

z/OS



DFSMS Using the New Functions

Version 2 Release 2

Note

Before using this information and the product it supports, read the information in "Notices" on page 81.

This edition applies to Version 2 Release 2 of z/OS (5650-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 2004, 2015.

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

Contents

About This Information	v	
Required product knowledge	v	
z/OS information	v	
How to send your comments to IBM	vii	
If you have a technical problem	vii	
Summary of changes	ix	
Summary of changes for z/OS Version 2 Release 2 (V2R2)	ix	
Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated February 2015	ix	
Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated September 2014	x	
Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated March 2014	x	
Summary of changes for z/OS Version 2 Release 1	x	
Introduction	xi	
What do I need to do to use the z/OS DFSMS enhancements?	xi	
Part 1. Using new DFSMS functions in z/OS V2R2	1	
 Chapter 1. Using the catalog enhancements in z/OS V2R2	3	
 Chapter 2. Using the SMS enhancements	7	
 Chapter 3. Using the Open/Close/End of Volume enhancements in z/OS V2R2.	11	
 Chapter 4. Using the Object Access Method (OAM) enhancements	13	
 Chapter 5. Using the new VSAM functions	15	
 Chapter 6. Using the DADSM/CVAF enhancements	17	
 Chapter 7. Using the DFSMSHsm enhancements	19	
Administering	20	
Administering storage tiers	20	
Administering distributed tape processing	22	
Administering message simplification	24	
Chapter 8. Using the DFMSrmm enhancements	25	
Part 2. Using New DFSMS Functions in z/OS V2R1	27	
Chapter 9. Using the Advanced Copy Services enhancements	29	
Chapter 10. Using the catalog enhancements in z/OS V2R1	31	
Administering	33	
Do Not Allocate a State Data Set as a GDS PDSE	33	
Enable JES3 allocation assist tape TS7700 in DEVSUPxx	33	
Set Up VSAM RLS Directory Only Caching.	33	
Chapter 11. Using the SMS enhancements	35	
Exploiting clusters and extent pools in SMS volume selection	35	
Providing accurate space statistics	36	
Providing option to suppress SMS messages	36	
Providing an option to guarantee that a PDS is created	36	
Changes to LISTCAT output.	36	
Chapter 12. Using the Open/Close/End of Volume enhancements in z/OS V2R1.	37	
Chapter 13. Using the PDSE enhancements	39	
Chapter 14. Using the Utility enhancements	41	
Chapter 15. Using the Object Access Method (OAM) enhancements	43	
Chapter 16. Using VSAM enhancements	45	
Administering	45	
Specifying a value for dynamic volume count	46	
Specifying eligibility for VSAM replication	46	
Specifying VSAM SMB access method bias using a data class	46	
Programming.	46	
Using SHOWCB enhancements to display an Access Control Block	47	
Using new attributes collected by DCOLLECT.	47	

Chapter 17. Using the zEnterprise Data Compression (zEDC) enhancements	49
Administering	53
Specifying zEDC compression at the system level	53
Specifying zEDC compression at the data set level.	54
Chapter 18. Using DADSM/CVAF enhancements	55
Chapter 19. Using the DFSMSdss enhancements	57
Administering.	57
Protecting the use of the RESET keyword	58
Operating	58
Using the RESET keyword with DFSMSdss commands.	58
Using DEBUG(SMSMSG) with the CONVERTV, COPY and RESTORE commands	58
Using FCCGVERIFY with the CGCREATED command	58
Using REPLACEUNCONDITIONAL with the RESTORE command for physical data sets	59
Specifying zEDC compression for dumps	59
Chapter 20. Using the DFSMShsm enhancements	61
An overview of storage tiers.	62
Last successful class transition date (LSCTD)	64
Class transition movement	65
Storage group processing priority	66
Recall	67
Reporting	67
Related tasks	67
Planning and Installation	67
Planning to use the DFSMShsm enhancements	68
Planning to use storage tiers.	68
Administering	69
Administration tasks for storage tiers.	69
Using the MIGRATIONSUBTASKS parameter with the DFSMShsm SETSYS command	69
Operating	69
Operation tasks for storage tiers	70
Operation tasks for serviceability and usability enhancements	70
Using the NEWNAME parameter with the DFSMShsm FRRECOV command	71
Operation tasks for zEDC compression	71
Using the ZCOMPRESS parameter with the DFSMShsm DEFINE DUMPCLASS command	71
Using the ZCOMPRESS parameter with the DFSMShsm SETSYS command	71
Chapter 21. Using the DFSMSrmm enhancements	73
Part 3. Appendixes.	75
Appendix. Accessibility	77
Accessibility features	77
Consult assistive technologies	77
Keyboard navigation of the user interface	77
Dotted decimal syntax diagrams	77
Notices	81
Policy for unsupported hardware	82
Minimum supported hardware	83
Trademarks.	85
Glossary	87
Index	93

About This Information

This information helps you to understand the enhancements to DFSMSdfp and the DFSMS features in recent releases of z/OS. Appropriate cross references are provided throughout the document if you need more background information about a specific concept or task.

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

For information about accessibility features of z/OS, for users who have a physical disability, please see "Accessibility," on page 77.

Required product knowledge

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

z/OS information

This information explains how z/OS references information in other documents and on the web.

When possible, this information uses cross document links that go directly to the topic in reference using shortened versions of the document title. For complete titles and order numbers of the documents for all products that are part of z/OS, see *z/OS Information Roadmap*.

To find the complete z/OS® library, go to IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome>).

How to send your comments to IBM

We appreciate your input on this publication. Feel free to comment on the clarity, accuracy, and completeness of the information or provide any other feedback that you have.

Use one of the following methods to send your comments:

1. Send an email to mhvrccfs@us.ibm.com.
2. Send an email from the "Contact us" web page for z/OS (<http://www.ibm.com/systems/z/os/zos/webqs.html>).

Include the following information:

- Your name and address.
- Your email address.
- Your telephone or fax number.
- The publication title and order number:
z/OS V2R2 DFSMS Using the New Functions
SC23-6857-04
- The topic and page number that is related to your comment.
- The text of your comment.

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

IBM or any other organizations use the personal information that you supply to contact you only about the issues that you submit.

If you have a technical problem

Do not use the feedback methods that are listed for sending comments. Instead, take one of the following actions:

- Contact your IBM service representative.
- Call IBM technical support.
- Visit the IBM Support Portal at z/OS Support Portal (<http://www-947.ibm.com/systems/support/z/zos/>).

Summary of changes

This information includes terminology, maintenance, and editorial changes. Technical changes or additions to the text and illustrations for the current edition are indicated by a vertical line to the left of the change.

