, CATALOG and DADSM return to their previous behavior, which allocates the minimum space allowed that is derived from Reduce Space Up To (%).

Note: VSAM and non-VSAM multi-striped data sets do not support SCR. For more information, see *DFSMS Using Data Sets*.

1.3.3 New IGD17296I message

Another enhancement is the new message, `IGD17296I`. This message is issued to the hardcopy and job logs when SMS Dynamic Volume Count (DVC) function is used to allocate or extend a data set, as shown in Example 1-2.

Example 1-2 New message IGD17296I

```
IGD17296I DYNAMIC VOLUME COUNT (DVC=nn) WAS USED TO {ALLOCATE|EXTEND}
DATA SET dsn
```

DVC is a field in the data class that defines the maximum number of volumes that data set can span and is used during the allocation or extend processing if the specified volume count is not sufficient. This new message allows the user to take preventive actions for potential space failures.

1.4 User-defined ACS read-only variable

New user-defined ACS read-only variables for use with ACS routines were introduced. This functionality allows you to specify unique values for your own use. They are similar to how systems symbols are defined and can be used to assign proper constructs without a need to modify the ACS routines.

The new **USER_ACSVAR** parameter is optional and can be specified in your Parmlib member **IGDSMSxx**. The new parameter has three positional values. Each value is a user-defined, eight-character string, as shown in the following example:

```
USER_ACSVAR(value1,value2,value3)
```

In addition to user-defined values, you can also use any symbols that are defined on IEASYMxx member on your environment. If you do not define USER_ACSVAR on your IGDSMSxx member, the default values are set to blank.

The ACS routines can reference each positional value as *&USER_ACSVAR(1)*, *&USER_ACSVAR(2)*, or *&USER_ACSVAR(3)*.

The values for variables are saved when SMS is started. During ACS processing, the variables are passed to the ACS routines in the form of a new ACS read-only variable. All ACS routines can access the new variables.

The value can be altered with **SETSMS** command, as shown in the following example:

```
SETSMS USER_ACSVAR(value1,value2,value3)
```

When setting USER_ACSVAR via SETSMS command, you can use a single dash (-) to keep current values that you do not want to change, as shown in the following example:

```
SETSMS USER_ACSVAR(-,-,PROD)
```

If any positional value is not specified in the command, it is set to blank. The following examples show setting the first, second, and third variables to blank:

- `SETSMS USER_ACSVAR(,-,PROD)`
- `SETSMS USER_ACSVAR(&SYSCLONE,,TEST)`
- `SETSMS USER_ACSVAR(&SYSNAME,-)`

Any combination of dash, blank, user values, or system symbols are allowed.

Example 1-3 uses USER_ACSVAR, which includes SYSPLEX name values to define the correct DATA CLASS to be assigned on ACS routines.

Example 1-3 Sample DATACLASS ACS code

```
PROC DATACLAS              /* DATA CLASS ACS ROUTINE */
   IF &USER_ACSVAR(1) = 'PRODPLX' THEN
      SET &DATACLAS = 'PROD_DC'
   ELSE
      IF &USER_ACSVAR(1) = 'TESTPLX' THEN
         SET &DATACLAS = 'TEST_DC'
      ELSE
         WRITE '***** UNEXPECTED VALUE FOR USER_ACSVAR'
   EXIT CODE(0)
END                        /* END DATA CLASS ROUTINE */
```

Note: To check the USER_ACSVAR content, you can issue a **D SMS,OPTIONS**. Use of the **WRITE** command on ACS routines displays the first value only.

Because new variables are defined on IGDSMSxx and ACS routines, systems that share SMS Control Data Set (CDS) should not start using USER_ACSVAR until all LPARs are running z/OS V2.2.

1.5 SMS RAS enhancements

As in previous releases, DFSMS in z/OS V2.2 continues to provide RAS enhancements.

1.5.1 New information console message

DFSMS serializes access to SMS control data sets, ACDS, and COMMDS, by issuing a **RESERVE** with the resource name **IGDCDSXS**.

The SMS resource name, **IGDCDSXS**, must be placed in the GRS RESERVE conversion RNL as a generic entry so it can be converted to the global ENQ. This configuration minimizes delays that can occur because of contention for resources and prevents potential deadlocks.

In z/OS V2.2, if **IGDCDSXS** is not specified in the GRS RESERVE conversion RNL, SMS issues a new informational message (**IGD06041I**) to the console, as shown in Example 1-4.

Example 1-4 New message IGD06041I

```
IGD06041I SMS RESOURCE NAME IGDCDSXS IS NOT FOUND IN GRS RESERVE CONVERSION RNL.
RETURN CODE retcode REASON CODE rsncode
```

1.5.2 Enhance SELECT/DESELECT command

The commands SELECT/DESELECT, are used to specify or delete one or more events or services that SMS is to trace.

Before z/OS V2.2, the **SELECT/DESELECT** command cannot have more than two lines in **IGDSMSxx** PARMLIB member. In z/OS V2.2, SMS provides support to have more than two lines in the **SELECT/DESELECT** command for better serviceability.

Migration and coexistence considerations: If you share **IGDSMSxx** PARMLIB member, ensure that all systems are on z/OS 2.2 before this enhancement is used.

1.5.3 Including data set name and storage name in ACS messages

In previous releases, SMS issued ACS messages **IGD01012I** and **IGD01015I** when data set allocation failed, as shown in Example 1-5.

Example 1-5 IGD01012I and IDG0105I Messages

```
IGD01012I DATA SET ALLOCATION REQUEST FAILED - THE ACS STORAGE GROUP ROUTINE
ASSIGNED A NON-TEMPORARY OR VSAM DATA SET TO A STORAGE GROUP WHICH IS NOT OF TYPE
POOL
IGD01015I DATA SET ALLOCATION REQUEST FAILED - THE ACS STORAGE GROUP ROUTINE
ASSIGNED A DSNTYPE=type
DATA SET TO A NON POOL TYPE STORAGE GROUP
[CANDIDATE STORAGE GROUP(S): sg1, sg2,...]
```

In z/OS V2.2, these messages are enhanced to include the data set name and the storage group name for problem diagnosis.

SMS trace entries that are related to these events are also enhanced.

1.5.4 Display Storage Groups in IGD17800I Message

Previous releases of SMS issued message **IGD17800I** when the specified volumes for a guaranteed space request cannot be found in the eligible storage group list. The storage groups that are searched by SMS is not externalized to the user.

The message **IGD17800I** is enhanced to display the storage groups that are searched. The new message format is shown in Example 1-6.

Example 1-6 IGD17800I Message

```
IGD17800I {DATA CLASS dcname| MANAGEMENT CLASS mcname| STORAGE CLASS scname|
STORAGE GROUP sgname|VOLUME volser}
DEFINITION NOT FOUND FOR DATA SET dsname
```

1.5.5 Externalizing message IGD17364I to hardcopy log

The RETENTION LIMIT value in the Management Class limits the use of a retention period and expiration date. Data sets that are assigned with a RETENTION LIMIT value of zero days that are specified in their Management Class are immediately expired.

Before z/OS V2.2, job logs had to be reviewed to find message **IGD17364I** to identify which data sets expired for this reason.

This new functionality also externalizes **IGD17364I** to the hardcopy log. By using this function, the expired data sets can be identified by searching for the **IGD17364I** messages in the hardcopy log, as shown in Example 1-7.

Example 1-7 IGD173264I Message

```
IGD17364I DATA SET dsname NOT AUTHORIZED TO EXPIRATION DATE SPECIFIED EXPIRATION
DATE RESET TO MAXIMUM ALLOWED yyddd
```

1.5.6 Storage Group Space alert messages

In previous releases, SMS VTOC Data Set Services (VDSS) issued message **IGD17380I** to notify a user that the cumulative space that was allocated on the selected storage group exceeded the high allocation threshold.

It might be better to change the alert based on a higher or lower value thresholds to capture more meaningful alerts. DFSMS V2.2 includes new alert thresholds for pool storage groups. These new alerts allows users to have more time to react to storage group space shortage conditions and helps to identify when you must change a storage group's space management settings or add volumes to a storage group.

The new threshold attributes in the storage group panel to define thresholds are shown in Example 1-8.

Example 1-8 New attributes in Storage Group Display

```
POOL STORAGE GROUP DISPLAY                    Page 2 of 2
Command ===>

CDS Name  . . . . . : SYS1.STPPLEX.SCDS
Storage Group Name  : OPENMVS

Dump Class . . . . . . . . . . . . . . . :
Dump Class . . . . . . . . . . . . . . . :
Dump Class . . . . . . . . . . . . . . . :
Allocation/Migration Threshold - High  . : 85
                                 Low . . :  1
Alloc/Migr Threshold Track-Managed - High: 85
                                     Low :
Total Space Alert Threshold %  . . . . . :
Track-Managed Space Alert Threshold %  . :
Guaranteed Backup Frequency  . . . . . . :
BreakPointValue  . . . . . . . . . . . . :
Processing Priority  . . . . . . . . . . : 50


Use UP Command to View previous Page;
Use HELP Command for Help; Use END Command to Exit.
```

When no threshold value is specified, it is set to zero and no alert messages is issued. However, when the alert threshold is reached, the new alert messages **IGD400I** and **IGD401I** are issued to the console, as shown in the Example 1-9.

Example 1-9 IGD400I and IGD401I alert messages

```
IGD400I TOTAL SPACE ALERT ON STORAGE GROUP (sgname) CURRENT USAGE (xx%), ALERT
THRESHOLD (yy%)
IGD401I TRACK-MANAGED SPACE ALERT ON STORAGE GROUP (sgname)CURRENT USAGE (xx%),
ALERT THRESHOLD (yy%)
```

To prevent the alert messages being issued too frequently, the messages are issued at an incremental interval. The calculated space usage includes “online and enable” volumes only and occurs when the following conditions are present:

- A space change occurs
- Varies the online or offline volume of a pool storage group
- Disables or enables an online volume of a pool storage group

Note: The **IGD401I** alert is issued only for the pool storage group that contains one or more EAV volumes.

The **IGD002I** message includes new information about total and track-managed space for each storage group, as shown in Example 1-10.

Example 1-10 DISPLAY SMS,SG(stgname)

```
COMMAND INPUT ===>                                            SCROLL ===> PAGE
RESPONSE=SC74
 IGD002I 15:30:43 DISPLAY SMS 108

 STORGRP  TYPE    SYSTEM= 1 2 3 4 5
 OPENMVS  POOL            + + + + +
   SPACE INFORMATION:
   TOTAL SPACE = 16240 MB USAGE% = 83 ALERT% = 80
   TRACK-MANAGED SPACE = 16240 MB USAGE% = 83 ALERT% = 80
 ***************************** LEGEND ****************************
 . THE STORAGE GROUP OR VOLUME IS NOT DEFINED TO THE SYSTEM
 + THE STORAGE GROUP OR VOLUME IS ENABLED
 - THE STORAGE GROUP OR VOLUME IS DISABLED
 * THE STORAGE GROUP OR VOLUME IS QUIESCED
 D THE STORAGE GROUP OR VOLUME IS DISABLED FOR NEW ALLOCATIONS ONLY
 Q THE STORAGE GROUP OR VOLUME IS QUIESCED FOR NEW ALLOCATIONS ONLY
 > THE VOLSER IN UCB IS DIFFERENT FROM THE VOLSER IN CONFIGURATION
 SYSTEM  1 = PLEX75     SYSTEM  2 = SC72       SYSTEM  3 = SC73
 SYSTEM  4 = SC74       SYSTEM  5 = SC75
```

In addition, a new keyword, ALERT, allows the message **IGD002I** to display all of the storage groups that reached one of their alert thresholds. When no storage groups reach the threshold, a new version of the **IGD004I** message is issued, as shown in Example 1-11.

Example 1-11 New version of IGD004I

```
IGD004I NO STORAGE GROUPS HAVE REACHED THEIR ALERT THRESHOLDS
```

Note: After an activation of a new configuration, the **DISPLAY SMS,SG** command might show “UPDATE STILL IN PROGRESS” in some of the storage group space information. Run the **V SMS,{STORGRP(storgrp)|SG(storgrp)|VOLUME(volser)|VOL(volser)},{SPACE|S}** command to force the space information to be updated for the storage group.

1.6 SMF VTOC audit log

The VTOC consists of Data Set Control Blocks (DSCBs) that describe the data sets that are on a volume. The life of a DSCB is managed by DFSMS software. However, other software can also update a DSCB.

Before z/OS V2R2, vendor's software updates were not recorded in an audit log.

1.6.1 Changes in z/OS V2R2

In z/OS V2.2, the audit logging of VTOC I/O is improved. The logging is done with a new SMF42 Subtype 27 record that captures updates to the VTOC for IBM and vendor-built channel programs.

When a DSCB is written, the following information is recorded:

- Date and time
- System ID, Job Name, Job number, Step name, and Product name
- User Security Token (mapped by ICHRUTKN)
- Activity (from IOBEUSER in IOB Extension):
  - Dxxx: DFSMS Activity
  - DCVF: CVAFDIR
  - DCRE, DREN: Data set create and rename
  - DEXT, DPAR: Data set extend and partial release
  - DDEL: Data set scratch (read before erase)
  - DFRG, DCON: DFSMSdss defrag and consolidate
  - DDMP, DRST: DFSMSdss dump and restore
  - IOBE: IOBE not provided
  - USER: IOBEUSER not specified
- Volume Serial and Device ID (UCB Channel number)
- Seek and Search ID (CCCC HHHH R)
- Caller's PSW following EXCP SAN Volume Controller
- Device is Reserved flag
- DSCB Key and Data field (140 bytes)

To disable VTOC audit logging, you can use the **NOTYPE** keyword in parmlib member **SMFPRMxx**, as shown in Example 1-12.

Example 1-12 Disable VTOC audit logging

```
NOTYPE(42(27))
```

EXCP can detect the following writes to the VTOC:

- CKD and IBM ECKD™ channel programs
- DADSM/CVAF
- DSS Defrag/Consolidate, Copy/Restore/Dump
- Vendor channel programs
- XDAP macro instruction (adds support for IOB Extension block)
- SuperZAP (AMASPZAP service)

The following writes are not logged:

- System Resident volume
- Temporary DASD data sets
- I/O to an offline volume
- Updates to DSCB FMT 1, 5, and 7; second DSCB written in channel program
- Use the IOBUSER field in the Input/Output Block (IOB) Extension to Update the VTOC for programs that builds its own EXCP channel program. This field is saved in the SMF 42 subtype 27 record, as shown in Figure 1-1 and Figure 1-2.

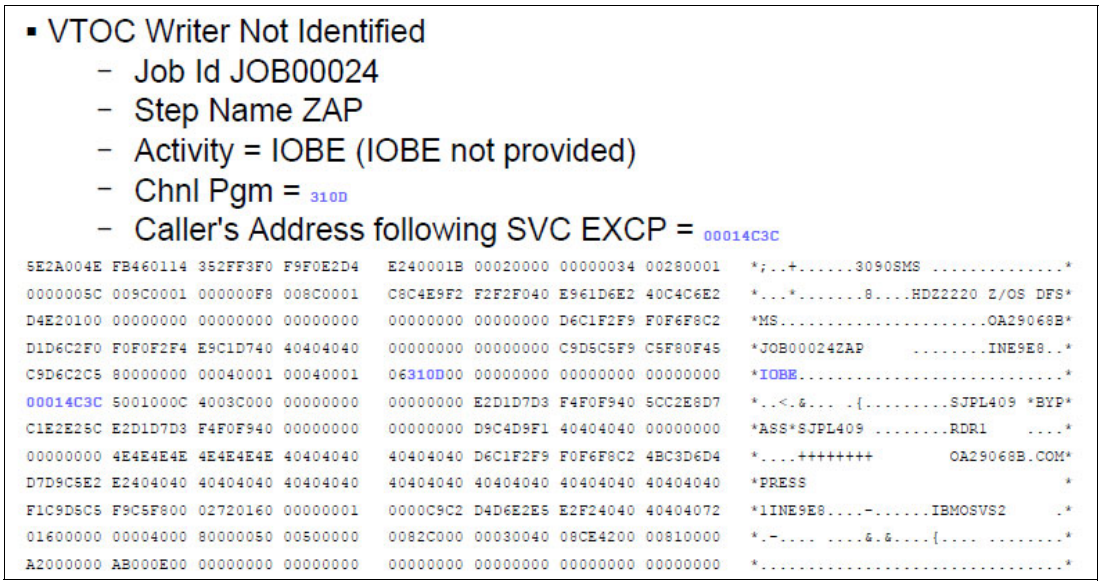


Figure 1-1 VTOC SMF 42 subtype 27 example: VTOC Writer Not Identified

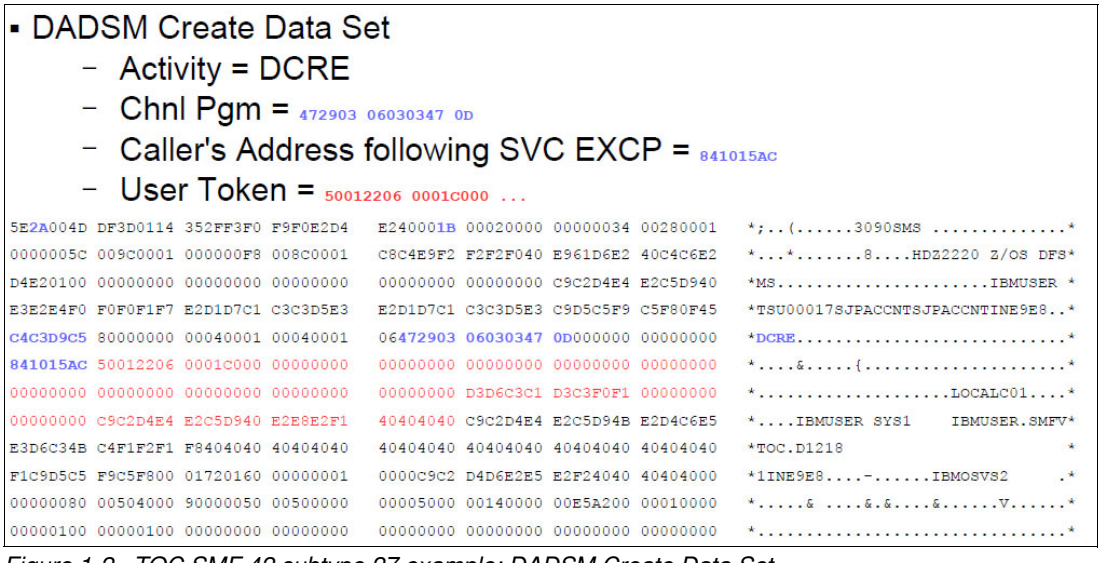


Figure 1-2 TOC SMF 42 subtype 27 example: DADSM Create Data Set

1.7 Open/Close/End of Volume

This section provides detailed information about Open/Close/End of Volume enhancements.

1.7.1 Tape performance and recovery

To keep data synchronized between cache and tape, the drive cache must be synchronized with the tape medium.

When a multi-file synchronization fails, all files must be rewritten because there is no way to determine what files were safely written to the tape medium.

In z/OS v2R2, it is possible to specify how many files should be written to tape volumes by a specific job before a synchronization occurs and determine the specific compromised files if there is a synchronization failure.

These enhancements can improve performance by avoiding the overhead and time that is used by unnecessary synchronizations. This enhancement minimizes the number of synchronizations that are performed to the optimum number and improves recovery by providing a means to determining the specific compromised files if there is a synchronize failure.

The new keyword, SYNC=(NUMFILES,'nnn'), must be set in DCBE macro to specify how many files must be written to tape volumes.

The keyword accepts a value of 1 - 32000 that is stored in the new DCBENMFL field. Up to 'nnn' consecutive files can be written that specify PASS RETAIN or CLOSE LEAVE before an explicit channel program is written, as shown in Figure 1-13.

Example 1-13 New keyword SYNC

```
DCBAD DCB DDNAME=DD1,DSORG=PS,MACRF=PM,BLKSIZE=80,RECFM=F,DCBE=DCBEB
DCBEB DCBE RMODE31=NONE,SYNC=(NUMFILES,100)
```

All data from those files, including file boundaries (tape marks), is buffered in the device cache. After 'nnn' files are written, a single explicit synchronization occurs.

If the synchronization fails, the message `IEC999I` is written to the system and job logs. The job then abends, as shown in Example 1-14.

Example 1-14 Message IEC999I

```
IEC999I jobname,stepname,volser LOST BLOCKS START: FILE 'n'
```

The name of next data set to rewrite can be found in `IEC205I` message or in the abend message.

1.7.2 Converting tape installation exits

The open, close, and end-of-volume (EOV) tape management exits allow tape management systems to avoid changing system control blocks or issuing channel programs against tapes to maintain a tape inventory.

During opening, processing, and closing a tape data set, the system calls the File Validation, File Start on volume, and File End on volume. Open or EOV calls the file validation exit for each volume, regardless of whether it was mounted or verified.

Many conditions affect whether various exits are called and their sequence. For example, during a call to open or EOV, the following sequence is typical:

- File end on volume exit (if EOV)
- Volume mount exit for volume verification (if volume was not mounted and verified)
- Volume mount exit for volume security (if volume was not mounted and verified)
- File validate exit (if labels exist)
- Volume mount exit for volume write function (if the tape has standard labels, the user is writing and the file is the first file on the volume)
- File start on volume exit

In z/OS V2R2, the Installation Tape Exits were converted to be dynamic. Now, an initial program load (IPL) is not necessary to manage changes with Tape Exits and multiple exit routines can be associated with each exit.

Table 1-1 shows the dynamic exit names, and their respective default exit routines.

Table 1-1 Dynamic exit names and default exit routines

Exit Name	Default Exit Routine
OCE_VOLUMEMOUNT	IFG019VM
OCE_FILESTART	IFG019FS
OCE_FILEVALIDATE	IFG019FV
OCE_FILEEND	IFG055FE
OCE_LABELANOMALY	IFG019LA

When multiple exits are associated, the order in which they are called is not defined unless the options **FIRST** or **LAST** are used. The parameter list can be changed by any exit routine, and, in all cases, the parameter list contains last changes from the last called exit routine.

Examples the use multiples exits routines that are defined in PROGxx PARMLIB member are shown in Example 1-15.

Example 1-15 Multiple exits routines

```
EXIT ADD
      EXITNAME(OCE_VOLUMEMOUNT)
      MODNAME(MYEXIT1)
      STATE(ACTIVE)
      DSNAME(MY.PDS)
      FIRST
EXIT ADD
      EX(OCE_FILESTART)
      MOD(MYEXIT2)
      LAST
EXIT DELETE
      EX(OCE_FILEEND)
      MOD(IFG055FE)
```

Also, if an exit routine ABENDs, it is made inactive immediately and stays inactive until it is activated manually or the system is loaded.

The following commands are available:

- To display Open/Close/EOV tape installation exits, use the following command:

  ```
  D PROG,EXIT,EXITNAME=OCE_*
  ```

- To display diagnostic data, including exit routines for Volume Mount, use the following command:

  ```
  D PROG,EXIT,EX=OCE_VOLUMEMOUNT,DIAG
  ```

- To add MYEXIT1 exit routine in MY.PDSE data set to File Start exit, use the following command:

  ```
  SETPROG EXIT,ADD,EX=OCE_FILESTART,MODNAME=MYEXIT1,DSN=MY.PDSE
  ```

- To add and make MYEXIT2 exit routine to be called the first every time after File End exit gets control, use the following command:

  ```
  SETPROG EXIT,ADD,EX=OCE_FILEEND,MOD=MYEXIT2,FIRST
  ```

- To make MYEXIT3 exit routine inactive for File Validate exit, use the following command:

  ```
  SETPROG EXIT,MODIFY,EX=OCE_FILEVALIDATE,MOD=MYEXIT3,STATE=INACTIVE
  ```

- To delete MYEXIT4 exit routine from File End exit, use the following command:

  ```
  SETPROG EXIT,DELETE,EX=OCE_FILEEND,MODNAME=MYEXIT4
  ```

1.7.3 DEVSUPxx enhancements

Device Support Options (DEVSUPxx) specifies the installation default for device support options. DEVSUPxx is processed during the NIP phase of IPL. After IPL, customers can use system command SET DEVSUP=XX to activate the DEVSUP changes.

Multiple DEVSUP member support

In previous releases, the command SET DEVSUP MVS™ and the keyword DEVSUP in IEASYSxx PARMLIB member support single value only. z/OS V2.1 includes an enhanced Multiple Virtual Storage (MVS) command and keyword to support multiple values. The enhancement is shown in Example 1-16.

Example 1-16 DEVSUP multiple values

```
SET DEVSUP=(xx,yy,zz...)
DEVSUP=(xx,yy,zz...)
```

Missing DEVSUP member message

When **SET DEVSUP=xx** command was issued but the DEVSUPxx member did not exist, the following message was issued:

```
IEE536I DEVSUP VALUE xx NOW IN EFFECT.
```

In z/OS V2R2, this message is not issued if DEVSUPxx member does not exist. In addition, **IEE538I** message is displayed to alert the user that the required DEVSUPxx was not found on parmlib.

Continued processing after keyword error

In z/OS V2R2, DEVSUP parameter processing does not stop after a keyword error is encountered. Instead, it issues the message **IEA25IE**, which was enhanced to display the statement in error, and the DEVSUPxx member it is coded, as shown in Example 1-17.

Example 1-17 IEA251E message

```
IEA251E DEVSUP8A: NON_VSAM_XTIOT PARAMETER 'TES' NOT VALID.
IEA252E DEVSUP8A: INVALID SYNTAX ON LINE 0002. MEMBER PROCESSING CONTINUES.
```

Preserve keyword values

In previous releases, there is a subset of DEVSUPxx keywords that were reset to their default values whenever a member was processed, unless the values were explicitly set in the member. However, this process created an inconsistency in how all keywords are processed.

In z/OS V2R2, the current values of all keywords are preserved when a DEVSUPxx member is processed, unless the member explicitly specifies new values.

1.7.4 Adding JOBID and SYSPLEXID SMF 14 and 15 enhancements

SMF writes a type 14 or 15 record whenever a data set is closed or processed by EOV. These records can be used to gather information about the user that is accessing the data set, date and time, and type of access, and others.

Usage and invocation

In z/OS V2.2, the JCTJOBID, JES2 job identifier, and SYSPLEXID are added to SMF 14 and 15 to give more information about where a job ran. The new field of SMF 14 and 15 records are automatically included when they are requested.

1.7.5 Including above the line CSECTs in OCE in-core module table

The Open, Close, and End-of-volume (OCE) functions maintains an in-core module table with the following entries:

- CSECT name
- LPA address
- PTF level

Currently, this table does not contain any of 12 CSECTs of IFG019RB module.

In z/OS V2R2, these CSECTs of IFG019RB module are included in an in-core module table above the line, which eliminates requests from Level 2 for PTF's level.

1.7.6 Eliminating A13-18 while ensuring correct mount

Before z/OS V2R2, the ABEND A13 RC=18 occurred in the following situations:

- When an EOV1 label is read on the last SL or AL tape volume while forward spacing to the wanted file or just before the wanted file. The following conditions are possible:
  - If opening to the end of the file, it cannot be treated as the end of the data set because it was for a previous file sequence number. Probable user error.

  - If the user is trying to extend a data set that abended during EOV, it is necessary to specify a volume count on the DISP=MOD DD statement. This volume count must exceed the number of known volumes for the data set.

- When there are multiple DDs for tape data sets to be created in the same step. The first DD contains a specific volser and the subsequent DDs specify a VOLREF back to the first DD.
- When one of the data sets being written overflows to another volume, that volume is not known to the subsequent DDs in the job step.

In z/OS V2R2, this abend was eliminated. This new functionality enables the use of multiple DD statements, which passes the correct volume to the next DD statement and avoids errors. As shown in Example 1-18, DD1 spills to scratch volume 222222. When DD2 is processed, open replaces 111111 with 222222 in the JFCB.

Example 1-18 Sample JCL

```
//JOB ...
//STEP1 EXEC ...
//DD1 DD DSN=DATASET1,
// DISP=(NEW,PASS,KEEP),
// VOL=SER=111111,
// UNIT=920,
// LABEL=(1,SL)
//DD2 DD DSN=DATASET2,
// DISP=(NEW,PASS,KEEP),
// VOL=REF=*.STEP1.DD1,
// UNIT=920,
// LABEL=(2,SL)
```

2

DFSMShsm

This chapter describes the z/OS V2R2 updates to DFSMShsm and includes the following topics:

- 2.1, “DFSMShsm overview” on page 18
- 2.2, “Classic Migration, Storage Tiers, and Data Migration” on page 19
- 2.3, “ARCMDEXT support for transition” on page 21
- 2.4, “Fast Replication Patch non-DFSMShsm backup utility” on page 22
- 2.5, “Fast replication messaging in a consolidated log” on page 22
- 2.6, “DFSMShsm common dump queue” on page 25
- 2.7, “Dump class definitions MINSTACK and MAXSTACK” on page 33
- 2.8, “Multiple copy pools onto the same tape” on page 34
- 2.9, “New UPDTCDS command” on page 34

2.1 DFSMShsm overview

Hierarchical Storage Manager (HSM) was introduced in 1978 to help with storage management for data sets and to help manage the data sets based on simple criteria. Since then, there much more functionality was introduced to meet the management challenges of today's unparalleled growth in data capture, storage, movement, and retrieval in terms of availability, performance, and recoverability.

2.1.1 Space management

Space Management is an automatic and periodic function that deletes expired and temporary data sets, releases unused space, and makes the movement of the data across different levels of storage technologies possible.

A key objective is to keep direct access storage devices (DASDs) always available for new data and meet the service-level agreements.

2.1.2 Availability management

The purpose of availability management on DFSMShsm is to enable users to retrieve data as needed, which can include when the data is damaged or lost. There are automatic and periodic functions that copy data sets or even volumes to create an alternative copy. If the original data changed, DFSMS creates another copy during the automatic function to keep the most current level possible.

Note: DFSMShsm functions (space and availability management) for SMS-managed data sets is based on DFSMS Management Class and Storage Groups Definitions.

For non-SMS data sets and volumes, you must specify how you manage your data in DFSMShsm Parmlib member.

For more information, see the DFSMShsm Storage Administration and DFSMSdfp Storage Administration.

2.1.3 DFSMShsm enhancements

The following enhancements are available in DFSMShsm in z/OS V2R2:

- Classic Migration, Storage Tiers, and Data Migration
- ARCMDEXT Support for Transition
- Fast Replication Patch non-HSM backup utility
- Fast Replication Messaging in a consolidated log
- Introduction of DFSMShsm Common Dump Queue
- Dump Class Definitions new Parameters: MINSTACK and MAXSTACK
- Multiple Copy pools onto the same tape (Dump Function)
- New UPDTCDS command

These new features and enhancements are described next.

2.2 Classic Migration, Storage Tiers, and Data Migration

This section describes the changes in migration functions that are related to space management.

2.2.1 Classic Migration

Until z/OS V2R1, DFSMShsm was limited to starting a migration from Level 0 (L0) by using a command at the data set or volume level. To create free space in a specific storage group, it was necessary to issue several commands; DFSMShsm processed one command at time, although the parameter MAXMIGRATIONTASKS was set to a number greater than 1.

In z/OS V2R2, the following keyword was introduced to start a migration for all volumes within a STORAGE GROUP:

```
MIGRATE STORAGEGROUP(sgname, sgname, …)
```

This command starts primary space management for all volumes within this STORAGE GROUP and uses the number of the tasks that were configured in the parameter MAXMIGRATIONTASKS. Up to 30 storage groups can be specified in this command.

The goal is to significantly improve the throughput for command-started space management and improve usability because processing a large number of volumes in parallel can be achieved by using a single command.

2.2.2 Storage tiers

The storage tiers management was introduced as a different view of the classic three levels of storage technologies (L0, ML1, and ML2). L0 can now be split in different tiers that are based on the response time that is expected of each type of physical disk on the back-end of the storage subsystem controller.

Note: Enterprise storage subsystems can have solid-state drives (SSDs), Enterprise, and nearline hard disk drives (HDDs). A group of these HDDs can be called as a TIER.

Figure 2-1 on page 20 shows data being “transitioned” from Smart Tier 0 to Smart Tier 1 at the end of the year, and then becoming eligible for migration after 90 days of inactivity.

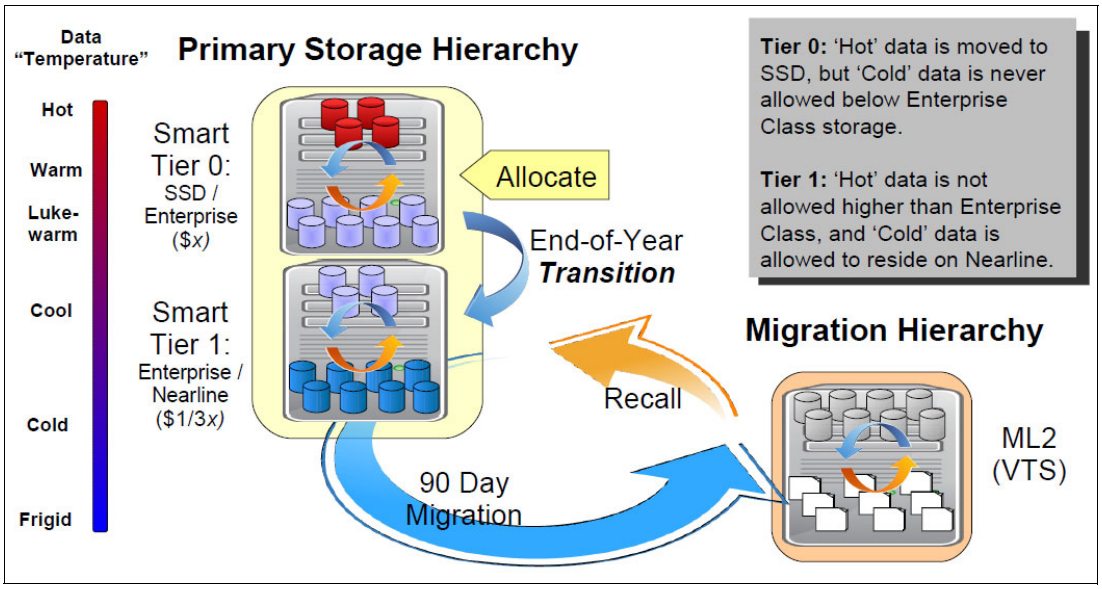


Figure 2-1 Storage tiers overview

Moving data between different levels of L0 includes the following advantages:

- Keeps data available without issuing RECALL
- Reduces CPU usage and removes unnecessary delays
- Ensures that higher performance devices are available to service the most important applications

If the storage subsystem controller has the FlashCopy® feature enabled, DFSMShsm can use this feature to move data faster by using the Storage Subsystem Controller (SSC). By moving this task to the SSC level, there is a reduction in the demand on general CPU usage; therefore, the CPUs are available to process other work.

2.2.3 Data migration

In z/OS V2R2, the various migrate commands are enhanced to support class transition at data set, volume, and storage group level.

The default behavior is to perform migration and transition processing for VOLUME and STORAGEGROUP operations.