Summary of changes for z/OS Version 2 Release 2 (V2R2)

The following changes are made for z/OS V2R2.

New

- Chapter 1, "Using the catalog enhancements in z/OS V2R2," on page 3.
- Chapter 2, "Using the SMS enhancements," on page 7
- Chapter 3, "Using the Open/Close/End of Volume enhancements in z/OS V2R2," on page 11
- Chapter 4, "Using the Object Access Method (OAM) enhancements," on page 13
- Chapter 5, "Using the new VSAM functions," on page 15
- Chapter 6, "Using the DADSM/CVAF enhancements," on page 17
- Chapter 7, "Using the DFSMShsm enhancements," on page 19
- Chapter 8, "Using the DFMSrmm enhancements," on page 25

Changed

- Chapter 10, "Using the catalog enhancements in z/OS V2R1," on page 31 has been updated with information about GDG FIFO parameters and IDCAMS support for RLS.
- Chapter 16, "Using VSAM enhancements," on page 45 has been updated to include the new EXTENDEDADDRESSABLE parameter of the IDCAMS ALTER command, and the ability that it gives VSAM linear data sets to use extended addressability without needing to be SMS-managed or extended-format.

Deleted

No content was removed from this information.

Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated February 2015

The following changes are made for z/OS V2R1 as updated February 2015.

New

Support for Multi-Target Mirror. For more information, refer to Chapter 9, "Using the Advanced Copy Services enhancements," on page 29.

Support for multiple Incremental FlashCopy targets (Incremental FlashCopy Version 2). For more information, refer to Chapter 9, "Using the Advanced Copy Services enhancements," on page 29.

Support for workload-based write pacing. For more information, refer to Chapter 9, "Using the Advanced Copy Services enhancements," on page 29.

Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated September 2014

The following changes are made for z/OS V2R1 as updated September 2014.

Changed

The DEFINE DUMPCLASS command now includes a ZCOMPRESS parameter, to specify whether to use zEnterprise® data compression (zEDC) Services on the dump data. The SETSYS command now includes a ZCOMPRESS parameter, to specify the type of compression used during migration or backup for all data sets. For more information, refer to Chapter 19, “Using the DFSMSdss enhancements,” on page 57 and Chapter 20, “Using the DFSMShsm enhancements,” on page 61.

Summary of changes for z/OS Version 2 Release 1 (V2R1) as updated March 2014

The following changes are made for z/OS V2R1 as updated March 2014.

New

New function is added for zEnterprise data compression (zEDC) of non-VSAM compressed format data sets. See Chapter 17, “Using the zEnterprise Data Compression (zEDC) enhancements,” on page 49 for more information.

Summary of changes for z/OS Version 2 Release 1

See the following publications for all enhancements to z/OS Version 2 Release 1 (V2R1):

- *z/OS Migration*
- *z/OS Planning for Installation*
- *z/OS Summary of Message and Interface Changes*
- *z/OS Introduction and Release Guide*

Introduction

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

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

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

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

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

This book describes the DFSMS enhancements in z/OS V2R2.

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

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

Evaluating

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

Planning

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

Installing

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

Administering

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

Operating

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

Customizing

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

Application programming

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

Diagnosing

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

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

Part 1. Using new DFSMS functions in z/OS V2R2

These topics describe how to use new DFSMS functions in z/OS V2R2.

Chapter 1. Using the catalog enhancements in z/OS V2R2

z/OS V2R2 includes several enhancements for DFSMS catalogs, described in this section.

Catalog RAS Enhancements: There are multiple RAS enhancements for DFSMS catalogs:

- **Catalog attributes health check:** This enhancement is designed to inspect all the catalogs currently defined in the user's environment for shareoptions and DASD status and report any inconsistencies between the two. The CATALOG_ATTRIBUTE_CHECK will notify the system programmer of any inconsistent catalog(s) in their environment. The notification will be done in the form of a report in the message buffer using SDSF. The system programmer can decide to redefine or alter the inconsistent catalog(s) with the correct shareoptions at a convenient time. See the following:
 - For new messages, see *z/OS MVS System Messages, Vol 9 (IGF-IWM)*
 - For more information on the CATALOG_ATTRIBUTE_CHECK, see *IBM Health Checker for z/OS User's Guide*
- **CSI Enhancements:** This enhancement includes new CSI fields. For more information, refer to Sample Z Entry Request Output in *z/OS DFSMS Managing Catalogs*.
- **DUMPON support for catalog front end modules:** When you request CAS dynamic dumping with the MODIFY CATALOG,DUMPON command, you can now specify modules involved in catalog front end processing. For more information, refer to MODIFY CATALOG in *z/OS DFSMS Managing Catalogs*.
- **Catalog to use connect with MLWTO:** This enhancement will assist all users of Catalog modify reports who are currently or will be in a situation where the maximum limit for the MLWTO is exceeded.
- **Restore a user catalog to any volume:** You can now logically restore a user catalog to any volume with the same device type as the volume from which it was dumped. Previously, you could restore only to the same volume. Physically restoring a user catalog is not changed — it must be done to the same volume as the volume from which the user catalog was dumped. For more information, refer to Restoring integrated catalog facility catalogs in *z/OS DFSMSdss Storage Administration*.

Catalog modify command security enhancements: This enhancement will allow the MODIFY command for Catalog to separate authorization for those sub-commands which provide reporting capabilities from those sub-commands which alter the catalog environment. This is achieved by defining a new RACF Resource for the OPERCMDS class and permitting selected users with READ or UPDATE access.

IDCAMS small enhancements: There are multiple enhancements in IDCAMS:

- A new parameter has been introduced to allow the user to control adding the TSO user id as a prefix when running LISTCAT as a TSO command. This new parameter is PREFIX/NOPREFIX.
- A new option "CIMODE" has been introduced in the PRINT command and REPRO command to process data sets using CI level processing. This will allow the PRINT and REPRO commands to be able to read a broken VSAM ESDS dataset and extract the good records from it.

IDCAMS RAS enhancements: There are multiple RAS enhancements in IDCAMS:

- **IDCAMSPARM(TEST) enhancement:** AMS commands use system adapters UGPOOL, UGSPACE, UGSPC16 to obtain core storage for command processing. Some of the commands such as PARM(TEST) have 31 bit support but still perform below the line, because those system adapters use a macro version of getmain/freemain (MF) that does not actually obtain the storage above the line as expected even though "LOC=ANY" is specified on the call of getmain.
- **REPROMERGEAT output enhancement:** IDCAMS will provide new optional keywords, MESSAGELEVE(ALL|SHORT) for REPRO MERGEAT. These options will provide existing or condensed output listings. ALL is the default.