The MIGRATE command features the following options:

- BOTH: Default, both migrations and transitions are performed
- MIGRATIONONLY: A data set is processed only if it is eligible for migration
- TRANSITIONONLY: A data set is processed only if it is eligible for a class transition

If a data set is eligible for migration and transition processing, it is migrated.

The default for MIGRATE DATASET is to perform a migration. The TRANSITION keyword indicates that a transition is performed.

Next, we describe the following methods that can be used to move data across different TIERs:

- DFSMSdss
- DFSMShsm

Use of DFSMSdss

In order to move a data set using DFSMSdss, you should:

- Check whether the data set is in use
- Close data set, each type of data set has a different method of closure
- Create a JOB with DFSMSdss (ADRDSSU) by using copy with delete option
- Reopen data set

Use of DFSMShsm:

Issue a command as shown in the following example:

```
MIGRATE DSNAME(...) TRANSITION
```

DFSMShsm performs all of the work with one command and starts the ACS routine by using **ACSENVIR = 'SPMGCLTR'**, with which you can assign new SMS constructs to the transitioned data set.

You can set up Serialization Error Exit keyword in Management Class to handle DB2®, CICS®, or zFS data sets that are selected for transition but are open by an application. This configuration enables DFSMShsm to communicate with the application and to close, transition, and reopen the data set properly.

You can also include the **SETSYS USERDATASETSERIALIZATION** command in your DFSMShsm PARMLIB to enable DFSMShsm to serialize (ENQ) only the data set during the data set processing.

The same can be done to move data within the same TIER. Suppose that you want to substitute the previously defined disk volumes by newly defined disk volumes. For example, this situation can occur when you substitute the storage subsystem controller or you want to use volumes with different sizes, such as EAV.

For SMS data, the new keyword **MOVE** was added to the **MIGRATE DSNAME, VOLUME** and **STORAGEGROUP** commands.

By using the **MOVE** keyword with the **MIGRATE** command, the following actions occur:

- Process every data set regardless of management class policy or threshold
- Start ACS routines to determine the new storage class and storage group
- Use management class serialization error logic and class transition movement techniques

Note: DFSMS Management Class Serialization Error logic must use the correct settings for each type of data. The volumes that are empty must be in DISNEW or DISALL state in the DFSMS storage group.

2.3 ARCMDEXT support for transition

The MD Exit was enhanced to support transition and migration, depending on certain conditions. A sample exit can be found in the SYS1.SAMPLIB. You can customize ARCMDEXT to make it applicable to your environment.

In z/OS V2R2, ARCMDEXT enables RC 20-40 to be used to override the destination, so the transition might be converted to a migration. For more information about z/OS V2R1, see APAR OA46627.

Figure 2-2 shows an example of MD Exit that was started during primary space management.

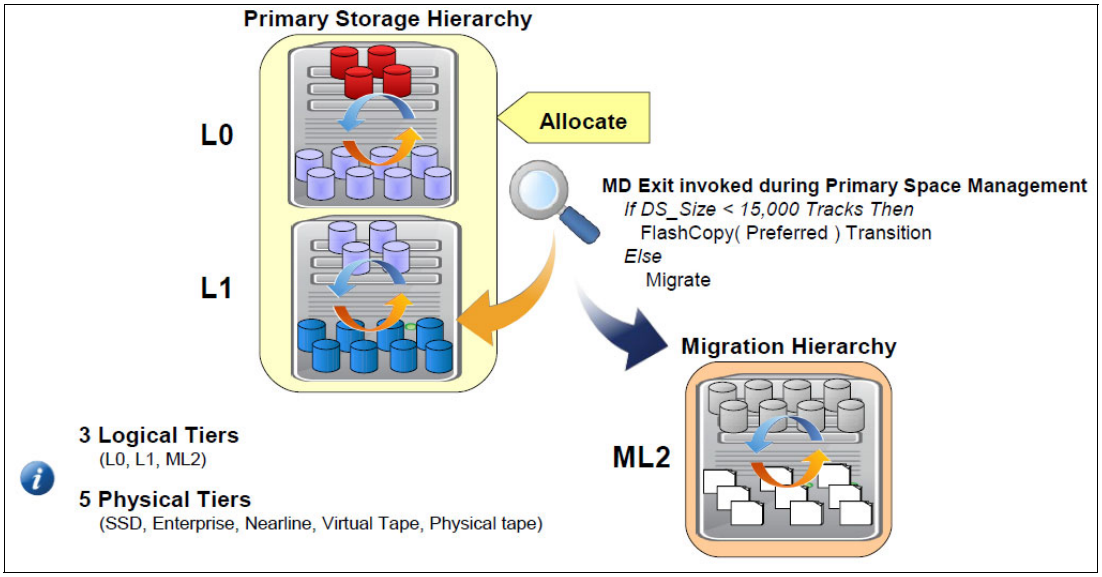


Figure 2-2 ARCMDEXT started during primary space management

2.4 Fast Replication Patch non-DFSMShsm backup utility

For TRANSITION and MOVE processing, all forms of Fast Replication can be used only if the change indicator is OFF and there is a valid HSM backup copy.

If you are using another backup utility in z/OS V2R2 and want DFSMShsm to perform a fast replication of your data, this new patch can be applied as shown in the following example:

```
PATCH .MGCB.+111 BITS(.... ..1.)
```

Note: If Preserve Mirror Required is specified, the change indicator can be ON.
In a GDPS environment, Preserve Mirror Required avoids unexpected FREEZEs during fast replication processing.

2.5 Fast replication messaging in a consolidated log

DFSMS fast replication is a function with which you can create a backup that is managed by DFSMShsm at volume-level with a minimum application outage if used with the FlashCopy feature from storage subsystem controllers.

By using DFSMS storage group type COPY POOL BACKUP, you can define volumes to be used as TARGET for backup versions of your sets of storage groups. Instead of defining one COPY POOL BACKUP for each storage group that you want to back up, you can use FRBACKUP DUMP or Automatic Dump to move data from the SG COPY POOL BACKUP to tape and make room for new copies.

Note: For more information about implementing COPY POOLs, see *DFSMShsm Fast Replication Technical Guide*, SG24-7069.

Until z/OS V2R1, DFSMShsm wrote messages that were related to a Fast Replication in a shared log, which means that if you are looking for a specific message about a determined Fast Replication process, you must find the correct log and then search through all of the messages for all HSM activity.

To eliminate the need to search through different logs, z/OS V2R2 added functionality to use a new **SETSYS** command to define Fast Replication messages that are stored in a particular data set.

The **SETSYS** command can be placed into DFSMShsm parmlib to be used the next time DFSMShsm is started. You can also issue the command from an ISPF command shell by MODIFYing DFSMShsm started task to dynamically activate this configuration, as shown in the following example:

```
SETSYS FASTREPLICATION(MESSAGEDATASET(YES|NO HLQ(hlq)))
```

Note: If you perform dynamic activation, ensure that you also have the command on DFSMShsm. Otherwise, DFSMShsm resets to its default after restart.

The high-level qualifier (HLQ) is the first HLQ to be used by Fast Replication messaging data set. The use of more than one qualifier on a HLQ parameter results in a command parse error.

DFSMShsm dynamically allocates message data sets with the following characteristics:

- Physical Sequential
- FBA
- LRECL of 121
- Block size of 1210

Message Data Sets must be SMS managed; therefore, SMS and ACS routines must be set up before allocation.

The data set that is created by DFSMShsm uses the following name:

```
msgdsprefix.FB.Ccccccc.ippppppp.Dyyddd.Thhmmss.z
```

This name includes the following components:

- `msgdsprefix` is the value of the HLQ parameter of the **SETSYS** command. If the HLQ parameter is not specified, the default prefix of `HSMMSG` is used.
- `ccccccc` is up to seven characters long to identify the copy pool that was processed.
  - For a non-DB2 copy pool, these characters are the first seven characters of the copy pool name.
  - For a DB2 copy pool, these characters are the first seven characters of the location qualifier in the DB2 copy pool name.
- `i` is an identifier that indicates whether the message data set contains messages for the following copy pools:
  - D: DB2 data
  - L: DB2 log
  - X: Non-DB2 copy pool data

- pppppppp is up to seven characters long and identifies the copy pool that was processed.
  - For a non-DB2 copy pool, these characters are the next seven characters of the copy pool name (the characters that follow ccccccc).
  - For a DB2 copy pool, these characters are the next seven characters of the location qualifier in the DB2 copy pool name.
- yyddd is the year and day when the FRBACKUP or copy pool auto dump operation started.
- hhmmss is the time in hours, minutes, and seconds when the FRBACKUP or copy pool auto dump operation started.
- z indicates the status of the FRBACKUP or copy pool auto dump operation and is set to one of the following values:
  - S: Success
  - F: Fail
  - I: In progress

After you set up the Fast Replication message data set, all Fast Replication messages that are written to the operator or log also are recorded. Duplicate messages are not removed.

To help you to identify the time when a message was issued and the DFSMShsm it is related to, all messages are prefixed with the following format:

```
yy/mm/dd hh:mm:ss Hx message-text
```

Where:

- yy/mm/dd hh:mm:ss represents the system date and time when the message was issued
- H is a constant
- x is the DFSMShsm host ID that issued the message

Two Fast Replication messages that were recorded from DFSMShsm host ID 2 are shown in Example 2-1.

Example 2-1 Messages written in Fast Replication message data set

```
14/11/21 10:09:10 H2 ARC1801I FAST REPLICATION BACKUP DUMP IS
STARTING FOR COPY POOL DSN$PAYROLLA123$DB, AT 10:09:10 ON 2014/11/21
14/11/21 10:09:11 H2 ARC1805I THE FOLLOWING 00003 VOLUME(S) WERE
SUCCESSFULLY PROCESSED BY FAST REPLICATION BACKUP OF COPY POOL
DSN$PAYROLLA123$DB
```

Note: Messages that indicate parse errors or termination before the ARC1801I or ARC1841I start messages are not collected in message data sets. This situation occurs because the message data set is allocated after the request is determined eligible for processing.

The new message ARC1870I is issued when a message data set is created, as shown in Example 2-2 on page 25.

Example 2-2 Message ARC1870I

```
ARC1870I MESSAGE DATA SET msgdsname CREATED FOR {FAST REPLICATION
{BACKUP | BACKUP DUMP | BACKUP DUMPONLY | PREPARE } | AUTO DUMP} OF {COPY
POOL cpname | VOLUME volser | DATA SET dsname}
```

You can use the **QUERY SETSYS** command to verify whether Fast Replication message data set creation is enabled in the environment. The message number that is related to the message data set is ARC1823I, as shown in Example 2-3.

Example 2-3 Message ARC1823I

```
ARC1823I MAXCOPYPOOL (FRBACKUP TASKS=0015, FRRECOV 093
ARC1823I (CONT.) TASKS=0015, DSS TASKS=0024),
ARC1823I (CONT.) FASTREPLICATION(DATASETRECOVERY=NONE
ARC1823I (CONT.) FCRELATION=EXTENT VOLUMEPAIRMESSAGES=NO
ARC1823I (CONT.) MESSAGEDATASET(NO HLQ=HSMMSG))
```

2.6 DFSMShsm common dump queue

DFSMShsm Dump is a function with which you can make copies of data at volume-level by using commands (BACKVOL or Fast Replication) or automatic functions. This function is part of the Availability Management procedures.

Before z/OS V2R2, DFSMShsm processed the request for dumps only in the host that started the request; however, the resources of a single DFSMShsm host might not be sufficient to handle the large copy pool sizes that must be processed.

Figure 2-3 shows Dump request and processing in a single DFSMShsm Host, even though this Host is part of HSMPlex.

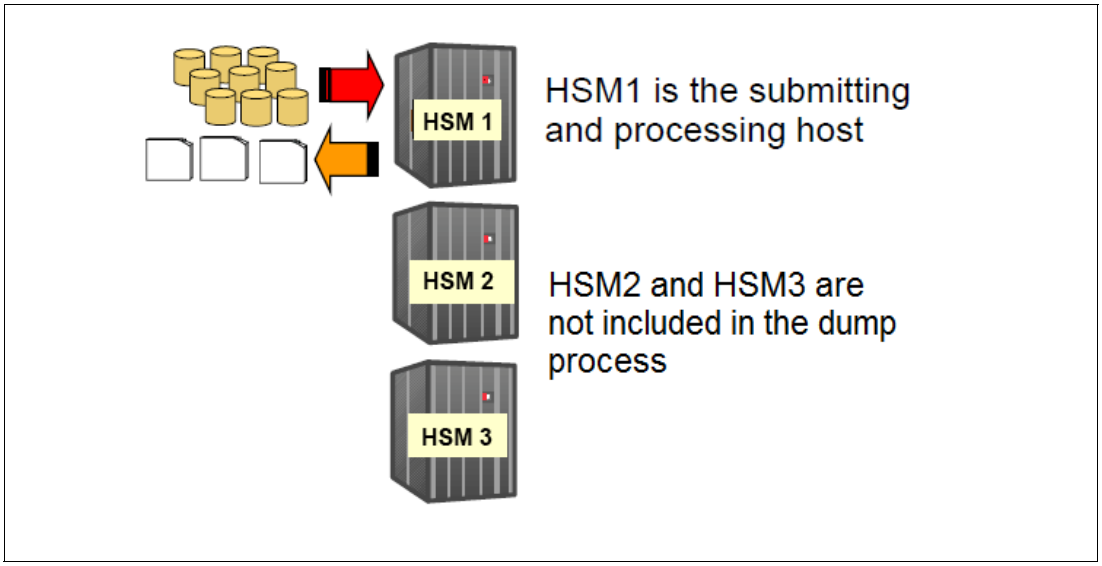


Figure 2-3 DFSMShsm Dump single host processing

The DFSMShsm in z/OS V2R2 introduces an architecture to distribute workloads across host members in the HSMPlex. Dumps now can be routed for processing in other DFSMShsm hosts, and split COPY POOL Dump requests among several DFSMShsm hosts.

This common dump queue provides a better flexibility in performing Dump functions. It also provides better storage resource usage and reduces overall processing times.

Unlike the common recall queue that implements a persistent CF list structure, common dump queue uses XCF messaging services (a function within XCF) to receive, organize, and distribute the Dump requests for DFSMShsm hosts.

To control Dump processing, a DFSMShsm host must be defined as Master Scheduler (MA). This host is responsible for receiving Dump requests from all DFSMShsm hosts within the Common Dump Queue group, and distributes the workload to eligible hosts. Figure 2-4 shows an example of how DFSMShsm Common Dump Queue works.

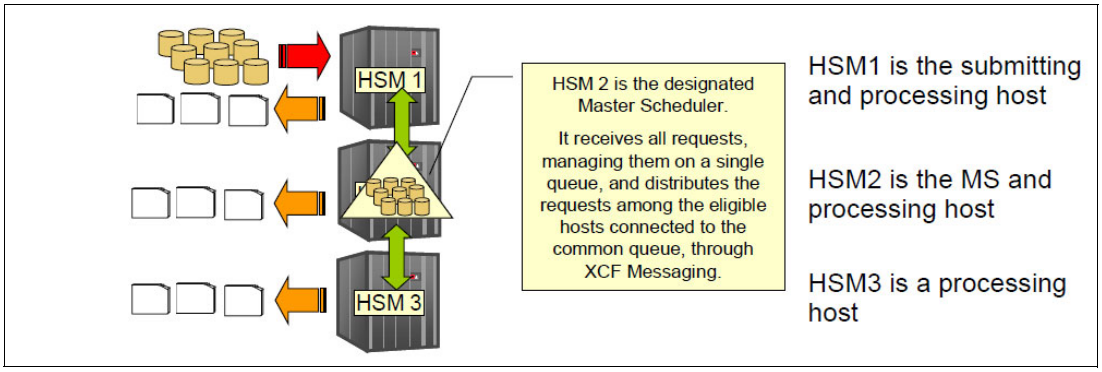


Figure 2-4 DFSMShsm Common Dump Queue

In the implementation of Common Dump Queue, the host can assume the following roles:

- Master Scheduler

  The Master Scheduler is the host that is responsible for receiving, managing, and distributing Dump requests within the XCF group. There can be only one Master Scheduler at a specific time within the group.

  Any DFSMShsm host is a Master Scheduler candidate if it has SETSYS CQ (DUMP(MSC(Y)) setup. Because Y is the default option for this parameter, *only the first DFSMShsm host to connect to the XCF group is assigned as Master Scheduler*. All other hosts are considered to be a candidate and assume if the current Master Scheduler fails or is shut down.

  The MS features the following responsibilities:

  - Accepting requests from submitting host members and locally submitted requests and maintaining them as a single common queue of all requests for that grouped function
  - Submitting Dump requests
  - Serving as the submitting host; if so, the command is not sent
  - Distributing the requests among the available tasks across all of the eligible hosts (processing hosts) that are connected to the common queue
  - Managing the work completed messages that are received from a processing host to support function, such as STACKING
  - Sending command complete message back to submitting host

- Submitting Host

  Any hosts that are connected to the Common Dump Queue are considered as submitting hosts. The MS host can also be a submitting host. In this scenario, there is no need to route the request to MS, only process and direct it to an available host.

The Submitting Host features the following responsibilities:

- Receiving requests from USERs or Automatic Functions and send to Master Scheduler by XCF messaging
- Processing command complete message from Master Scheduler and posts user complete

- Processing Host

  To become a Processing host, it is necessary to be connected to the Common Dump Queue with DUMP function not set to HOLD and with DUMPTASKS greater than 0. MS and Submitting Hosts can also be a Processing Host.

  The Processing host receives requests from the MS and process and sends a response about completion to the MS.

The architecture of Common Dump Queue allows the following flexible configurations:

- Multiple queues in the same HSMplex; same and different functions:
  - Hosts that submit and process requests from the group
  - Hosts that only process requests
  - Hosts that submit requests, do but not process them
- No system affinity
- Supports priority

Note: In addition to the flexible configuration, a DFSMShsm host can connect to only one Common Dump Queue.