IDCAMS VERIFY RECOVER ENHANCEMENT: This enhancement will now allow for 3 ways to run the VERIFY command:

- IDCAMS VERIFY: Original way as it is now where IDCAMS opens the data set for output and then issues the VERIFY macro with no options and then closes the data set.
- IDCAMS VERIFY RECOVER: Currently the RECOVER option causes VSAM Record Management VERIFY to back out or complete any interrupted CA reclaim in addition to regular IDCAMS VERIFY functions. There will be no change to this way of running VERIFY.
- EXAMINE/IDCAMS VERIFY RECOVER: This will be a new enhancement that is functional only when EXAMINE and VERIFY RECOVER are run in the same IDCAMS job step. EXAMINE has been enhanced to pass the error information that it finds (such as index CI#, data CI#, error type, and so forth) to VERIFY. As long as there is no concurrent access on the data set, VERIFY will attempt to repair any errors that it can.

Using the Generation Data Group enhancements:

- **Extended format for generation data groups (GDGs):**

z/OS V2R2 introduces a new extended format for generation data groups (GDGs). Extended format GDGs can contain up to 999 generation data sets (GDSes). The previous GDS limit was 255 GDSes per GDG. New GDGs can be defined with this new extended format. For existing GDGs, the previous GDS limit still applies.

To support this enhancement, the IDCAMS DEFINE GDG command includes a new optional parameter (EXTENDED) that you can specify to enable a new GDG to contain up to 999 GDSes. If you do not specify that parameter, the default value (NOEXTENDED) takes effect, setting a limit of 255 GDSes for the GDG.

A new GDGEXTENDED parmlib variable lets you specify whether to allow the EXTENDED value to be used on DEFINE of a GDG. If GDGEXTENDED(NO) (the default) is specified, then the DEFINE of a GDG with the EXTENDED parameter is not allowed. If GDGEXTENDED(YES) is specified, then the DEFINE of a GDG with the EXTENDED parameter is allowed. For more information, see the description of IGGCATxx in *z/OS MVS Initialization and Tuning Reference*.

The LIMIT parameter on the IDCAMS DEFINE GDG command is changed to accept a maximum value of 999 for extended GDGs. The previous maximum LIMIT value of 255 still applies to GDGs which are not defined as EXTENDED.

For extended GDGs, the IDCAMS ALTER LIMIT command is also enhanced to let you set a new GDS limit of up to 999 for the GDG. The z/OS Generic

Tracking Facility has also been used to help determine if any calls to Catalog Management are only requesting the classic GDG limit, and not the extended GDG limit.

For more details about these enhancements, see the descriptions of the ALTER command and the DEFINE GENERATIONDATAGROUP command in *z/OS DFSMS Access Method Services Commands*.

- **GDGSCRATCH parmlib variable:**

z/OS V2R2 introduces a new GDGSCRATCH parmlib variable that specifies whether the default option on DEFINE of a generation data group (GDG) should be SCRATCH or NOSCRATCH. If GDGSCRATCH(NO) (the current default) is specified, then the flag in the catalog GDG record will be set to NOSCRATCH. If GDGSCRATCH(YES) is specified, the flag in the catalog GDG record will be set to SCRATCH. If the NOSCRATCH or SCRATCH keyword is specified on the IDCAMS DEFINE GDG statement, it will be honored regardless of the GDGSCRATCH(YES | NO) IGGCATxx parmlib setting. If GDGSCRATCH(YES | NO) is not in the parmlib, then the existing default of NOSCRATCH is used.

For more details about this enhancement, see the description of the DEFINE GENERATIONDATAGROUP command in *z/OS DFSMS Access Method Services Commands* and the description of IGGCATxx in *z/OS MVS Initialization and Tuning Reference*.

- **New option for GDG management - PURGE:** This enhancement adds a new PURGE option to the DEFINE and ALTER commands. Additionally, it adds a new LISTCST field to print the PURGE option. The PURGE option is valid only when the SCRATCH option is specified. It will override expiration dates when deleting generation data sets (GDSes). If not specified, the default is NOPURGE, but this can be overridden by the new GDGPURGE parmlib variable.

Chapter 2. Using the SMS enhancements

In z/OS V2R2, SMS introduces these enhancements.

SMS Space Constraint Relief

SMS space constraint relief processing provides relief for space-related failures for SMS managed data sets. When space constraint relief is requested, SMS retries an allocation that was unsuccessful due to space constraints on the volumes. Depending on the allocation characteristics, SMS may retry with a one-step or two-step process after the initial phase failed for insufficient space.

In the one-step process, SMS retries the allocation after reducing the requested space quantity based on the Reduce Space Up to (%) attribute. SMS simultaneously removes the 5-extent limit so that SMS can use as many extents as the data set allows. The one-step process is used when allocating a data set that has a volume count of 1, allocating a data set with guaranteed space or extending an existing data set to a new volume.

In the two-step process, SMS first uses a best-fit volume selection method to spread the primary quantity over multiple volumes, up to the volume count, if needed. If this fails due to space constraints, SMS continues with the best-fit method after reducing the primary quantity and removing the 5-extent limit. The two-step process is generally used when the volume count is greater than 1.

z/OS V2R2 provides the following enhancements to SMS space constraint relief:

- Supports a new parameter in the data class that indicates whether space reduction on guaranteed space allocation is permitted or not.
- Supports the space reduction function on non-striped guaranteed space allocations when allocating a new data set or extending an existing data set to a new volume. This space reduction function will remain unsupported for striping allocation as the system requires all stripes to have the same space quantity.
- Allocates the largest possible space that satisfies the percentage specified in the parameter, Reduce Space Up to (%), during space reduction processing for both guaranteed space and non-guaranteed space allocation requests. If the reduced space is not available on the current volume, the secondary extent is allocated on another volume. For VSAM each extent must return space in multiples of the CA size. If the reduced minimum secondary allocation amount falls below the minimum CA size, then the minimum CA size is used as the minimum secondary allocation amount.
- Issues a new SMS message to the job log and hardcopy log when the Dynamic Volume Count (DVC) function is invoked.

For more information, refer to the topic about Specifying attributes to handle space constraints in *z/OS DFSMSdfp Storage Administration*.