To use Common Dump Queue, DFSMShsm creates an XCF group that is named ARCD with four character suffix `basename`, as shown in the following example:

XCF Group name: `ARCDbasename`

For example, if you define your basename as QUE1, the Dump Group name is ARCDQUE1. To connect to this XCF Group, the **`SETSYS COMMONQUEUE(DUMP(CONNECT(QUE1)))`** command must be issued.

Each DFSMShsm host is identified on an XCF group by `HOST#hsmid`, where `hsmid` is the host name.

For more information about commands to display XCF groups, Common Dump Queue, and DFSMShsm information, see “Commands” on page 31.

2.6.1 Sample implementation

In our implementation scenario, we consider an HSMplex with three DFSMShsm hosts (A, B, and C). This example covers Common Dump Queue implementation only.

Before the Common Dump Queue is created, you must define a system as MS. You can define your MS based on system usage, resources, availability, and other factors that are unique to your configuration. Avoid having only one host eligible for MS; it is suggested you have at least one other eligible host to take over MS responsibilities if there is a planned outage.

In this example, host A is used as MS, while host B is an MS submitting and processing host, and host C is a submitting and processing host.

Issue the **SETSYS CQ(DUMP(CON(QUE1)))** command on the DFSMShsm host that you want to be the MS to start the connection and create an XCF Group. Because we decided to use host A as MS, we issue the command on host A first.

We suggest having more than one host that is defined as the MS candidate. In this example, DFSMShsm host B was also defined as an MS candidate.

Issuing the **SETSYS** command on host A gave us the response that is shown in Example 2-4.

Example 2-4 Response from host A

```
ARC1566I CONNECTION TO THE DUMP GROUP ARCDQUE1 WAS
ARC1566I (CONT.) SUCCESSFUL
ARC1569I HOST=A HAS TAKEN OVER THE MASTER SCHEDULER
ARC1569I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1569I (CONT.) ARCDQUE1
```

Issuing the **SETSYS** command on host B gave us the response that is shown in Example 2-5.

Example 2-5 Response from host B

```
ARC1566I CONNECTION TO THE DUMP GROUP ARCDQUE1 WAS
ARC1566I (CONT.) SUCCESSFUL
```

Next, we connected host C to Common Dump Queue without the MS option, as shown in the following example:

```
SETSYS CQ(DUMP(CON(QUE1) MSC(N)))
```

Keyword MSC in the **SETSYS** command was omitted for hosts A and B. In this case, the default Y is used. When host A connect to XCF group was issued, it created ARCDQUE1 group on XCF. It was also selected as MS, as described in message ARC1569I.

If host A is disconnected from the group, host B takes over the MS. To reestablish host A as MS, connect it to the group again. For more information, see “Changing the MS” on page 29.

Note: It is not required that the first DFSMShsm host to connect the group is an MS candidate. In this case, all Dump work continues being processed locally until an MS-capable host joins the group.

If you use the same ARCCMDxx for all DFSMShsm in a HSMPlex, you can use the **ONLYIF** command to define each host role.

In addition, MS host communicates with processing hosts and takes new STACKing parameters to balance workload. For more information about MINSTACK and MAXSTACK, see 2.7, “Dump class definitions MINSTACK and MAXSTACK” on page 33.

2.6.2 Disconnecting from a Common Dump Queue

Before shutting down DFSMShsm, it is suggested to disconnect it from the Common Dump Queue to prevent receiving Dump requests. If it is the Master Scheduler, redirect the MS role to another eligible host.

The preferred way for disconnecting from Common Dump Queue is issuing the following regular disconnect command:

```
SETSYS COMMONQUEUE(DUMP(DISCONNECT))
```

Running the regular disconnect command waits for all submitted and received work to be completed for this host before disconnecting from the queue. For MS host, all MS assigned work also is completed before disconnecting from the group. In this case, a new MS host is assigned. It might take some time for DFSMShsm host to complete the current Dump requests; therefore, consider providing enough time before forcing disconnect or shutdown.

A HOLD DUMP that is issued before the disconnect command prevents queued requests from being selected. After the disconnect completes, a RELEASE DUMP allows the queued requests to be processed locally. This process might speed up the disconnect processing of an MS host.

Example 2-6 shows the response messages from a disconnect command from an MS host.

Example 2-6 Host A Response to DISCONNECT command

```
ARC1568I HOST=A HAS LOST ITS MASTER SCHEDULER
ARC1568I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1568I (CONT.) ARCDQUE1
ARC1567I DISCONNECTION FROM THE DUMP GROUP ARCDQUE1
ARC1567I (CONT.) WAS SUCCESSFUL
```

For non-MS hosts, only ARC1567I messages are issued.

As a last resort, it is also possible to use a **FORCE** command to force a DFSMShsm to be disconnected from the queue. In this case, all submitted work from this host fails, but might still complete in the common queue. If the host was the MS, this role is assigned to another candidate MS host after this host disconnects. If there are no eligible hosts, local processing of work takes over. The **FORCE** command is shown in the following example:

```
SETSYS COMMONQUEUE(DUMP(DISCONNECT FORCE))
```

The same ARC1568I and ARC1567I is displayed for MS host and ARC1567I for other hosts.

2.6.3 Changing the MS

During host maintenance, initial program loads (IPLs), or other planed outages, you might want to switch MS responsibilities to another host so that the Common Dump Queue can continue processing requests while the original MS host is unavailable for processing.

Use the following command to remove MS responsibilities from a specific host:

```
SETSYS COMMONQUEUE(DUMP(MSC(N))
```

In Example 2-7, the COMMONDUMPQUEUE name is ARCDQUE1 and two hosts can be an MS (host A has this role).

The response to issuing the command with option **MSC(N)** to DFSMShsm host A is shown in Example 2-7 on page 30.

Example 2-7 Host A Output

```
ARC1568I HOST=A HAS LOST ITS MASTER SCHEDULER
ARC1568I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1568I (CONT.) ARCDQUE1
```

After host A loses MS responsibilities, DFSMShsm host B takes over. You can identify this change by message ARC1569I, as shown in Example 2-8.

Example 2-8 Host B Output

```
ARC1569I HOST=B HAS TAKEN OVER THE MASTER SCHEDULER
ARC1569I (CONT.) RESPONSIBILITIES FOR THE COMMON DUMP QUEUE GROUP
ARC1569I (CONT.) ARCDQUE1
```

2.6.4 Shutdown considerations

When the DFSMShsm host is part of Common Dump Queue, you should disconnect the host from Common Dump Queue before shutting down.

Disconnecting the host orderly quiesces MS responsibilities on this host to reassign them to an MS candidate. If the command is not issued, the shutdown fails queued requests and waits for processed requests to complete before shutting down. For more information about disconnecting a DFSMShsm host from Common Dump Queue, see 2.6.2, “Disconnecting from a Common Dump Queue” on page 28.

2.6.5 MS loss of connectivity considerations

If the MS looses connectivity with other hosts, all of the assigned work is completed. A message also is sent that indicates it cannot complete any more work.

The submitting hosts rescind submitted requests. If there is another MS host available for processing, the requests are resubmitted to that host. Otherwise, Dump tasks are reset to become local.

2.6.6 Holding, releasing, and canceling Dump considerations

When Common Dump Queue is used, issuing any **HOLD** command to hold all Dump, Auto, or FRBACKUP Dumps causes the host to be ineligible to process requests. It also prevents MS duties from being performed if issued on MS host.

The **HOLD** command is best reserved for stopping active dump processing or scheduling because it fails any submitted Dump that is in the WAIT queue.

To configure a Common Dump Queue host that is not selected for dump processing, it is preferable to use **SETSYS MAXDUMPTASKS(0)** because a host can still submit DUMP requests to Common Dump Queue. Requests are processed in other available hosts in the Common Dump Queue. If the host is the MS, this process does not affect MS responsibilities.

Issuing the **RELEASE DUMP** command releases MS duties for the MS host and enables hosts for processing if their MAXDUMPTASKS(x) is greater than zero.

The **CANCEL** command must be issued from the system that originated the request. To cancel active requests with the TCBADDRESS, you must issue **CANCEL** on the host that is processing the request.

2.6.7 Altering Dump priority

The **ALTERPRI** command can be used to change the priority of the requests that are waiting on Common Dump Queue to be run. The **ALTERPRI REQUEST** or **USERID** commands change the priority of queued requests in the Common Dump Queue for requests that originated in the host that issued the **ALTERPRI** command that also matches the **REQUEST** or **USERID** filter criteria.

2.6.8 Commands

This section describes some sample commands that can be used to gather DFSMShsm, XCF, and Common Dump Queue information.

Use the following display commands to check DFSMShsm hosts that are connected to XCF Group:

- Identify the XCF Group, as shown in Example 2-9.

Example 2-9 Display XCF,GROUP Output

```
D XCF,GROUP
IXC331I 08.21.32 DISPLAY XCF
GROUPS(SIZE): ARCDQUE1(2)
```

Note: The group that is related to Common Queue Dump always start with ARCD.

- Identify the DFSMShsm hosts that are connected, as shown in Example 2-10.

Example 2-10 Display XCF,GROUP,ARCDQUE1 Output

```
D XCF,GROUP,ARCDQUE1
IXC332I 08.22.00 DISPLAY XCF
GROUP ARCDQUE1: HOST#A
HOST#B
```

- Display SYSTEM, JOBID, and STATUS information of each member that is connected, as shown in Example 2-11.

Example 2-11 Display XCF,GROUP,ARCDQUE1,ALL Output

```
D XCF,GROUP,ARCDQUE1,ALL
IXC333I 08.22.12 DISPLAY XCF
MEMBER NAME:     SYSTEM:     JOB ID:    STATUS:
HOST#A           SYSTEM1     DFHSM      ACTIVE
HOST#B           SYSTEM1     DFHSM      ACTIVE
INFO FOR GROUP ARCDQUE1 MEMBER HOST#A ON SYSTEM SYSTEM1
FUNCTION: HSM DUMP XCF GROUP
MEMTOKEN: 01000003 000C0001 ASID: 0038 SYSID: 01000001
```

- Display the current Common Dump Queue settings on a DFSMShsm host by using the DFSMShsm **QUERY SETSYS** command and identifying message ARC1500I, as shown in Example 2-12 on page 32.

Example 2-12 QUERY SETSYS Output

```
ARC1500I PLEXNAME=ARCPLEX0,PROMOTE PRIMARYHOST=NO,
PROMOTE SSM=NO,COMMON RECALL QUEUE BASE NAME=*****,
COMMON RECALL QUEUE TAPEDATASETORDER=*****,
COMMON DUMP QUEUE BASE NAME=QUE1 AND MSC=***
```

Note: MSC has three possible values: ***, YES, and NO.*** means that the Master Schedule Candidate setting was not specified, but the default for it allows it to be eligible.

- The DFSMShsm command **QUERY REQUEST** display requests that originated from the host that is issuing the query commands that match the **REQUEST** or **USER** filter criteria and have different behaviors, depending on the role of DFSMShsm Host in Common Dump Queue.
  - For a Submitting or Processing Host that is connected to a Common Dump Queue:
    - The local tasks and queue are searched and the MS member's group queue is searched.
    - XCF messaging is used to send the command and receive responses to be printed on the QUERY host.
  - For a Master Scheduler Host, the local tasks and queue are searched.

  In Example 2-13, how the messages ARC0161I and ARC0167I were enhanced to support this display is shown.

Example 2-13 QUERY REQUEST Output

```
ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=B
ARC0101I QUERY REQUEST COMMAND STARTING ON HOST=A
ARC0101I QUERY REQUEST COMMAND COMPLETED ON HOST=A
ARC0161I DUMPING VOLUME PRIM01 FOR USER **OPER**,
ARC0161I (CONT.) REQUEST 00000037 ON HOST=A
ARC0167I DUMP MWE FOR VOLUME PRIM03 FOR USER **OPER**,
ARC0167I (CONT.) REQUEST 00000037, WAITING TO BE PROCESSED,00000
ARC0167I (CONT.) MWE(S) AHEAD OF THIS ONE
ARC0161I DUMPING VOLUME PRIM02 FOR USER **OPER**,
ARC0161I (CONT.) REQUEST 00000037 ON HOST=B
ARC0101I QUERY REQUEST COMMAND COMPLETED ON HOST=B
```

- Display the current status on DFSMShsm host by using the **QUERY ACTIVE** command. A new message ARC1560I is reported to display the current state of the Common Dump Queue and active requests that are running it. The state and active requests that are reported on are those requests that originated from the host that is issuing the query command, as shown in Example 2-14.

Example 2-14 QUERY ACTIVE Output

```
ARC1560I COMMON DUMP QUEUE FACTORS: GROUPNAME=ARCDQUE1,
ARC1560I (CONT.) CONNECTION STATUS=CONNECTED, MASTERSCHEDULER
ARC1560I (CONT.) HOSTID=A, MASTER SCHEDULER STATUS=ENABLED
```

- Issuing the **QUERY WAITING** command displays the number of requests that are queued in the Common Dump Queue that originated from the host that is issuing the query command and has different behaviors, depending on the role of DFSMShsm Host in Common Dump Queue.
  - For a Submitting or a Processing Host that is connected to a Common Dump Queue, the local queue is searched and the MS member's group queue is searched.

– For a Master Schedule Host, the local queue is searched.

Messages ARC0168I and ARC1542I were enhanced to support this display, as shown in Example 2-15.

Example 2-15 QUERY WAITING Output

```
ARC0101I QUERY WAITING COMMAND STARTING ON HOST=A
ARC1542I WAITING MWES ON COMMON QUEUES: COMMON RECALL QUEUE=00000000,COMMON
ARC1542I (CONT.) DUMP QUEUE=00000002,TOTAL=00000002
```

- Display Common Queue that uses new option DUMP displays new message ARC1562I for active requests and new ARC1563I for requests that are waiting to be processed, as shown in Example 2-16 (the **QUERY COMMONQUEUE(DUMP)** command is issued).

Example 2-16 QUERY COMMONQUEUE(DUMP) Output

```
ARC1562I VOLUME BH5T03 DUMPING FOR USER KWRES08, REQUEST 00000092 ON HOST=A,
ARC1562I (CONT.) BEING PROCESSED FROM A COMMON QUEUE, GROUP NAME=ARCDQUE1
ARC1563I VOLUME BH5T04 DUMPING FOR USER KWRES08, REQUEST 00000086 FROM HOST=A,
ARC1563I (CONT.) WAITING TO BE PROCESSED ON A COMMON QUEUE, 00000 MWES AHEAD OF
ARC1563I (CONT.) THIS ONE, GROUP NAME=ARCDQUE1
```

- Use **CANCEL REQUEST** or **USERID** commands cancel queued requests in the Common Dump Queue for requests that originated in the host that issued the **CANCEL** command that also matches the **REQUEST** or **USERID** filter criteria, as shown in Example 2-17.

Example 2-17 CANCEL REQUEST Output

```
ARC1008I VOL=BH5T03 BACKVOL REQUEST 00000089 WAS CANCELLED
ARC0931I (H)CANCEL COMMAND COMPLETED, NUMBER OF REQUESTS CANCELLED=1
COMMAND REQUEST 00000091 SENT TO DFSMSHSM
```

- The **ALTERPRI REQUEST** or **USERID** commands change the priority of queued requests in the Common Dump Queue for requests that originated in the host that issued the **ALTERPRI** command that also matches the **REQUEST** or **USERID** filter criteria, as shown in Example 2-18.

Example 2-18 ALTERPRI REQUEST command output

```
ARC0980I ALTERPRI REQUEST COMMAND STARTING ON HOST=A
ARC0982I DUMP MWE FOR VOLUME BH5T03 FOR USER KWRES08,  329
ARC0982I (CONT.) REQUEST 00000092, REPRIORITIZED TO HIGH
ARC0981I ALTERPRI REQUEST COMMAND COMPLETED ON HOST=A,  330
ARC0981I (CONT.) RC=0000
```

2.7 Dump class definitions MINSTACK and MAXSTACK

Before z/OS V2R2, the **STACK** value in **DEFINE DUMPCLASS** took precedence over the number of available tasks.

In a scenario of an environment that is set up with Dump STACK(100), if there are 100 volumes and 10 available dump tasks, all 100 volumes are dumped by using a single task to the same tape.

This configuration reduces the throughput and causes an increase in the execution window.

Starting with z/OS V2R2, it is now possible to indicate that the use of the available dump tasks is more wanted than optimizing stacking. This process can be achieved by running the **MINSTACK** command, which defines the minimal number of dumps that are stacked in a single tape.

In addition, **MAXSTACK** was created as an alias of **MINSTACK** parameter.

To use this new feature, you must change your Dump Class definition. The new DUMPCLASS syntax is shown in Example 2-19.

Example 2-19 DEFINE DUMPCLASS syntax

```
DEFINE DUMPCLASS(DCLASS1 MINSTACK(minv) MAXSTACK(maxv))
```

As shown in Example 2-19, `minv` represents the preferred minimum number of dump copies that the system must use to place on a tape volume. Although this setting causes DFSMShsm to use more dump tasks, there can be fewer dump copies on tape.

Also shown in Example 2-19, `maxv` represents the preferred maximum number of dump copies that the system must use to place on a tape volume.

To use parameters, the MAXSTACK value must be equal to or larger than the MINSTACK. If you do not specify MINSTACK, MAXSTACK value is used.

Note: It is expected to reduce the processing window because of parallelism; however, it uses more tapes and tape drives simultaneously.

2.8 Multiple copy pools onto the same tape

In previous versions of z/OS, every copy pool was assigned to its own tapes and any remaining free space on the tape was wasted.