SMS RAS enhancements

SMS provides support for the following RAS items:

- SMS issues a RESERVE with the resource name, IGDCDSXS, to serialize the access to SMS control data sets, ACDS and COMMDS. It is recommended that the SMS resource name, IGDCDSXS, be placed in the GRS RESERVE conversion

RNL as a generic entry so it can be converted to the global ENQ. The purpose is to minimize delays due to contention for resources and prevent potential deadlocks. In V2R2, SMS will issue a new informational message, IGD06041I, to the console if IGDCDSXS is not specified in the GRS RESERVE conversion RNL for the system that is to participate in a global resource serialization complex.

- In V2R2, SMS issues IGD030I messages indicating a syntax error when parameters, SELECT and DESELECT, specified in the SMS PARMLIB member extended to a third line. For example:

```
SELECT(ACSINT,ACSPRO,  
       VTOCA,VTOCC,  
       IDAX,CATG,MSG,MODULE) <-- Error on the third line
```

This item is to remove the restriction and allow these SMS PARMLIB parameters to extend beyond the second line.

- ACS messages IGD01012I and IGD01015I 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.
- Prior to V2R2, SMS issued message IGD17800I when the specified volume(s) for a guaranteed space request or the volume provided by AMS for an AIX define can not be found in the eligible storage group list. The storage group(s) searched by SMS is not externalized to the user. To improve problem diagnosis, message IGD17800I is enhanced to display the storage group(s) that are searched.
- The RETENTION LIMIT value in the Management Class limits the use of retention period and expiration date. Data sets that are assigned with a RETENTION LIMIT value of zero days specified in their Management Class are immediately expired with an IGD17364I message issued to the job log. This has caught some users by surprise, and they needed to recover these data sets. To facilitate this task, this item will also externalize IGD17364I to the hardcopy log. This allows the user to identify these expired data sets by searching for the IGD17364I messages in the hardcopy log. Prior to V2R2, the user may need to go through the joblogs to identify these data sets.

User-defined ACS read-only variable

SMS provides the ability to specify user-defined values for use with ACS routines. You use a new parameter in the IGDSMSxx member of PARMLIB, as follows: USER_ACSVAR(*value1,value2,value3*). The values for USER_ACSVAR are saved when SMS is started and, during ACS processing, are passed to the ACS routines in the form of a new ACS read-only variable, &USER_ACSVAR. You can alter the values with the SETSMS command.

For details, refer to the following topics:

- For information about ACS routines, refer to ACS routines in *z/OS DFSMSdfp Storage Administration*.
- For details on the IGDSMSxx member of PARMLIB, refer to IGDSMSxx in *z/OS MVS Initialization and Tuning Reference*.
- For details on the SETSMS command, refer to SETSMS in *z/OS MVS System Commands*.

Storage group space alert messages

SMS provides new alert thresholds for pool storage groups. You can define the thresholds with these attributes:

- Total Space Alert Threshold %
- Track-Managed Space Alert Threshold %

For more information about the attributes, refer to the topic about Values for defining a pool storage group in *z/OS DFSMSdfp Storage Administration*.

New alert messages are issued to the console when the alert thresholds have been reached:

- IGD400I, for total space
- IGD401I, for track-managed space

SMS calculates the space usage of a pool storage group when a space change occurs on an online and enabled volume in the pool storage group or when an enabled volume in the pool storage group is varied online or offline.

A new keyword, ALERT, is added to the DISPLAY SMS command, to display all of the storage groups that have reached one of their alert thresholds. You can use DISPLAY SMS,STORGRP(ALERT) or DISPLAY SMS,STORGRP(ALERT),LISTVOL. When no storage groups have reached the threshold, a new version of message IGD004I is issued.

In addition, the IGD002I messages issued for DFSMS DISPLAY SMS,STORGRP commands include new information about total and track-managed space for each storage group.

For details about the messages, refer to *z/OS MVS System Messages, Vol 8 (IEF-IGD)*

Chapter 3. Using the Open/Close/End of Volume enhancements in z/OS V2R2

z/OS V2R2 includes several enhancements for Open/Close/End of Volume functions, described in this section.

RAS Enhancements: There are several RAS enhancements for Open/Close/End of Volume:

- **DEVSUPxx enhancements:** In previous releases, a subset of DEVSUPxx keywords were reset to their default values whenever a member was processed, unless the values were explicitly set in the member. This was true for some keywords, but not for all. Starting in V2R2, the current values of all keywords are preserved when a DEVSUPxx member is processed, unless the member explicitly specifies new values.

In addition, beginning in V2R2, multiple DEVSUPxx members can be specified on a single SET DEVSUP command invocation, by specifying their two-digit member suffixes separated by commas, for example:

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

Dynamic exits for Open/Close/End of Volume: This enhancement provides dynamic versions of the following Open/Close/End of Volume tape installation exits, which can be changed and put into effect without an IPL. In previous releases, these exits required an IPL for changes to go into effect.

- Volume mount -- dynamic version is OCE_VOLUMEMOUNT
- File start – dynamic version is OCE_FILESTART
- File validate – dynamic version is OCE_FILEVALIDATE
- File end – dynamic version is OCE_FILEEND
- Label anomaly – dynamic version is OCE_LABELANOMALY.

Every exit routine of each dynamic exit receives the same main and function-specific parameter lists. Any changes in parameter lists made by one exit routine will be passed to other exit routines, if multiple exit routines are defined for dynamic exits. Open/Close/End of Volume calls every defined exit routine one at a time. For more information, see the *Tape label processing installation exits* chapter of *z/OS DFSMS Installation Exits*.

Controlling synchronization of files written to tape: Open/Close/End of Volume tape processing provides for potential performance improvements and improved recovery with the following new function:

- Controlling synchronization for files written to tape. Using a new keyword on the SYNC parameter of the DCBE macro, you can specify how many files should be written to tape volumes by a given job, before a synchronization occurs. This function can help improve performance by allowing you to minimize the number of synchronizations being performed to the optimal number, avoiding the overhead and time consumed by unnecessary synchronizations. This new function can also help improve recovery by giving a way to determine the specific compromised files in the event of a synchronize failure.

To use this function, specify the number of files to be written before synchronization in the new SYNC=(NUMFILES,nnn) keyword of the DCBE macro. The job must then pass the DCBE to the OPEN for tape, and that specified number

of files can be written specifying PASS RETAIN or CLOSE LEAVE before an explicit synchronize channel program is written. All the data from those files, including file boundaries (tape marks) is buffered in the device cache; after the specified number of files has been written, an explicit synchronize occurs, which moves the files from cache to the tape medium. If a synchronization failure occurs, message IEC999I reports the file sequence number of the first file that did not get written to the tape medium, so the lost blocks can be clearly identified and recovered. For more information, see the DCBE section of *DFSMS Macro Instructions for Data Sets*.

Chapter 4. Using the Object Access Method (OAM) enhancements

z/OS DFSMS V2R2 provides the following enhancements to the object access method (OAM):

- **System-managed tape enhancements for the TS7700:** Enhancements for Release 3.2 of the TS7700 Virtualization Engine, which include a new tape attach feature for the TS7720, as well as support for 496 devices per distributed library. Support for the added devices also involves doubling the number of allowed subsystems on a scratch allocation request from 253 to 506 and a new DEVSUPxx PARMLIB option (GREATER_253). For more information, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries*.
- The OSREQ application programming interface now allows OAM applications to provide object data buffers in 64-bit addressable virtual storage above the 2G “bar” when storing object data using the OSREQ STORE function or when retrieving object data using the OSREQ RETRIEVE function. These buffers can be used for objects from 1 byte to 2000M for all destinations in the OAM storage hierarchy: DB2 (4K, 32K, LOB), file system, tape, and optical (with the existing restriction of a maximum object size of 256M for optical); this includes the source system handling in OAMplex configurations for objects less than or equal to 50M for optical writes and reads and tape reads that are routed to another system in the OAMplex. The ability to use 64-bit addressable virtual storage buffers above the 2G “bar” on the OSREQ application programming interface can provide virtual storage constraint relief for OAM applications that have difficulty acquiring sufficient virtual storage within a 2G address space.

This new functionality can be used as an alternative to:

- Storing an object up to 2000M in parts with the OSREQ store sequence functions (STOREBEG/STOREPRT/STOREEND) where multiple OSREQ API invocations are required and the maximum size for each part is limited by the practical amount of virtual storage available below the 2G “bar” for each STOREPRT API invocation
- Retrieving an object (or a partial object) with OSREQ RETRIEVE where multiple OSREQ API invocations are required and the maximum size for each retrieve request is limited to 256M

Therefore, this new functionality could also improve the efficiency of the interactions with OAM through the OSREQ application programming interface, because only a single OSREQ API invocation is needed to store an object up to 2000M in size or to retrieve an object up to 2000M in size.

This new functionality includes:

- TSO/E OSREQ command processor changes for STORE and RETRIEVE functions
- OSREQ macro changes for the STORE and RETRIEVE functions and associated new reason codes
- CBROSR2 sample program changes in support of the OSREQ macro changes.

Notes:

1. The OSREQ application programming interface remains AMODE 31.
2. The maximum possible OAM object size remains unchanged at 2000M.

Chapter 5. Using the new VSAM functions

z/OS V2R2 introduces the following enhancements for VSAM:

- VSAM dynamic buffer addition – LSR buffering is enhanced with a function called VSAM dynamic buffer addition, which adds buffers if no buffer is available for a given VSAM request. In previous releases, such a request would fail and you would have the option of retrying the request immediately or waiting for other requests to finish and thus free up buffers. When dynamic buffer addition occurs, VSAM issues the new informational message IDA9990I indicating the number and size of buffers and the shared pool to which they were added. No action is required in response to the message, but for performance reasons the user is encouraged to rebuild the shared pool with BLDVRP, specifying an increase in pool size.

NSR buffering is always built with a sufficient number of buffers, but with the new dynamic buffer addition function, VSAM adds enough buffers to improve I/O efficiency for processing spanned records. This change is transparent and no message is issued. With either NSR or LSR, no message is produced if dynamic buffer addition is not able to add buffers due to mode or storage constraints.

- Feedback code added to message IDA9999I – when the VSAM Auto Dump function attempts but fails to capture a dump, message IDA9999I is issued with new information including a feedback code indicating the VSAM error and the job name. You can use this information, in the absence of a dump, to diagnosis and respond to the error. For reference, feedback codes are listed in *z/OS DFSMSdfp Diagnosis* and *z/OS DFSMS Macro Instructions for Data Sets*.
- VSAM RLS locking at the control area (CA) level, rather than the data set level. This improves the performance of serialization for the following:
 - VSAM key-sequenced data sets (KSDS): control interval (CI) split, CI reclaim, and spanned record processing
 - Variable-length relative record data sets (VRRDS): CI split and CI reclaim processing.

To support this enhancement, new values are added to the SMF type 42 record. For more information, refer to the topic about SMF type 42 in *z/OS MVS System Management Facilities (SMF)*

- VSAM chained I/O for spanned records – For spanned records with NSR, if there are enough extra buffers defined for the data set, VSAM attempts to write all segments of the entire spanned record out to DASD with one single I/O (chained I/O), to improve integrity and I/O performance. For more information, see "Acquiring Buffers" in *z/OS DFSMS Using Data Sets*.

Chapter 6. Using the DADSM/CVAF enhancements

z/OS DFSMS V2R2 provides the following DADSM/CVAF enhancements:

- A new parameter with the OBTAIN macro allows the caller to specify that if the resource is not available, the I/O read should not queue on the resource and wait. For more information, refer to Reading DSCBs from the VTOC Using OBTAIN in *z/OS DFSMSdfp Advanced Services*. A related CVAF status code has been added. For more information, refer to CVSTAT field codes in *z/OS DFSMSdfp Diagnosis*.
- The audit logging of VTOC I/O is improved. For z/OS DASD volumes, the VTOC consists of Data Set Control Blocks (DSCBs) that describe the data sets that reside on the volume. DFSMS software manages the life of a DSCB, but it is also possible for vendor software to update a DSCB. When a DSCB is written, a VTOC audit log now records the DSCB, along with the job name, date and time, and other fields to help identify the activity. The logging is done with a new SMF42 Subtype 27 record. For details, refer to Record Type 42 in *z/OS MVS System Management Facilities (SMF)*.

You can use the NOTYPE keyword of parmlib member SMFPRMxx to disable VTOC audit logging, as follows: NOTYPE(42(27)). For information, refer to SMFPRMxx in *z/OS MVS Initialization and Tuning Reference*.

The XDAP (Execute Direct Access Program) macro instruction now supports an Input/Output Block (IOB) Extension for EXCP I/O. Specifying IOEB=Y indicates that an IOBE is provided to EXCP, with the address of the IOBE in register 0. This allows a VTOC writer using the XDAP macro to identify the application that is updating the VTOC, for the purposes of audit logging of VTOC I/O. For more information, refer to the topic about executing direct access programs in *z/OS DFSMSdfp Advanced Services*.

For a program that builds its own EXCP channel program to update the VTOC, you can identify this activity using the IOBEUSER field in the Input/Output Block (IOB) Extension. This field is saved in the SMF 42 subtype 27 record.