In z/OS V2R2, different copy pools can share tapes if the pools the same Dump Class and start at same time by the Automatic Dump process. This process occurs automatically.

If you do not want different copy pools on the same tape, consider changing the DUMP CLASS in STORAGE GROUP.

2.9 New UPDTCDS command

DFSMShsm features a new command **UPDTCDS** that can be used to simplify expiration date update for a copy pool dump copy.

Instead of issuing several **FIXCDS** commands to perform expiration date updates (which can lead to the wrong records being inadvertently updated and take some time to complete), you can now use a single command to perform expiration date updates, as shown in Example 2-20.

Example 2-20 UPDTCDS command

```
UPDTCDS COPYPOOL(name) VERSION(ver) DUMPEXPIRATION(DCLASS(name) NEWDATE(date))
```

This command updates all of the CDS records that must be updated to reflect the new expiration date. The Fast Replication Dump record and dump record for each volume dumped is updated.

3

DFSMSrmm

In this chapter, we describe the following enhancements that are now available in z/OS V2R2 for DFSMSrmm:

- **WHILECATALOG** attribute and new possibilities of EXPDT retention method
- Improvements to **CHANGEVOLUME**, **CHANGEDATASET, SEARCHVOLUME**, **SEARCHDATASET** subcommands and to UXTABLE

This chapter includes the following topics:

- 3.1, “DFMSMSrmm overview” on page 38
- 3.2, “WHILECATALOG attribute and EXPDT retention method” on page 38
- 3.3, “RAS DFSMSrmm enhancements” on page 41

3.1 DFMSMSrmm overview

Removable Media Manager (RMM) is a component of the DFSMS family and a z/OS feature. DFSMSrmm manages your installation's tape volumes and the data sets on those volumes. With DFSMSrmm, you can manage automated, virtual, and manual tape libraries whether they are on-site or off-site (vaults). Shelves where volumes are stored (except for automated libraries) also can be managed by DFSMSrmm.

Different DFSMSrmm subsystems that share the control data set make up a RMMplex.

3.2 WHILECATALOG attribute and EXPDT retention method

The data set expiration can be specified in JCL through the EXPDT=nnnnn parameter. The use of only the expiration date is a simple way to control how long this data set is available. Although it is faster to use only the expiration date for retention purposes, more complicated scenarios require the use of a set of VRS policies, which are used to set the retention date of a volume during regular housekeeping jobs.

Until V2R2, it was impossible to allow the catalog status of a data set control its retention without the use of a VRS policy. Now, the user can directly specify WHILECATALOG for data sets that are managed by their expiration date.

DFSMSrmm PARMLIB member was introduced to assist you in creating retention rules that are based on catalog activity along with an expiration date. The value that is defined on PARMLIB is used as default for the **WHILECATALOG** setting for new tape data sets.

The new parameter **WHILECATALOG** has the following settings that can be adjusted for use in your environment. This attribute is applicable only to data sets that are managed by the EXPDT retention method:

- **WHILECATALOG(OFF)**

  This value is the default value. Tape data sets expire on their expiration date, regardless of their catalog status.

  WHILECATALOG(UntilExpired)

  Data sets expire on their expiration date if they are cataloged. If they are not cataloged, their expiration date is decreased, as defined by the CATLGDAYS PARMLIIB option, which can make them eligible for expiration.

- **WHILECATALOG(ON)**

  Data sets do not expire if they are cataloged. If they are not cataloged, they expire on their expiration date.

Note: VRS has its own WHILE CATALOGED attribute, which is not related to the attribute that is described in this book. For more information about VRS processing, see *DFSMSrmm Implementation and Customization*.

The **WHILECATALOG** attribute can be set in PARMLIB or can be changed by using the **CHANGEDATASET** DFSMSrmm subcommand. You can have different default values for GDG and non-GDG data sets. GDG data sets might be uncataloged automatically as they are rolled off when newer generations of the data sets are written.

To use the new **WHILECATALOG** attribute, you must define it as a default in the EXPDT parameter of the DFSMSrmm PARMLIB. Example 3-1 shows a sample DFSMSrmm PARMLIB with WHILECATALOG(ON) that is set up for GDG and non-GDG data sets.

Example 3-1 Sample DFSMSrmm PARMLIB with WHILECATALOG(ON)

```
OPTION  DSNAME(RMM.CONTROL.DSET)    -
        JRNLNAME(RMM.JOURNAL.DSET) -
        CDSID(WTSCPLX1)             -
        RETPD(0)                    -
        RETENTIONMETHOD(EXPDT(GDG(WHILECATALOG(ON)) -
        NONGDG(WHILECATALOG(ON)))) -
        TVEXTPURGE(EXPIRE(30))      -
        CATRETPD(0)                 -
        UNCATALOG(Y) -
        OPMODE(W)
```

Figure 3-1 shows a sample expiration processing for VOLUME 1 when **WHILECATALOG(ON)** is set to all data sets on volume. Though the current EXPDT value passed, the data sets are not expired until they are removed from catalog.

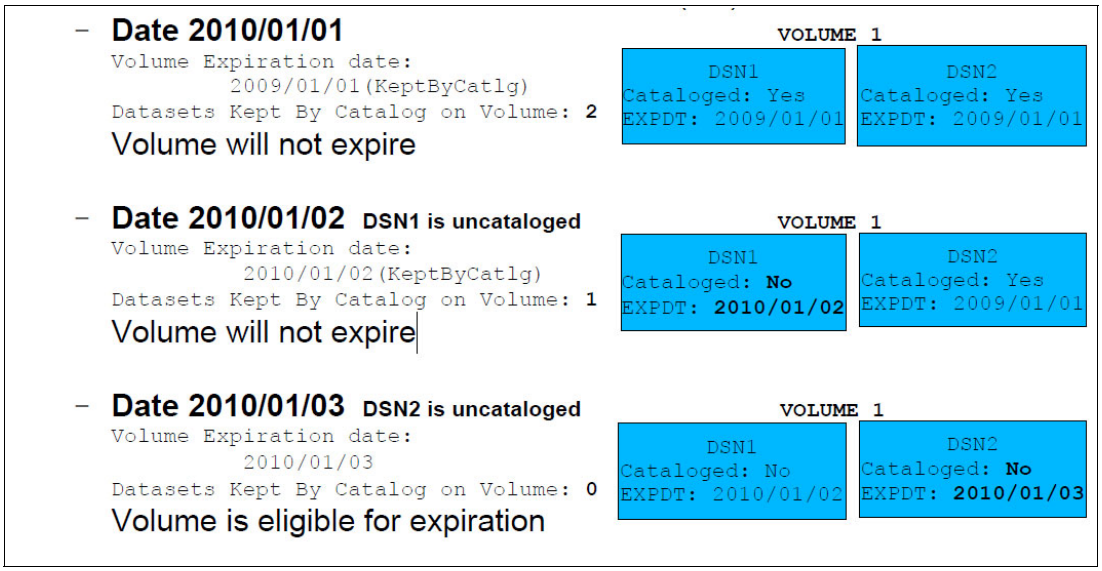


Figure 3-1 WHILECATALOG(ON) and expiration

Similarly, Figure 3-2 on page 40 shows expiration processing for VOLUME 1 when **WHILECATALOG(UntilExpired)** is set to all data sets on volume. In this example, when DSN1 was uncataloged, the EXPDT was updated to reflect the current date. The tape was not expired because DSN2 was still cataloged; therefore, it kept the tape PRIVATE.

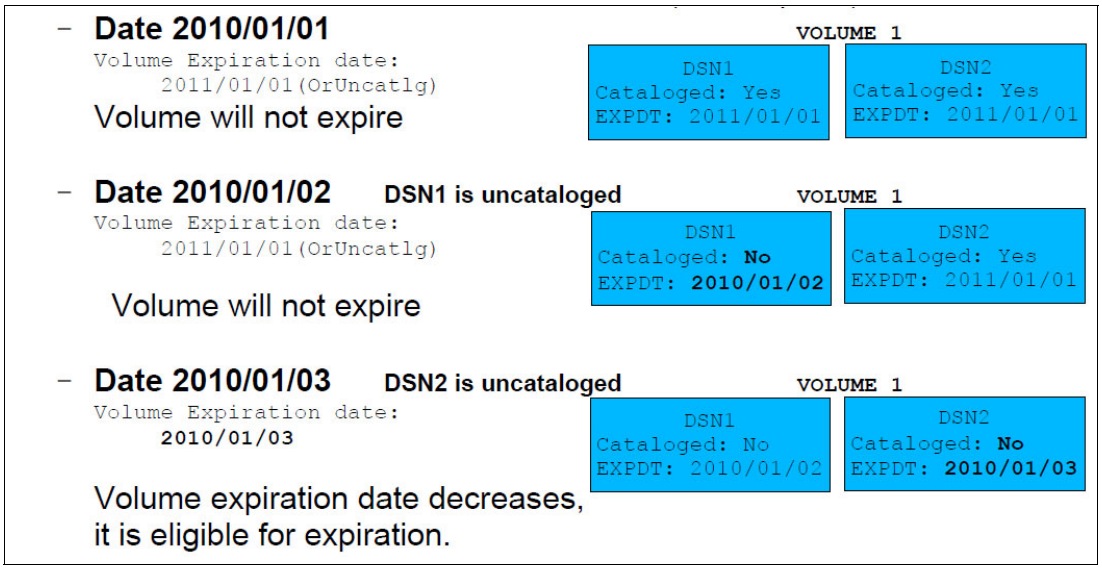


Figure 3-2 WHILECATALOG(UntilExpired) and expiration

For **WHILECATALOG(OFF)**, the processing continues to work as on previous releases.

The default for WHILECATALOG can be defined in PARMLIB member and different defaults can be specified for GDG and for non-GDG data sets.

To use this function in RMMplex with different z/OS versions, see toleration research the PSP buckets for the latest information.

If a volume is kept by catalog (because of a data set with WHILECATALOG=ON), it is not expired, even if EXPROC is run on lower-level systems in the RMMplex. A data set that had WHILECATALOG set to ON or UNTILEXPIRED is uncataloged on a lower system. Its expiration date changes the same way as it changes in z/OS V2R2 by using the default CATLG_DAYS value of two days.

All data sets that were created before z/OS V2R2 have the WHILECATALOG attribute set to OFF. If you intend to change the assigned WHILECATALOG to any data sets, you can use the **CHANGEDATASET** command to update DFSMSrmm control records.

Use the command that is shown in Example 3-2 to change data set attributes via batch or TSO command.

Example 3-2 Updating WHILECATALOG data set attribute

```
RMM CD 'RMMUSER.Z22CAT10.DSN0' VOLUME(A11570) WHILECATALOG(ON)
RMM CD 'RMMUSER.Z22CAT10.DSN1' VOLUME(A11570) WHILECATALOG(OFF)
RMM CD 'RMMUSER.Z22CAT10.DSN2' VOLUME(A11570) WHILECATALOG(UNTILEXPIRED)
```

You can also use **SEARCHDATASET** command to create a list of data sets that you want to update. Example 3-3 shows a simple SEARCHDATASET that results in a list of **CHANGEDATASET** commands to update WHILECATALOG attribute.

Example 3-3 SEARCHDATASET

```
RMM SEARCHDATASET DSNAME(*) VOLUME(vol_ser) LIMIT(*) -
CLIST('RMM CD ','VOLUME(vol_ser) WHILEDATALOG(OFF)')
```

3.3 RAS DFSMSrmm enhancements

The DFSMSrmm subcommands **SEARCHVOLUME**, **SEARCHDATASET**, **CHANGEVOLUME**, and **CHANGEDATASET** were enhanced to provide extra flexibility for searching and changing data into DFSMSrmm.

Also, UXTABLE and expiration processing were updated to provide better tape retention management.

3.3.1 UXTABLE RETPD change

Starting with z/OS V2R2, users can code a permanent retention period for UXTABLE.

To use this function, code **RETPD=PERM** in your UXTABLE.

3.3.2 SEARCHVOLUME and SEARCHDATASET

In addition to the current creation date start and end that can be used when searching for data sets or volumes, it is now possible to include start and end times for your search.

This specification provides extended flexibility for performing searches for updating data sets and volumes. To use this time specification, add a comma and time in the format `hhmmss` after the date. Example 3-4 shows a sample **SEARCHDATASET** command to display data sets that were created between 2013/110 at 22:00:00 and 2013/111 at 04:00:00.

Example 3-4 SEARCHDATASET including creation time

```
RMM SEARCHDATASET OWNER(*) LIMIT(*)
CRDATE(START(2013/110,220000) END(2013/111,040000))
```

3.3.3 CHANGEVOLUME and CHANGEDATASET

The **CHANGEVOLUME** command was enhanced to provide a better performance when it is used to change only EXPDT attribute from the records. This enhancement is applied by default, and no user action is necessary.

Because z/OS V2R2 also uses expiration time to determine whether the tape is eligible for scratch processing, **CHANGEVOLUME** and **CHANGEDATASET** commands can be used to update expiration time of volumes and data sets.

The command that is shown in Example 3-5 uses the **CHANGEDATASET** command to change the data set expiration date to 15:30:00.

Example 3-5 Updating expiration time of data sets

```
RMM CHANGEDATASET 'USER1.DUMP' VOLUME(THM000) EXPTM(153000)
```

3.3.4 Use of expiration time

In a batch environment, it is common that tapes are used during the night. Depending on the time a tape is created, the retention period for the tape can be shorter. If your job created a tape data set on 2015/166 at 23:30:00 with a retention period of one day and your housekeeping job runs shortly after midnight, your data set expires a few hours after its creation.

Beginning with z/OS V2R2, scratch processing also includes an expiration time when you are deciding whether to expire a tape. This setting prevents tapes from being scratched before the required time.

4

DFSORT

In this chapter, we describe the enhancements that are now available in z/OS V2R2 for DFSORT.

This chapter includes the following topics:

- 4.1, “DFSORT overview” on page 44
- 4.2, “Functional enhancements” on page 44
- 4.3, “Use of zHPF” on page 46
- 4.4, “Message ICE099A update” on page 47
- 4.5, “Joinkey indicators in SMF data” on page 47

4.1 DFSORT overview

DFSORT is a program that is used to sort, merge, and copy data. Consider the following points:

- When you sort records, you arrange them in a particular sequence and choose an order to make the data more useful.
- When you merge records, you combine the contents of two or more sorted data sets into one record.
- When you copy records, you make a duplicate of each record in your data set.

4.2 Functional enhancements

DFSORT provides several functions for date formatting. z/OS V2R2 introduced the AGE and WEEKNUM functions to help extract the information that you need in a simple way.

4.2.1 AGE function

z/OS V2R2 helps you calculate the date duration between an input date and current date by using the AGE function. You can select the AGE output from the following options:

- AGE=YMD

  Produces an 8-byte result that features a duration in years (0 - 9999), months (00 - 12), and days (00 - 31).

- AGE=YM

  Produces a 6-byte result that features a duration in years (0 - 9999) and months (00 - 12).

- AGE=YD

  Produces a 7-byte result that features duration in years (0 - 9999) and days (00 - 366).

The date conversion function AGE can be used with DFSORT INREC, OUTREC, and OUTFIL statements for BUILD and OVERLAY operands.

Figure 4-1 shows the input data, along with DFSORT control cards that uses AGE=YMD, and the resultant output if the current date in the scenario is 05/06/2015.

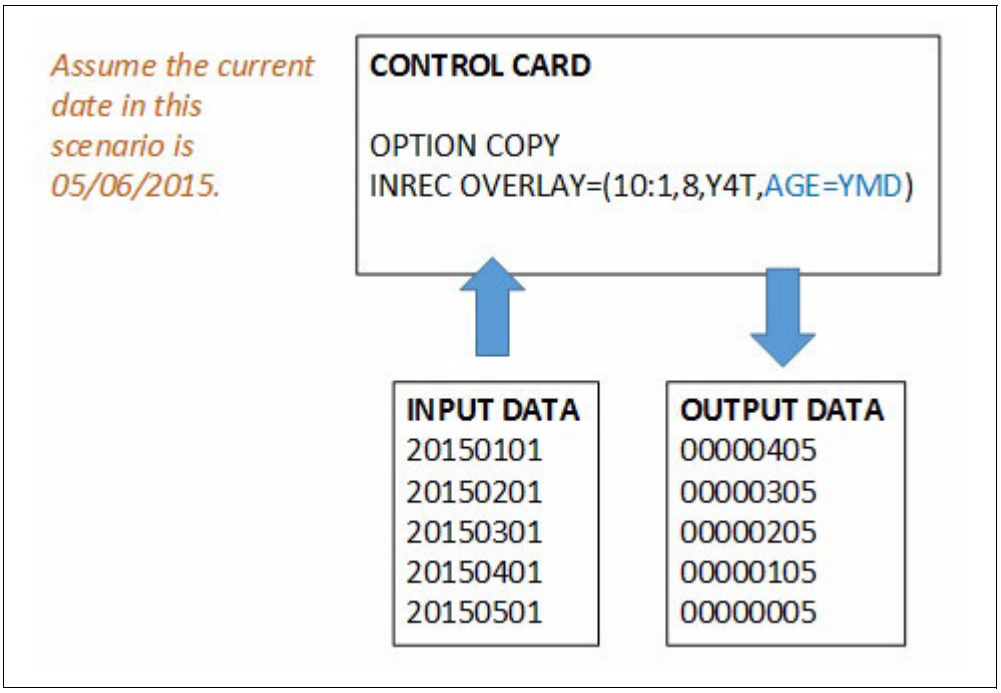


Figure 4-1 AGE=YMD

4.2.2 WEEKNUM function

Another new function in z/OS V2R2 is WEEKNUM. This function converts a specific Julian or Gregorian date to the respective week of the year.