Chapter 7. Using the DFSMShsm enhancements

In z/OS V2R2, DFSMShsm is enhanced with the following new functions:

- **Storage tiers enhancements**

The DFSMShsm MIGRATE command is enhanced to support the following:

- Processing of one or more storage groups. Volumes will be processed in parallel based on the maximum migration task level.
- Class transitions at the volume level, storage group level, and data set level. Only SMS-managed volumes and/or data sets are supported.
- Data set moving at the volume level, storage group level, and data set level.

For more information on the use of storage tiers, see “An overview of storage tiers” on page 62.

For more information on class transitions, see *z/OS DFSMShsm Storage Administration*.

- **Distributed tape processing enhancements**

With these enhancements, administrators have the ability to distribute the processing of dump copies (FRBACKUP DUMP, BACKVOL DUMP, auto-dump) to all DFSMShsm tape resources available in a group of DFSMShsm hosts.

New parameters on the SETSYS COMMONQUEUE command define the group and its members (hosts) in the HSMplex. These members are considered the functional group in the HSMplex. The HOLD, RELEASE, QUERY, CANCEL, and ALTERPRI commands all now support this function. XCF system facilities are used to provide this support.

A MINSTACK parameter is added to both the DEFINE DUMPCLASS and BACKVOL DUMP commands, to specify the preferred minimum number of dump copies that the system should place on a tape volume. This specifies that distributing the work is more important than stacking, even though doing so could use more tape tasks. The existing STACK parameter (now also aliased as MAXSTACK) on both commands continues to specify the preferred maximum number. The LIST DUMPCLASS command is also changed to show the minimum and maximum stack values for both DEFINE and BACKVOL.

In addition, dump copies from different copy pools with the same DUMPCLASS may be placed on the same tape, up to the maximum stack value.

A new UPDTCDs command lets you update specific fields within the control data sets. Specifically, it enables you to update the expiration date of copy pool dump copies.

For more information, see the specific command descriptions in *z/OS DFSMShsm Storage Administration*.

- **Message simplification**

To simplify the analysis of fast replication errors during FRBACKUP, FRRECOV COPYPOOL and copy pool auto dump, DFSMShsm now collects all related DFSMShsm and DFSMSdss messages and records them in a unique fast replication message data set. (These messages also continue to be recorded in their respective logs.) Choose and control this option with the new MESSAGEDATASET(YES|NO HLQ(*hlq*)) parameters on the SETSYS FASTREPLICATION command. On the FRBACKUP and FRRECOV COPYPOOL commands, use a MESSAGEDATASET parameter to specify the date and time used in the message data set name. The message simplification function also

supports distributed tape processing. For more information, refer to SETSYS command in *z/OS DFSMShsm Storage Administration*.

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

Tasks	Procedure that you must perform:
"Administering storage tiers"	<ul style="list-style-type: none">• "Using the MIGRATE command to perform class transitions at the volume level"• "Using the MIGRATE command to perform class transitions at the storage group level" on page 21• "Using the MIGRATE command to perform class transitions at the data set level" on page 21• "Using the MIGRATE command to perform data set moving" on page 21
"Administering distributed tape processing" on page 22	<ul style="list-style-type: none">• "Using the SETSYS COMMONQUEUE function(CONNECT(basename)) command to connect hosts as members of a functional group in an HSMplex" on page 22• "Using the QUERY COMMONQUEUE command to display requests queued or active in the group's common queue (plus other QUERY options)" on page 22• "Using the DEFINE DUMPCLASS or BACKVOL DUMP command to define the minimum and maximum number of dump copies to stack on a tape volume" on page 23• "Using the UPDTCD command to update fields in the control data sets" on page 23• "Using ALTERPRI to change the priority of queued requests in the common queue" on page 23
"Administering message simplification" on page 24	<ul style="list-style-type: none">• "Using the SETSYS FASTREPLICATION command to direct messages to a message data set" on page 24

Administering

The new V2R2 functions require the following administration tasks.

Administering storage tiers

This topic describes the administration tasks for using the storage tiers enhancements in V2R2.

Using the MIGRATE command to perform class transitions at the volume level

When the MIGRATE VOLUME command is issued, in addition to performing migrations, transitions will also be processed. The management class will be used

to determine if each data set on the volume is eligible to be migrated, transitioned, or both. When a data set is eligible for both, it will be migrated.

New keywords BOTH, MIGRATIONONLY, and TRANSITIONONLY are provided for VOLUME processing to indicate that only migrations should be processed, only transitions should be processed, or that both should be processed. The default will be to process both (i.e. the BOTH keyword). The DAYS(0) parameter is also supported.

For more information, refer to MIGRATE command in *z/OS DFSMShsm Storage Administration*.

Using the MIGRATE command to perform class transitions at the storage group level

With z/OS V2R2, a new MIGRATE STORAGEGROUP command is supported. When issued, migrations and/or transitions will be processed. The management class will be used to determine if each data set in the storage group is eligible to be migrated, transitioned, or both. When a data set is eligible for both, it will be migrated.

A list of storage groups can also be specified on the MIGRATE STORAGEGROUP command. If more than 30 storage groups are specified, only the first 30 will be processed. If more than 30 storage groups are specified, an ARC0797I message will be issued and the first 30 storage groups will be processed.

The keywords BOTH, MIGRATIONONLY, and TRANSITIONONLY are provided for STORAGEGROUP processing to indicate that only migrations should be processed, only transitions should be processed, or that both should be processed. The default will be to process both (i.e. the BOTH keyword). The DAYS(0) parameter is also supported.

For more information, refer to MIGRATE command in *z/OS DFSMShsm Storage Administration*.

Using the MIGRATE command to perform class transitions at the data set level

The new optional TRANSITION keyword is now supported for the MIGRATE DATASETNAME command, to indicate that the data set should go through transition processing. Note that TRANSITION is mutually exclusive with the migration-related parameters.

A similar optional TRANSITION keyword is also now supported for both the HMIGRATE user command and the ARCHMIG macro for a data set.

For more information, refer to MIGRATE command in *z/OS DFSMShsm Storage Administration*.

Using the MIGRATE command to perform data set moving

The new optional MOVE keyword is now supported for the MIGRATE DATASETNAME, MIGRATE STORAGEGROUP and MIGRATE VOLUME commands, to indicate that the data set should go through move processing. You might use this to move data from one storage group to another, or from one volume to another within the same storage group. Note that MOVE is mutually exclusive with the migration-related parameters.

A similar optional MOVE keyword is also now supported for both the HMIGRATE user command and the ARCHMIG macro for a data set.

For more information, refer to MIGRATE command in *z/OS DFSMSHsm Storage Administration*.