There are two versions of this function: standard USA format and ISO format. Consider the following points:

- WEEKNUM=USA:

  This format returns an integer 1 - 54 that represents the week of the year (considering the week starts on Sunday).

- WEEKNUM=ISO

  This format returns an integer 1 - 53 that represents the week of the year (considering the week starts on Monday).

As with the AGE function, the week conversion function WEEKNUM can be used with DFSORT INREC, OUTREC, and OUTFIL statements for BUILD and OVERLAY operands.

Figure 4-2 on page 46 shows the INPUT, DFSORT control cards that use WEEKNUM=ISO and the OUTPUT.

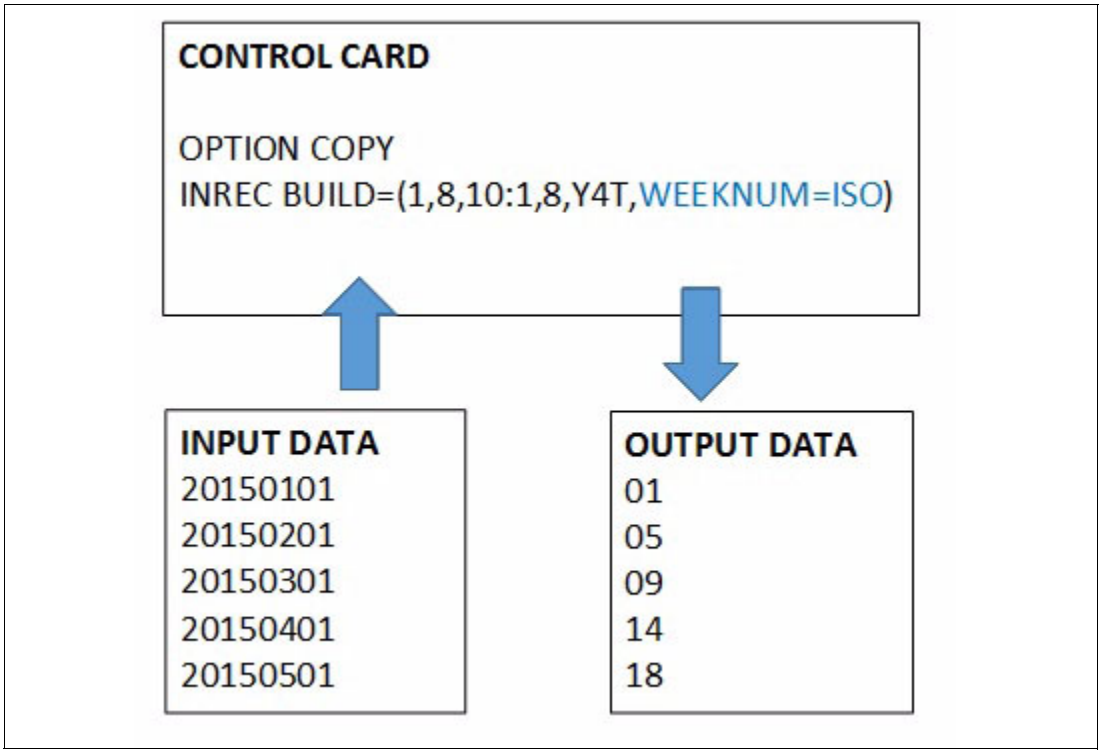


Figure 4-2 WEEKNUM=ISO

Note: If WEEKNUM=USA is used, there is a slight difference in the result starting on second line. Because 02/01/2015 is a Sunday, it is considered week six in the USA format.

If there is an error, message **ICE288I** is issued for these new functions as it is issued to other data conversion functions. The message text was not changed, as shown in Example 4-1

Example 4-1 ICE288I Message

```
ICE288I INPUT OR OUTPUT DATE VALUE OUT OF RANGE FOR DATE CONVERSION OR DATE DATE
ARITHMETIC
```

4.3 Use of zHPF

High Performance Ficon (zHPF) is a protocol extension of FICON®, which communicates in a single packet a set of commands that must be run. It allows the control unit to stream the data for multiple commands back in a single data transfer section for I/Os that are started by various access methods. This configuration improves the channel throughput on small block transfers. Since the early implementation in 2009, zHPF is improved to cover various applications.

Generally, DFSORT uses EXCP for processing basic and large formatted sequential input and output data sets.

In z/OS V2R2, DFSORT was updated to prefer the use of BSAM for SORTIN, SORTOUT, and OUTFIL when zHPF is available.

You can identify the use of the BSAM access method for your work by reviewing the **ICE084I** message. No DFSORT messages are related to zHPF usage on any logs or sysouts, as shown in Example 4-2.

Example 4-2 Sortin and Sortout examples

```
ICE084I 0 BSAM ACCESS METHOD USED FOR SORTOUT
ICE084I 0 BSAM ACCESS METHOD USED FOR SORTIN
```

The zHPF support is automatically selected when available and no user actions are necessary.

4.4 Message ICE099A update

The message ICE099A was updated in z/OS V2R2. This message is issued when an error is detected during member verification. When a BLDL verification failed (possibly because a member of partitioned data set does not exist), the message is issued.

The original message shows only the data set name. The message update shows the member name, which helps users to verify the error faster, as shown in Example 4-3.

Example 4-3 ICE099A Updated Message

```
ICE099A BLDL FAILED FOR (dd name) DATA SET, MEMBER (member name)
```

4.5 Joinkey indicators in SMF data

In z/OS V2R2, DFSORT now provides joinkeys indicators on SMF records so that users can track the jobs by using joinkeys.

SMF record type 16 was updated to include the following joinkey bits:

- ICEJOINM: Indicates Joinkeys Main Task; eighth bit on ’26’x
- ICEJOIN1: Indicates Joinkeys Sub Task1; seventh bit on ’24’x
- ICEJOIN2: Indicates Joinkeys Sub Task2; eighth bit on ’24’x

This update enables you to produce reports to analyze the use of joinkeys jobs.

5

Virtual Storage Access Method

This chapter describes the following enhancements that are provided at DFSMS V2R2 that apply to the Virtual Storage Access Method (VSAM):

- Faster search for Access Method Block List (AMBL)
- Support for DB2 greater than 2 M Open clusters
- Chained I/O for Spanned Records
- Message on all auto dumps
- Improved recovery for EOV with non-SMS managed clusters
- Dynamic Buffer Addition (DBA) function
- Index Record Locking

This chapter includes the following topics:

- 5.1, “VSAM overview” on page 50
- 5.2, “VSAM performance enhancements” on page 50
- 5.3, “VSAM RAS enhancement” on page 53
- 5.4, “VSAM RLS enhancements” on page 56

5.1 VSAM overview

VSAM is a collection of programs that help applications to request I/O operations. It is loaded at the initial program load (IPL) in the extended pageable link pack area (PLPA) common area. All access methods (VSAM included) belong to the DFSMS product.

An access method has the following general functions:

- Buffering to decrease the number of I/O operations and to make them more efficient
- Blocking and deblocking logical records into physical records
- Writing the I/O channel program
- Synchronize the task that is requesting the I/O operation with the ending of the I/O operation through an Wait/Post mechanism.
- Perform I/O error recovery, if necessary

The following access methods also are available in DFSMS:

- BPAM
- QSAM
- BSAM
- BDAM
- OAM

5.2 VSAM performance enhancements

In this section, we describe the performance enhancements for non record-level subscriber (RLS) clusters.

5.2.1 Faster search for Access Method Block List

Access Method Block List (AMBL) is a VSAM internal control block that is associated with opened clusters. Within VSAM, the AMBLs are organized in sequential chains. This chain can be long when many clusters are opened. Then, when an AMBL must be located (mainly during Close processing) through the chain, the performance can be affected.

For example, DB2 might keep up to 200,000 linear clusters open at a time, which leads to large AMBL chains. Assume a Close processing of a cluster in this DB2, it causes VSAM searches the AMBL chains to check whether the Close processing is the last Close processing for the cluster. This check can take some time, which causes performance degradation.

z/OS V2R2 addresses this issue with large AMBL chains and improved performance, mainly for DB2 linear clusters. VSAM now builds a binary tree to track the opens and closes for each cluster. In a sense, it looks similar to the Index volume table of contents (VTOC), which is a binary tree pointing to VTOC.

A faster search can be done to determine the last Open processing. Performance is not enhanced for a few opened clusters, but for a larger number.

This enhanced feature is in the z/OS V2R2 product and no action on the part of the installation is necessary.

5.2.2 Chained I/O for spanned records

A VSAM spanned record is a logical record that is larger than its control interval (CI) size. It allows the logical record to extend across or span CI boundaries. To use this kind of record, you must define your cluster by using the SPANNED parameter with the Data Class or **IDCAMS DEFINE** command.

The spanned record feature is useful when an application requires long logical records or when data records vary significantly in their length. In this case, the use of spanned records can reduce the amount of required DASD space.

In a VSAM with the SPANNED parameter defined, VSAM decides whether it is a spanned record, depending on the size of record and size of CI. The information about spanned record is written on record definition fields (RDFs) at CI end.

When a record is larger than CI, VSAM split the record in segments that fit into CI, fills the CIs exclusively with this record and, if the last segment of this record is smaller than CI, the free space cannot be used, except to extend the same record, as shown in Figure 5-1.

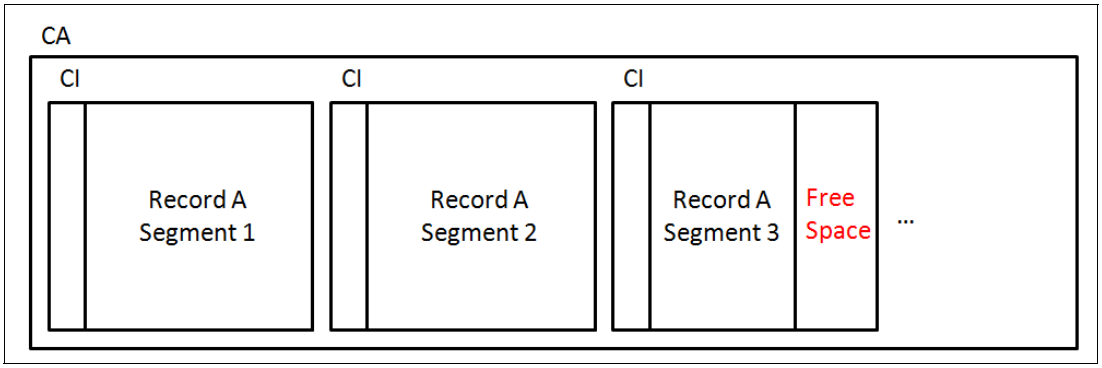


Figure 5-1 CA, CI, Record with segments and free space

The maximum size of a spanned record is the size of the control area (CA) because all segments of a spanned record must be in same CA. This limitation still can be a problem. Certain Alternative Index (AIX) clusters have long logical records. For example, the logical record at the data component of the index in the AIX can contain all of the client's account numbers (primary key) of a bank branch office (secondary key).

Before z/OS V2R2, segments of a spanned record in a CI are written serially one segment with its CI at a time. Then, one I/O operation per each segment.

Figure 5-2 shows one logical record larger than CI being written before DFSMS V2R2.

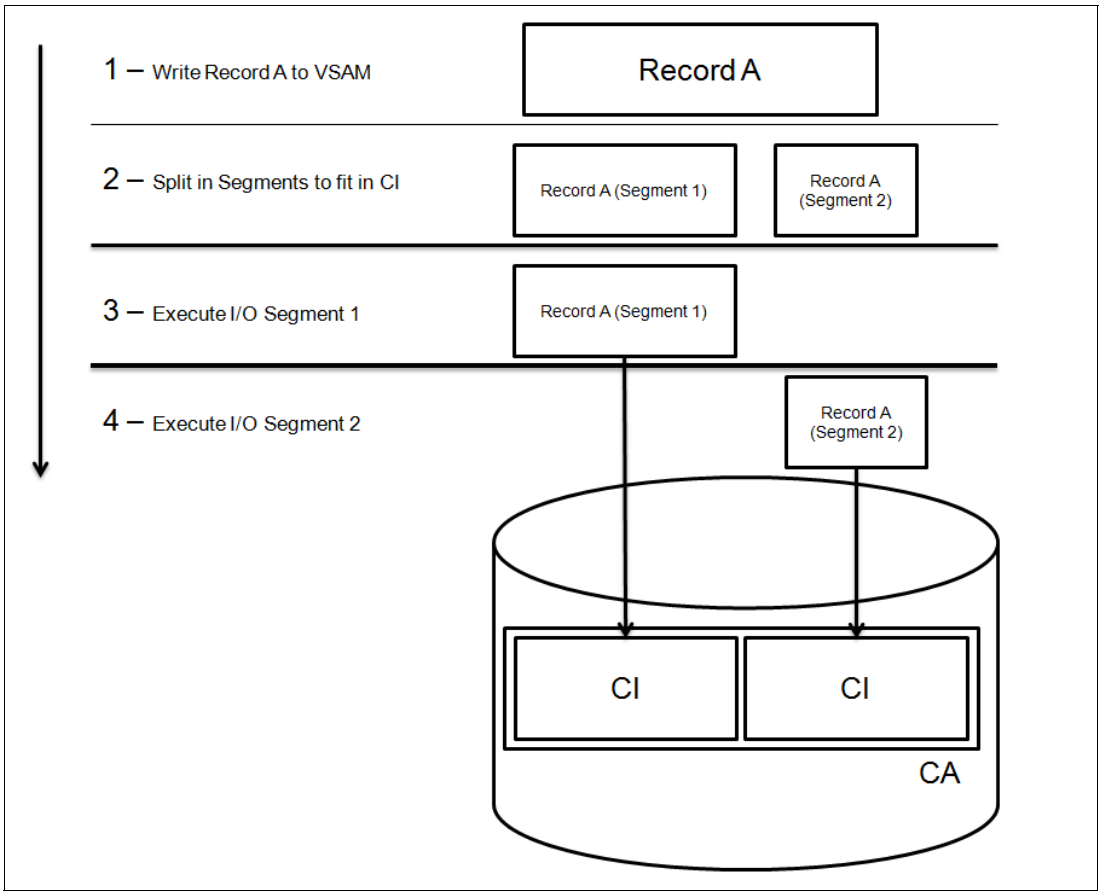


Figure 5-2 Spanned Record write before DFSMS V2R2

In z/OS V2R2, there is a new feature that chains the I/Os for the multiple segments (in multiple CIs) of a spanned record so that only one I/O operation is needed to write out the full spanned record. This feature is now available for NSR buffering only (not LSR or GSR).

Figure 5-3 shows one record larger than CI. It allows the logical record to extend across or span CI boundaries and is written in only one I/O operation.

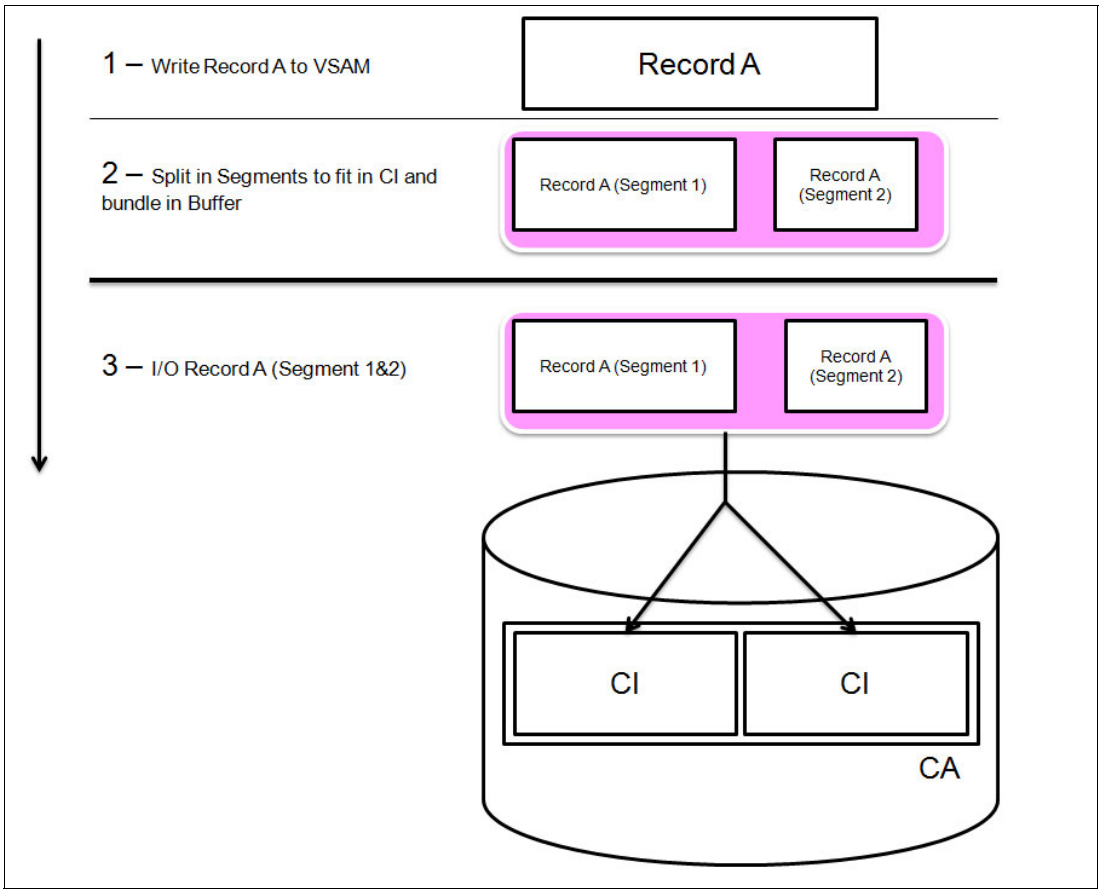


Figure 5-3 Spanned record write at DFSMS V2R2

With this change, it is expected that the I/O operations are reduced when spanned records are used. Also, if the write is interrupted, the record is kept in a consistent state and no recovery is needed.

To use this feature, it is necessary for enough NSR buffers to contain all of the segments of a record. If not enough buffers are available, VSAM acquires as many buffers as possible, but multiple I/Os are necessary to write out the record.

The feature is available by default and there are no migration and coexistence considerations.

5.3 VSAM RAS enhancement

This section describes enhancements in VSAM that are related to Reliability, Availability, and Serviceability (RAS). No actions are needed to use this enhanced features at DFSMS V2R2.

5.3.1 Support for DB2 greater than 2 M opened clusters

In IBM DB2 V11 for z/OS and DFSMS V2R1, the limit of opened clusters at one time is 200,000. Although this limitation is in place, the practical number can be much less, depending on the virtual storage that is available below the bar.

Many opened clusters lead to large amounts of virtual storage for VSAM control blocks that must be in 31-bit addressable private storage.

In z/OS V2R2, some of the control blocks for linear clusters (VSAM LDS) are moved above the bar, if requested so by the Media Manager caller.

Media Manager is the VSAM I/O driver. An I/O Driver is a set of programs that are running in supervisor state, Program Status Word (PSW) key zero, in charge of translating the I/O channel program, and fixing the pages of the I/O channel program and the I/O buffers.

This new feature can free the virtual storage that is below the bar and open more clusters because the storage constraint was relieved by using the 64-bit addressable memory.

This feature is intended to be used by future releases of IBM DB2 for z/OS and is not available in this release to other callers.

Important: The Media Manager interface for this improvement is not an externally documented interface. Only authorized callers can request the use of the mentioned feature.

5.3.2 Message on all auto dumps

When certain unexpected errors are detected, VSAM automatically generates a dump, and z/OS issues a message to indicate which memory dump data set was used to capture the memory dump, as shown in Example 5-1.

Example 5-1 Dump generated

```
*IEA911E COMPLETE DUMP ON SYS1.DUMP00
*DUMPID=001 REQUESTED BY JOB (CATALOG )
*FOR ASIDS(0022,002B)
*INCIDENT TOKEN: LOCAL    SYSTEM1  05/18/2006 21:56:50
```

If a memory dump is not taken, the user might not know why the memory dump was not taken. One reason for not having the memory dump might be a Dump Analysis and Elimination (DAE) action.

With z/OS V2R2, the message **IDA9999I** is enhanced to show VSAM Request Parameter List (RPL) control block feedback code that started the memory dump and SDUMP RC to explain why the memory dump was not captured. RPL is a VSAM control block that is used by the application at read/write to inform VSAM about the details of the requested I/O operation and to get an RPL feed back code that indicates how the I/O ended.

Example 5-2 shows message **IDA9999I** before z/OS V2R2.

Example 5-2 IDA9999I at DFSMS V2R1

```
IDA9999I VSAM AUTO DUMP FAILED TO TAKE A DUMP DUE TO SDUMPX RSN rsn
```

Example 5-3 shows the enhanced message with z/OS V2R2.

Example 5-3 IDA9999I at DFSMS V2R2

```
IDA9999I VSAM AUTO DUMP FAILED TO TAKE A DUMP FOR RPL FEEDBACK
CODE rpl_feedback_code DUE TO SDUMPX RSN/RC
sdump_reason/return_codes FOR JOB jobname
```

The enhanced message allows information about the error to be externalized, even though the memory dump was not captured.

5.3.3 Improved recovery for EOV with non-SMS clusters

End of Volume (EOV) is a z/OS component in charge of processing an end of extent situation for a DASD located cluster or an end of cartridge situation for tape data set.

When VSAM was used in non-SMS volumes before z/OS V2R2, an EOV does not clean up after itself if failure occurs; that is, some incorrect records can be left in VTOC and VSAM volume record (VVR) at catalogs. To discover and eventually fix the error, the installation must run the **`Verify`** and **`Diagnose`** commands.

z/OS V2R2 introduces changes to clean up the VTOC and VVR entries when an EOV failure occurs. This change allows subsequent extents to not be affected from invalid records that are left over from the previous fail extent.

This change is an internal change and no actions are needed.

Note: SMS-managed clusters have this clean up implemented.

5.3.4 Dynamic Buffer Addition function

The use of buffers in virtual storage is a technique in VSAM to improve performance. VSAM creates a resource pool that is a set of VSAM control blocks and several buffers with the same size (a buffer pool).

The resource pools can be shared among VSAM clusters, depending on the buffering technique that is being used. When local sharing resources (LSR) is used, the buffer pool is shared between clusters that are accessed by tasks of the same z/OS system.

LSR resource pools are built by applications through the BLDVRP macro before the first LSR cluster is opened.

During a peak use of this shared buffer pool, a logical error of “no buffers available” can occur. A typical scenario for such logical error is when the number of buffers was underestimated and the Defer Write option is On. By using a least recently used (LRU) algorithm, VSAM performs the Write I/O operations to free some of the buffers in the resource pool. However, this process takes time and the lack of available buffers condition occurs.

The user application program can redrive the request after other requests finish; however, this process can be a problem in the case of alternative indexes because there might not be enough buffers to back out the updates to the AIX. This issue leaves the index in an inconsistent state.

The buffering option was enhanced in z/OS V2R2. For LSR buffering, if there are no buffers available, VSAM tries to avoid the error by dynamically adding buffers to the shared resource pool.

In case of success, the message that is shown in Example 5-4 appears; otherwise, the behavior of previous releases occurs.

Example 5-4 Sample IDA9990I message

```
IDA9990I VSAM DBA ADDED xxxx DATA|INDEX BUFFERS BUFFERS of yyyyy
BYTES EACH TO SHRPOOL zzz BECAUSE THE SHARED POOL WAS TOO
SMALL FOR PROCESSING.
```

This new DFSMS V2R2 function also improves performance and provides a better RAS. There are no actions that are needed to use this enhancement. Note the buffer pools do not shrink after peak usage reduces.

Note: The intent of this enhancement is not to replace well-sized LSR pools; rather, it is to prevent errors during a peak usage or underestimated numbers of buffers.

5.4 VSAM RLS enhancements

VSAM RLS is a way to access the VSAM clusters in a task shared mode. The VSAM RLS code is in the SMSVSAM address space. It allows any number of user program tasks in the same Parallel Sysplex to access VSAM clusters to keep data integrity because of the use of structures at the Coupling Facility. The granularity of the serialization is performed at record level.

This section describes the enhancements that are related to this feature.

5.4.1 Index Record Locking

This enhancement improves VSAM RLS performance.

Although VSAM RLS uses a record level lock, VSAM might lock at cluster level to keep cluster integrity during usage. Before z/OS V2R2, the following operations require a cluster level lock:

- CI Split
- CI Reclaim
- Spanned Record activity
- CA SPLIT
- CA Reclaim

Cluster level locks create a single point of contention because another parallel I/O operation against the cluster cannot run at the same time.

z/OS V2R2 removes this single point of contention and can improve performance. The locks are made in the index component at the CI level and not at cluster level, which allows CI splits, CA reclaims, and spanned record activities to be done concurrently.

Information about Index Record Locks are written in type 42 SMF record.

The information is written in the following records:

- SMF42xxH*

  Number of component_1 class_4 (index record) locks (obtain/alter/promote).

- SMF42xxI*

  Number of component_1 class_4 (index record) locks that caused true contention.

- SMF42xxJ*

  Number of component_1 class_4 (index record) locks that caused false contention.

- SMF42xxK*

  Number of component_1 class_4 (index record) release lock requests.

  Where xx can be:

  - FP (Subtype 15, sysplex wide)
  - FS (Subtype 15, per system)
  - GS (Subtype 16, sysplex wide)
  - GT (Subtype 16, per system)
  - HE (Subtype 17, sysplex wide)
  - HJ (Subtype 17, per system)

Toleration APAR OA42676 is required to be applied on all V1R13 and V2R1 systems around the sysplex before bringing in any V2R2 systems.

Releases before V1R13 cannot be in the same sysplex as V2R2 systems. Broken clusters can occur as index record locking is a serialization change.

6

Catalogs and IDCAMS

This chapter describes the changes to catalogs and IDCAMS that were introduced with z/OS V2R2.

This chapter includes the following topics:

- 6.1, “Overview” on page 60
- 6.2, “Catalog RAS enhancements” on page 60
- 6.3, “IDCAMS enhancements” on page 63
- 6.4, “Generation Data Group enhancements” on page 66

6.1 Overview

Catalogs are vital to enable large environments to exist and be manageable. Without catalogs, the process of locating and accessing data sets on thousands of volumes is difficult. At the same time, IDCAMS assists users and support teams to print, merge, and fix data sets, among other options. The catalog and IDCAMS enhancements are described in this chapter.

6.2 Catalog RAS enhancements

The catalog enhancements provide increased Reliability, Availability, and Serviceability (RAS). These changes are described next.

6.2.1 Catalog modify command security enhancement

Having the access to perform **F CATALOG** commands, such as **F CATALOG,ALLOCATED** or **F CATALOG,REPORT** can be helpful to operations and other teams to retrieve information about catalog status. However, this access also provides the means to **CLOSE, UNALLOCATE** and other commands that are not intended to be performed by these teams.

Changes in z/OS V2R2

Starting in z/OS V2R2, a RACF resource profile was created to be checked by catalog command processing to ensure that the issuer of a command is authorized to run that command.

This function allows users that use SDSF to issue operator reporting commands, the results of which they can view at their computer. However, users are not permitted to issue commands that change the catalog configuration.

The only requirement for activating this facility is to define a new **OPERCMDS** resource profile, as shown in Example 6-1.

Example 6-1 New OPERCMDS resource profile

```
MVS.MODIFY.STC.CATALOG.CATALOG.SECURE
```

Then, **PERMIT** users and operators as needed to this resource by using the **RACF PERMIT** command. **READ** access allows the user to perform report commands, and **UPDATE** allows the user to issue any of the **MODIFY CATALOG** subcommands.

For more information about the new command security enhancement, see *DFSMS Managing Catalogs*, SC23-6853.

6.2.2 Catalog health check

By default, catalogs are defined with SHAREOPTIONS(3 4). A shared catalog must be defined with SHAREOPTIONS(3 4) and must be on a shared volume. Catalogs that are on shared volumes are damaged if referred to by another system and the share options are inconsistently set (not 3 4).

Changes in z/OS V2R2

z/OS V2R2 provides a new Catalog Health Check to inspect all the catalogs that are defined in the environment for share options and volume status (shared or non-shared) and report any inconsistencies between the two. System programmers can decide to redefine or alter the inconsistent catalogs with the correct share options during system downtime.

Usage and Invocation

After the IBM Health Checker for z/OS is up and running, an exit routine (IGG0CLHX) adds the new check to the system. If the check is successfully added, it starts running at the interval that is specified in the exit routine unless it is disabled or the interval is modified. The default interval for the check to run is every 720 hours (30 days). This interval can be changed by using the **F HZSPROC** command.

6.2.3 LISTCAT statistics for catalogs

Before z/OS V2R2, it was difficult to collect accurate statistics about a catalog when catalogs were not closed cleanly because of errors. Information about how many control areas (CAs) were reused were viewed only from the **CATSTATX** command, which collects information since the last initial program load (IPL) only.

However, this statistic problem was solved before DFSMS V2R2 by issuing the following command:

```
F CATALOG,REPORT,PERFORMANCE(RESET)
```

Example 6-2 shows **CATSTATX** command output.

Example 6-2 CATSTATX command output

```
F CATALOG,REPORT,CATSTATX(UCAT.BH5CAT)
IEC351I CATALOG ADDRESS SPACE MODIFY COMMAND ACTIVE
IEC359I EXTENDED CATALOG STATS
*CAS*****************************************************************
*  CATALOG NAME     = UCAT.BH5CAT                                     *
*  INSERTS (ADDS)   =      191                                        *
*  UPDATES          =       77                                        *
*  RETRIEVES        =   18,170                                        *
*  RETRIEVES FOR UP =      195                                        *
*  ERASES (DELETES) =      101                                        *
*  CA-RECLAIMS      =        0                                        *
*  CA-REUSES        =        0                                        *
*  BUFNI SETTING    =       11                                        *
*  BUFND SETTING    =       16                                        *
*  STRNO SETTING    =        8                                        *
*  AVG ELAPSED TIME =     0.397 MSEC                                  *
*  AVG CPU TIME     =    95.505 USEC                                  *
*CAS*****************************************************************
IEC352I CATALOG ADDRESS SPACE MODIFY COMMAND COMPLETED
```

Changes in z/OS V2R2

To keep statistics updated in VSAM volume record (VVR) and close catalogs properly, the CAS command **MODIFY CATALOG,CLOSE** was enhanced and internally addresses this issue. The major benefits on this enhancement are that statistics are available from a **LISTCAT** as with any other Key-Sequenced Data set (KSDS).

6.2.4 Restoring a catalog to Any Volume

When a catalog is restored from a backup memory dump, it must be restored to the same volume serial number (VOLSER) from where it was dumped.

Starting with z/OS V2R2, a catalog backup can be restored to any volume. To perform this restoration, the device capacity must match (3389 versus 3390), and a logical restore must be performed.

The restriction for physical data set restore still exists.

6.2.5 Connecting multi-line Write To Operator (WTO)

Starting with z/OS V2R2, large catalog multi-line operator messages (greater than 255 lines) are no longer cut off. Instead, the messages are displayed.

6.2.6 New catalog search interface fields

The Catalog Address Space (CAS) uses internal control blocks that are called CAXWAs to process catalogs. The CAXWAs are in common storage and when they are damaged, an IPL is used to fix the issue. Some vendors use the CAXWAs to retrieve information about a catalog.

Changes in z/OS V2R2

In z/OS V2R2, CAXWAs are created in CAS private storage; therefore, if there is a failure, a CAS can restart and be used to repair broken CAXWAs.

In addition, a new entry type of “Z” (used in the CSIDTYPS field) is used to tell the CSI that the information that is wanted is dynamic data from CAS control blocks rather than information that is stored in the catalog. No other types are allowed to be specified with the Z entry type.

For Z entry requests, the catalog name is specified in the filter key field (CSIFILTK); the catalog name field (CSICATNM) should be blank.

Example 6-3 shows a sample invocation of CSI by using the new CSI entry type.

Example 6-3 Sample CSI invocation

```
/* Initialize variables */
CSIFLTK = CatName; ! Name to be found
CSICATNM = ' '; ! Blank the catalog name
CSIRESNM = ' '; ! Blank the resume name
CSIDTYPS = 'Z '; ! CB data request
CSICLDI = ' '; ! Don't return index
CSIRESUM = ' '; ! Not in resume mode
CSIS1CAT = ' '; ! Search all cats
CSIOPTNS = 'F'; ! Use fullword lengths
CSINUMEN = 3; ! Retrieve 3 fields
CSIENTS(1) = 'CATACT '; ! Set 1st field name
CSIENTS(2) = 'CATUCB '; ! Set 2nd field name
CSIENTS(3) = 'CATFLAGS'; ! Set 3rd field name
CSIUSRLN = Length(L_Work_Area);
Call IGGCSI00(L_Reason_Area,L_Input_Area,L_Work_Area)
Retcode(L_CSI_Retcode);
```

```
/* Process output here */
Return Code(L_Retcode);
```

Note: For more information about CSI and new CSI field, see *DFSMS Managing Catalogs*, SC23-6853/

6.3 IDCAMS enhancements

In this section, we describe the enhancements to IDCAMS in z/OS V2R2.

6.3.1 DIAGNOSE

To quickly access the VVR/NVR that is associated with the catalog entry, the catalog record has a relative byte address (RBA) pointer to the VSAM volume data set (VVDS).

When VVR/NVR are moved and VVR RBA is not valid, the catalog management identifies this condition and updates the catalog record. Catalog management then performs a sequential search of the VVDS to re-establish the RBA.

Previous releases of DIAGNOSE identified this mismatch as an error and reports “not found” for the VVDS entry.

To improve the information from DIAGNOSE reporting, z/OS V2.2 introduced enhancements to perform the sequential VVDS search and re-establish the RBA. This new sequential VVDS search is performed automatically during DIAGNOSE and no user action is necessary.

6.3.2 PARM(TEST) command

The **PARM(TEST)** command is used to display diagnostic tracing. Debugging is enhanced to include 31-bit support and allocate memory and process work above the line. The use of above the line memory is a constraint relief and reduces the chances of abends that are caused by lack of memory.

6.3.3 REPRO MERGECAT command

To merge entries from one catalog to another, the **REPRO MERGECAT** command produces up to six lines for each entry that is processed. The following lines are available:

- One blank line
- One or two lines for the **IDC0639I** message (sphere conversion started)
- A blank line
- One or two lines for the **IDC01402I** (sphere conversion ended)

Changes that were introduced in z/OS V2R2 enhance IDCAMS to provide a new optional keyword, **MESSAGELEVEL(*ALL* | *SHORT*)**, for **REPRO MERGECAT**. The **SHORT** option reduces the number of lines for each entry. The **ALL** option is default and is the same as the current output, as shown in Example 6-4.

Example 6-4 New REPRO MERGECAT keyword, MESSAGELEVEL

```
REPRO INDATASET(SMSCAT) OUTDATASET(SMSCAT1) -
MERGECAT MESSAGELEVEL(SHORT) FILE(DD1)
```

When **MESSAGELEVEL(SHORT)** is used, **IDC0639I** and **IDC01402I** messages are replaced by the new **IDC01401I** message, as seen in Example 6-5.