Administering distributed tape processing

This topic describes the administration tasks for using the distributed tape processing enhancements in V2R2.

Using the SETSYS COMMONQUEUE function(CONNECT(basename)**) command to connect hosts as members of a functional group in an HSMplex**

Use the new DUMP(CONNECT(*basename*)) subparameter on the SETSYS COMMONQUEUE command to specify the XCF functional group name suffix that this host will use for remote dump processing. When specified, this system will be allowed to send and receive dumps to and from other systems in the HSMplex started with the same *basename*.

The DFSMSHsm host that manages all of the dump requests in the common dump queue (CDQ) is called the master scheduler. All dump requests are sent to the master scheduler, and it assigns work to all of the other hosts eligible in the group to process requests from the CDQ. Each group needs a master scheduler to manage CDQ processing for the group; otherwise, dump requests are processed locally.

A system becomes a candidate for being the common dump queue master scheduler when the MASTERSCHEDULERCANDIDATE(YES) subparameter is specified with SETSYS COMMONQUEUE DUMP(CONNECT(*basename*)). (Note that this is the default if not specified at all, so you should specify MASTERSCHEDULERCANDIDATE(NO) if you wish to specifically prevent this system from becoming a master scheduler for this group.)

The first host that connects to the group and is a master scheduler candidate becomes the master scheduler.

To prevent the master scheduler host from selecting a CDQ-connected host to process dumps to tape, use SETSYS MAXDUMPTASKS(0) on hosts that should not be selected for CDQ dump processing. An alternative is to use the HOLD DUMP command, but this also holds all master scheduler duties on a host that is a master scheduler candidate. It is recommended that SETSYS MAXDUMPTASKS(0) be used to configure hosts to not process dumps, while HOLD DUMP be reserved for situations where ongoing dump processing needs to be held for environmental reasons.

For more information, refer to the description of the COMMONQUEUE parameter in SETSYS command in *z/OS DFSMSHsm Storage Administration*.

Using the QUERY COMMONQUEUE command to display requests queued or active in the group's common queue (plus other QUERY options)

Use the QUERY COMMONQUEUE with the DUMP subparameter to display the status of dump requests that are being processed and queued in the common

dump queue of which the host issuing the command is a member. This query will provide a snapshot of all dump activity across all group members.

The QUERY SETSYS command displays the current settings for the common queue function. Also, for QUERY ACTIVE, QUERY WAITING, QUERY REQUEST, and QUERY USER, in a common dump queue environment, only requests from the host that issued the query command are returned.

For more information, refer to QUERY SETSYS command in *z/OS DFSMShsm Storage Administration*.

Using the DEFINE DUMPCLASS or BACKVOL DUMP command to define the minimum and maximum number of dump copies to stack on a tape volume

Use the new MINSTACK parameter on the DEFINE DUMPCLASS and BACKVOL DUMP commands to specify the preferred minimum number of dump copies that the system should place on a tape volume. The existing STACK parameter, which is now also aliased as MAXSTACK, can still be used to specify the preferred maximum number. In comparison to STACK or MAXSTACK, MINSTACK may allow more volumes to be dumped concurrently and thus complete sooner at the expense of using more tapes.

Note that the output from the LIST DUMPCLASS command has been changed accordingly. A new MINSTACK column is now included, and the existing STACK column has been moved.

For more information, see the DEFINE DUMPCLASS and BACKVOL DUMP command descriptions in *z/OS DFSMShsm Storage Administration*.

Using the UPDTCDS command to update fields in the control data sets

You can use the UPDTCDS command to update fields within the control data sets, specifically, the expiration date of copy pool dump copies.

You might first issue the LIST COPYPOOL(*cpname*) command to display the current versions, dump classes and expiration dates for existing dump copies associated with a copy pool. Then, use the UPDTCDS command to change the expiration date for copy pool dump copies in the control data sets. On the UPDTCDS command, you specify the copy pool, version, dump class and new expiration date.

This example updates the expiration date for copy pool COPYPOOL1, version 124, class ONSITE to 2050/12/31.

```
UPDTCDS COPYPOOL(COPYPOOL1) VERSION(124) DUMPEXPIRATION(DUMPCLASS(ONSITE) NEWDATE(2050/12/31))
```

For more information, refer to UPDTCDS command in *z/OS DFSMShsm Storage Administration*.

Using ALTERPRI to change the priority of queued requests in the common queue

For environments where the host is connected to a dump or recover group common queue, you can alter the priority of queued requests by the request number and by the userid that issued the request. This will change the priority of

all requests that are queued in the DUMP or RECOVER group common queue of which the system issuing the command is a member.

For more information, see the ALTERPRI command description in *z/OS DFSMShsm Storage Administration*.

Administering message simplification

This topic describes the administration tasks for using the message simplification enhancements in V2R2.

Using the SETSYS FASTREPLICATION command to direct messages to a message data set

The existing optional FASTREPLICATION parameter on the SETSYS command specifies which copy method to use for data set level recovery, along with other fast replication options. To request that DFSMShsm direct messages for each fast replication backup or recover copy pool command (or auto dump of a copy pool) to a unique fast replication message data set, issue either SETSYS FASTREPLICATION(MESSAGEDATASET HLQ(*hlq*)) or SETSYS FASTREPLICATION(MESSAGEDATASET(YES) HLQ(*hlq*)), where *hlq* will override the default high level qualifier of HSMMSG that will be used for fast replication message data sets. To disable this function, issue FASTREPLICATION(MESSAGDATASET(NO)).

For more information, see the SETSYS FASTREPLICATION command description in *z/OS DFSMShsm Storage Administration*.

Using MESSAGEDATASET with the FRBACKUP and FRRECOV COPYPOOL commands to specify the date and time used in the message data set name

MESSAGEDATASET is an optional parameter that allows you to specify the date and time used in the name of the fast replication command message data set. Specifying the date and time ensures that you can identify the message data set that corresponds with the FRBACKUP request or FRRECOV COPYPOOL request. If you don't specify the date and time with the MESSAGEDATASET parameter, DFSMShsm uses the date and time when the message data set name is generated.

For more information, refer to the FRBACKUP and the FRRECOV command descriptions in *z/OS DFSMShsm Storage Administration*.

Chapter 8. Using the DFSMSrmm enhancements

The functional enhancements available with z/OS V2R2 DFSMSrmm provide you with these benefits:

Enhancements to EXPDT retention method

The DFSMSrmm has been enhanced with these new functions:

- *Specify expiration time for volumes and data sets:* DFSMSrmm now specifies an expiration time, in addition to an expiration date, for volumes and data sets, thus allowing greater control of exactly when a volume or data set is to expire.