Example 6-5 REPRO Output reduced by using MESSAGELEVEL(SHORT)

```
REPRO  IFILE(INFILE) OFILE(OUTFILE) LEVEL(GILTST1)   -
  MESSAGELEVEL(SHORT) MERGECAT
IDC01401I SUCCESSFUL MERGE FOR GILTST1.TEST.V000001
IDC01401I SUCCESSFUL MERGE FOR GILTST1.TEST.V000002
```

If the REPRO MERGECAT fails, only **IDC0639I** message is suppressed.

6.3.4 TSO LISTCAT command

The **LISTCAT** command is used to list catalog entries. A new parameter **PREFIX/NOPREFIX** was introduced to **LISTCAT** that allows the user to add the TSO user ID when **LISTCAT** is run, as shown in Example 6-6.

Example 6-6 New LISTCAT parameter

```
LISTCAT ENTRY(A.B.C) NOPREFIX
searches the catalog for A.B.C

LISTCAT ENTRY(A.B.C) PREFIX
searches the catalog for userid.A.B.C
```

6.3.5 PRINT and REPRO commands

The **PRINT** command can be used to print VSAM data sets, non-VSAM data sets, and catalogs. The **REPRO** command is used to copy some or all records from VSAM and non-VSAM data sets, catalogs, master catalogs, and volume catalogs.

Before z/OS V2.2, the **REPRO** and **PRINT** commands did not process damaged VSAM CIs or records.

In z/OS V2.2, a new keyword, **CIMODE**, was added to these commands. This new keyword enables a broken VSAM ESDS data set to be read and good records extracted from it. CIMODE can be used only with the following keywords:

- INFILE
- INDATASET
- ERRORLIMIT
- FROMADDRESS
- SKIP
- TOADDRESS
- COUNT
- NOMERGECAT
- NOREPLACE
- NOREUSE

Example 6-7 shows how this new keyword is used.

Example 6-7 CIMODE example

```
PRINT INDATASET(SMS.ESDS) CIMODE
```

6.3.6 VERIFY RECOVER command

The **VERIFY** command is used to correctly reflect the status of a VSAM data set after an error occurs while a VSAM data set is closing.

In z/OS V2R2, the following methods are available to run the **VERIFY** command:

- IDCAMS VERIFY
- IDCAMS VERIFY RECOVER
- EXAMINE/IDCAMS VERIFY RECOVER

IDCAMS VERIFY is the simplest way to run the command. IDCAMS opens the data set for output and then issues the **VERIFY** macro with no options and then, closes the data set.

In **IDCAMS VERIFY RECOVER**, the **RECOVER** is used to fix errors that are caused by an incomplete CA reclaim. This option causes VSAM Record Management **VERIFY** to back out or complete any interrupted CA reclaim in addition to regular **IDCAMS VERIFY** functions. There is no change to this method of running **VERIFY**.

The **EXAMINE** command analyzes and reports on the structural integrity of the index and data components of a key-sequenced data set cluster (KSDS) and of a variable-length relative record data set cluster (VRRDS).

The third method, **EXAMINE/IDCAMS VERIFY RECOVER**, is a new way to use the **VERIFY** command. **EXAMINE** was enhanced to pass the error information that it finds (such as index CI#, data CI#, error type, and so forth) to **VERIFY**. If there is no concurrent access on the data server, **VERIFY** attempts to repair any error that it can.

This new feature detects and correct the following errors that are reported by the messages that were issued by **EXAMINE**:

```
IDC11718I DATA COMPONENT HIGH-USED RBA IS NOT EQUAL TO CA SIZE

IDC11728I DATA FOUND IN EMPTY CI
```

This enhancement works only when **EXAMINE** and **VERIFY RECOVER** are run in the same IDCAMS job step, as shown in Example 6-8.

Example 6-8 EXAMINE/IDCAMS VERIFY RECOVER job example

```
//STEP10 EXEC PGM=IDCAMS
//SYSUDUMP DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//SYSABEND DD SYSOUT=*
//SYSIN DD *
EXAMINE NAME(SYSPLEX.CARVSM28.KSDS01) DATATEST INDEXTEST
VERIFY DATASET(SYSPLEX.CARVSM28.KSDS01) RECOVER
```

In Example 6-9, some of the new messages alerts are shown.

Example 6-9 EXAMINE/IDCAMS VERIFY RECOVER output

```
EXAMINE NAME(SYSPLEX.VSMEVR03.KSDS01) DATATEST NOINDEXTEST 04870042
IDC01701I DATATEST BEGINS
IDC11728I DATA FOUND IN EMPTY CI
IDC01713I DATA CONTROL INTERVAL DISPLAY AT RBA 419 FOLLOWS
000000 C1C2C3C4 C5C6C7C8 C9D1F0F0 F0F0F2F5 F1F8D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 *ABCDEFGHIJ00002518KLMNOPQRSTUVWX*
000020 E8E98182 83848586 87888991 92939495 96979899 A2A3A4A5 A6A7A8A9 C1C2C3C4 *YZ.....................ABCD*
000040 C5C6C7C8 C9D1D2D3 D4D5D6D7 D8D9E2E3 E4E5E6E7 E8E98182 83848586 87888991 *EFGHIJKLMNOPQRSTUVWXYZ.........*
```

```
000060 92939495 96979899 A2A3A4A5 A6A7A8A9 C1C2C3C4 C5C6C7C8 C9D1D2D3 D4D5D6D7 *.............ABCDEFGHIJKLMNOP*
000080 D8D9E2E3 E4E5E6E7 E8E98182 83848586 87888991 92939495 96979899 A2A3A4A5 *QRSTUVWXYZ.....................*
...lines removed to fit example on page...
000760 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *................................*
000780 00000000 00000000 00F70701 13F60701 12000AF5 070111F4 07011000 0AF30701 *.........7...6.....5...4.....3..*
0007A0 0FF20701 0E000BF1 F1060200 F1F00602 03000A07 0004F807 0105000A F7070106 *.2.....11...10........8.....7...*
0007C0 F6070107 000AF507 0108F407 0109000A F307010A F207010B 000CF5F0 F105030C *6.....5...4.....3...2.....501...*
0007E0 06000200 0FF0F0F0 F0F2F4F8 000701F0 F0F0F0F2 F4F70007 000007F9 07F90000 *.....0000248...0000247.....9.9..*
IDC01714I ERROR LOCATED AT OFFSET 00000018
IDC21702I MINOR ERRORS FOUND BY DATATEST
IDC01708I 749 CONTROL INTERVALS ENCOUNTERED
IDC01710I DATA COMPONENT CONTAINS 749 RECORDS
IDC01711I DATA COMPONENT CONTAINS 0 DELETED CONTROL INTERVALS
IDC01712I MAXIMUM LENGTH DATA RECORD CONTAINS 2000 BYTES
IDC01722I 99 PERCENT FREE SPACE
IDC11995I RECOMMENDATION: ISSUE IDCAMS EXAMINE AND VERIFY RECOVERY TO ATTEMPT TO
REPAIR THE PRECEDING ERROR
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 4
VERIFY DATASET(SYSPLEX.VSMEVR03.KSDS01) RECOVER
IDC11997I THE PRECEDING IDC11728I ERROR HAS BEEN ADDRESSED.
IDC11994I RECOMMENDATION: RERUN IDCAMS EXAMINE.
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
```

Consider the following points regarding Example 6-9 on page 65:

- The **IDC11994I** message is shown after **IDC11996I** or **IDC11997I** to recommend users to rerun **IDACAMS EXAMINE** to verify whether the preceding error messages that were found by IDCAMS EXAMINE are now fixed.
- The IDC11995I message indicates that at least one error in the data set was found that can be addressed by issuing **IDCAMS EXAMINE** and **IDCAMS VERIFY RECOVER** in the same job step.
- The IDC11997I message indicates that attempt an attempt was made to repair the preceding IDC117xxl errors.
- The IDC11998I message indicates that **IDCAMS VERIFY RECOVER** does not attempt to repair the preceding messages because of concurrent access on the data set.
- The **IDC11999I** message indicates that **IDCAMS VERIFY RECOVER** attempted to repair the preceding IDC117xxl message but failed.

6.4 Generation Data Group enhancements

Generation Data Group (GDG) can be set up to store multiple versions of relevant data to users and applications. The number of Generation Data Sets (GDS) that can be active within a GDG and the default SCRATCH parameter were enhanced to provide better management of GDS. A new PURGE option specifies that when the SCRATCH option is set, unexpired GDSs are to be deleted during roll off processing. These updates are described next.

6.4.1 GDG extended

The LIMIT attribute is used to define how many GDSs can be active within a GDG entry. Although the maximum limit of 255 entries is enough to handle most user needs, it might not be sufficient to hold all versions that are required by application programs that are running several times a day for an extended time.

z/OS V2R2 introduces a new EXTENDED keyword to allow up to 999 generations to be active within a GDG. The extra number of active GDSs can hold more than two years of data for a daily output.

Usage and invocation

Before you can define GDGs by using the EXDENDED keyword, you must enable GDGEXTENDED option in SYS1.PARMLIB member IGGCATxx. The default value for this attribute is NO, meaning EXTENDED GDGs are not allowed. To enable Generation Data Group Extended (GDGE) support, set GDGEXTENDED, as shown in Example 6-10. You will need to either IPL, or restart CATALOG address space in order to refresh IGGCATxx member, and activate changes.

Example 6-10 SYS1.PARMLIB member IGGCATxx

```
GDGEXTENDED(YES)
```

Note: Because this keyword is new to z/OS V2R2, do not implement it on an IGGCAT member that is shared by other logical partitions (LPARs) until all LPARs are at z/OS V2R2 or higher.

After you enable GDGE support in your environment, you can define EXTENDED GDGs by coding the keyword EXTENDED (or its abbreviation EXT) on the **DEFINE GDG** command. Example 6-11 shows a **DEFINE GDG** command with EXTENDED keyword and LIMIT of 500 entries.

Example 6-11 DEFINE GDG command

```
DEFINE GDG(NAME(KWRES08.TEST.GDG) EXTENDED LIMIT(500))
```

All other attributes (SCRATCH, EMPTY, FOR, TO...) affect GDGE entries the same way standard GDGs.

Migration and coexistence

All systems in an installation should be at z/OS V2R2 or higher before enabling GDGEs. Coexistence APAR OA46009 prevents access to GDGEs on lower-level systems. Not applying coexistence APAR can give unpredictable results.

We suggest you to enable GDGE in either all, or none LPARs in your environment. Although having a mixed environment is possible, it might be confusing to have specific LPARs where EXTENDED keyword is permitted when defining GDG base.

6.4.2 GDG SCRATCH Parmlib value

When defining GDG entries, if SCRATCH option is not coded, the default NOSCRATCH is assigned to the GDG and data sets are not uncataloged when rolled off.

Although this function can be useful for users that are expecting to keep their data cataloged after they are rolled off from GDG, unattended users waste system resources by keeping data longer than needed and face errors if all available GDSs (G0001V00 - G9999V00) are cataloged.

Starting with z/OS V2R2, a default value can be defined for SCRATCH processing to override the NOSCRATCH keyword that is used in the **DEFINE GDG** command.

Invocation and use

The GDGSCRATCH keyword in SYS1.PARMLIB IGGCATxx allows the system administrator to override IDCAMS DEFINE GDG NOSCRATCH keyword. If you do not set this variable or it is set to NO, the keyword that is specified in the **DEFINE GDG** command takes precedence. Example 6-12 shows the attribute to enable GDG SCRATCH on parmlib.

Example 6-12 Enable GDG SCRATCH

```
GDGSCRATCH(YES)
```

Note: Setting up GDGSCRATCH(YES) does not change the original setting of defined GDGs.

After GDGSCRATCH is enabled, *all* subsequent GDG defines use the SCRATCH attribute, regardless the **DEFINE** command options that are used. Add GDGSCRATCH(YES) to SYS1.PARMLIB(IGGCATxx) system wide, if wanted.

6.4.3 GDG PURGE keyword

During roll off processing, if a data set is identified as unexpired because of FOR or TO keywords, the GDS is not scratched, even if the SCRATCH keyword was used on GDG definition. The PURGE keyword was introduced to provide a way to delete unexpired data sets during roll off processing.

Invocation and use

In addition to PURGE keyword on **DEFINE** or **ALTER GDG** commands, you can define an override value for PURGE in SYS1.PARMLIB IGGCATxx member. The default value for **GDGPURGE** is NO. Example 6-13 shows the syntax to implement **GDGPURGE**.

Example 6-13 Sample GDGPURGE activate

```
GDGPURGE(YES)
```

Note: The PURGE attribute is used only when SCRATCH is also set.

Example 6-14 shows the **ALTER** command to enable PURGE for a GDG.

Example 6-14 Enable PURGE for GDG

```
ALTER KWRES08.GDG1 PURGE
```

You can check the GDG attributes (including PURGE) by using an **LISTCAT** command. Example 6-15 shows the output of this command.

Example 6-15 LISTCAT ENT(KWres08.GDG3) ALL output

```
GDG BASE ------ KWRES08.GDG3
     IN-CAT --- UCAT.BH5CAT
     HISTORY
       DATASET-OWNER-----(NULL)     CREATION--------2015.253
       RELEASE----------------2     LAST ALTER------0000.000
     ATTRIBUTES
       LIMIT-------------------5     SCRATCH    NOEMPTY    LIFO       PURGE
 NOEXTENDED
     ASSOCIATIONS--------(NULL)
```

Migration and coexistence

GDG PURGE support is not available on pre-z/OS V2R2 systems, although this support might be set by using a z/OS 2.2 system to DEFINE the GDG. In this scenario, the PURGE setting is ignored.

Related publications

The publications that are listed in this section are considered particularly suitable for a more detailed discussion of the topics that are covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide more information that is related to the z/OS V2R2 updates. Some publications that are referenced in this list might be available in softcopy only:

- *z/OS V2R2: JES2, JES3, and SDSF*, SG24-8287-00
- *z/OS V2R2: Security,* SG24-8288-00
- *z/OS V2R2: Storage Management and Utilities,* SG24-8289-00
- *z/OS V2R2: Availability Management,* SG24-8290-00
- *z/OS V2R2: Performance,* SG24-8292-00
- *z/OS V2R2: Operations,* SG24-8305-00
- *z/OS V2R2: Diagnostics,* SG24-8306-00
- *z/OS V2R2: Sysplex,* SG24-8307-00
- *z/OS V2R2: UNIX System Services* SG24-8310-00
- *z/OS V2R2: User Interfaces,* SG24-8311-00
- *z/OS V2R2: ServerPac*, SG24-8500-00

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, draft, and other materials at the following website:

ibm.com/redbooks

Other publications

The following publications also are relevant as further information sources:

- *z/OS V2R2 DFSMShsm Data Areas*, GC14-7504-02
- *z/OS V2R2 DFSMS Access Method Services Commands*, SC23-6846-02
- *z/OS V2R2 DFSMS Advanced Copy Services*, SC23-6847-02
- *z/OS V2R2 DFSMS Implementing System-Managed Storage*, SC23-6849-01
- *z/OS V2R2 DFSMS Installation Exits*, SC23-6850-02
- *z/OS V2R2 DFSMS Macro Instructions for Data Sets*, SC23-6852-03
- *z/OS V2R2 DFSMS Managing Catalogs*, SC23-6853-02
- *z/OS V2R2 DFSMS OAM Application Programmer's Reference*, SC23-6865-01
- *z/OS V2R2 DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*, SC23-6866-01
- *z/OS V2R2 DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*, SC23-6867-01
- *z/OS V2R2 DFSMS Using the New Functions*, SC23-6857-04
- *z/OS V2R2 DFSMS Using the Volume Mount Analyzer*, SC23-6859-01

- *z/OS V2R2 DFSMS Using Data Sets*, SC23-6855-03
- *z/OS V2R2 DFSMS Using ISMF*, SC23-6856-01
- *z/OS V2R2 DFSMS Using Magnetic Tapes*,C23-6858-01
- *z/OS V2R2 DFSMSdfp Advanced Services*, SC23-6861-02
- *z/OS V2R2 DFSMSdfp Diagnosis*, SC23-6863-01
- *z/OS V2R2 DFSMSdfp Storage Administration*, SC23-6860-02
- *z/OS V2R2 DFSMSdfp Utilities*, SC23-6864-01
- *z/OS V2R2 DFSMSdss Storage Administration*, SC23-6868-03
- *z/OS V2R2 DFSMShsm Diagnosis*,GC52-1387-01
- *z/OS V2R2 DFSMShsm Implementation and Customization Guide*, SC23-6869-02
- *z/OS V2R2 DFSMShsm Managing Your Own Data*, SC23-6870-01
- *z/OS V2R2 DFSMShsm Storage Administration*, SC23-6871-03
- *z/OS V2R2 DFSMSrmm Application Programming Interface*, SC23-6872-01
- *z/OS V2R2 DFSMSrmm Diagnosis Guide*, SC23-6876-01
- *z/OS V2R2 DFSMSrmm Implementation and Customization Guide*, SC23-6874-01
- *z/OS V2R2 DFSMSrmm Managing and Using Removable Media*, SC23-6873-01
- *z/OS V2R2 DFSMSrmm Reporting*, SC23-6875-01
- *z/OS V2R2 DFSMStvs Planning and Operating Guide*, SC23-6877-01

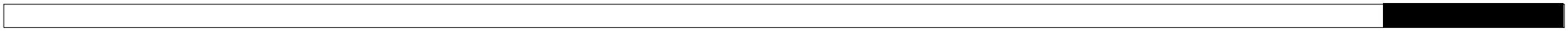
Help from IBM

IBM Support and downloads:

ibm.com/support

IBM Global Services:

ibm.com/services



(0.1”spine)
0.1”<->0.169”
53<->89 pages

SG24-8289-00

ISBN 0738441287

Printed in U.S.A.