The expiration time will be set to the current time whenever RETPD is used in the JCL, as a default, or in ADDDATASET, ADDVOLUME, CHANGEDATASET, or CHANGEVOLUME subcommands. When EXPDT is used instead of RETPD (in JCL or on a DFSMSrmm subcommand), the expiration time will be set to midnight.

In addition, the CHANGEDATASET and CHANGEVOLUME commands have been enhanced to enable the user to specify the time (in *hhmmss* format) that should be used, in addition to the date, when deciding when to expire volumes and data sets.

The ability to specify both the expiration time and date will be helpful to customers using the EXPDT retention method. For example, you might have data sets created shortly before midnight with a default retention period of one day. This new function will allow you to ensure these data sets do not expire in a few minutes and are kept at least for 24 hours.

- *Search volumes and data sets by date/time ranges:* The SEARCHDATASET and SEARCHVOLUME commands have been enhanced to enable the user to specify the time (in *hhmmss* format) in addition to the date when searching for volumes and data sets by the creation date.

The ability to select records by both the creation time and date will be helpful to customers using the EXPDT retention method. For example, the daily night batch processing might create tapes starting from 8 PM and until 5 AM on the next day. This new function will allow you to assign the same expiration date to all of these volumes, even though they were created on different calendar days.

- *Specify that a data set not expire while cataloged:* A new WHILECATALOG option has been added to the EDGRMMxx parmlib member OPTION command and to the CHANGEDATASET TSO subcommand to enable the user to specify either that:

- the dataset will be kept as long as it is cataloged. If the dataset is uncataloged, it will still be kept if the expiration date has not been reached yet.
- the dataset will be kept as long as it is cataloged, but no later than the expiration date.

Part 2. Using New DFSMS Functions in z/OS V2R1

These topics describe how to use new DFSMS functions in z/OS V2R1.

Chapter 9. Using the Advanced Copy Services enhancements

z/OS V2R1 adds the following Advanced Copy Services enhancements:

- Multi-Target Mirror support.

A Multi-Target Mirror environment allows a device to be the primary of more than one PPRC pair. It simplifies recovery scenarios for cascaded configurations, when swapping from the local site to the intermediate site. With Multi-Target support, during a failover to the intermediate site, the intermediate device can become a primary to both the local and the remote, eliminating the need to suspend the intermediate to remote site pair when swapping back to the local site. It provides for the capability of an incremental resynchronization between the two secondary volumes in both planned and unplanned swap scenarios.

For more information, refer to *What is remote copy in z/OS DFSMS Advanced Copy Services*.

- Multiple Incremental FlashCopy targets (Incremental FlashCopy Version 2).

A FlashCopy source can now have more than one Incremental FlashCopy target. This is provided by Incremental FlashCopy Version 2 (V2). The previous support, which limits a FlashCopy source to 1 Incremental FlashCopy target, is now referred to as Incremental FlashCopy Version 1 (V1). For Incremental FlashCopy V2, the required software and microcode must be present.

With Incremental FlashCopy V2, you can request multiple Incremental FlashCopy targets by issuing multiple FRBACKUP FCINCREMENTAL commands, or with the FRBACKUP FCINCREMENTAL(ALL) command, which specifies that an incremental copy is to be created, and that all subsequent copy pool backup copies should also be incremental.

You can cause the system to revert to Incremental FlashCopy V1 by specifying MULTINCRFLC=NO in the DEVSUPxx member of parmlib.

For more information, refer to the topics about Incremental FlashCopy in *z/OS DFSMS Advanced Copy Services*, *FRBACKUP in z/OS DFSMShsm Storage Administration* and *DEVSUPxx in z/OS MVS Initialization and Tuning Reference*.

- Workload-based write pacing.

Workload-based write pacing is an enhanced form of write pacing that exploits MVS workload management (WLM). It uses the importance value on each write I/O to determine the write pacing level to be used in the calculation for how much write pacing delay to inject, if any. For more information, refer to the topic about workload-based write pacing in *z/OS DFSMS Advanced Copy Services*.

- XADDPAIRed primary volumes can remain offline when you issue the XSTART command for restart or the XADDPAIR command for suspended pairs. Utility volumes and secondary volumes must be online when you issue the XADDPAIR command.

For more information, refer to the topic about XRC operational considerations in *z/OS DFSMS Advanced Copy Services*.

Chapter 10. Using the catalog enhancements in z/OS V2R1

z/OS V2R1 includes several enhancements for DFSMS catalogs, described in this section.

VSAM record-level sharing (RLS) catalog updates:

- **VSAM RLS directory only caching:** This enhancement adds new DIRONLY parameter to DATACLAS RLSCFCACHE, which specifies that RLS not cache the data or index part of the VSAM data set in the coupling facility cache structure. See “Set Up VSAM RLS Directory Only Caching” on page 33

Generation data set (GDS) support for PDSE data sets: This enhancement removes the restriction against defining an SMS-managed partitioned data set extended (PDSE) as a generation data set (GDS). Both allocating a PDSE and defining a generation data group with generation data sets, including PDSEs, is unchanged. See “Do Not Allocate a State Data Set as a GDS PDSE” on page 33 and *DEFINE GENERATIONDATAGROUP* in *z/OS DFSMS Access Method Services Commands*.

You must have a system at the z/OS V2R1 level or higher to exploit the ability to define a PDSE as a generation data set. Attempts to define a PDSE as a generation data set on a system below the z/OS V2R1 level fails. In a mixed sysplex environment, systems below the z/OS V2R1 level see PDSE generation data set as a simple generation data set. Note also that DFSMShsm and DFSMSdss are unable to migrate or copy a PDSE generation data set from a z/OS V2R1 or higher system to pre-V2R1 systems. See *Data Set Organization of Generation Data Sets* in *z/OS DFSMS Using Data Sets*.

The LISTCAT ENTRY output is enhanced to indicate when a generation data set is a PDSE by adding the DSNTYPE field with a value of LIBRARY.

New CSI field names: You can now access the following fields using the Catalog Search Interface (CSI): ASSOC, ASSOCSYB, BUFND, BUFNI, HILVLRBA, INDXLVLS, SEQSTRBA, STRNO, and TRACKS.

See the *Field Name Directory* section in *z/OS DFSMS Managing Catalogs*.

JES3 allocation assist tape TS7700: For scratch and specific allocations, this enhancement allows you to use JES3 to direct the allocations to candidate clusters for scratch mounts or to particular distributed library clusters for specific mounts in the TS7700 Virtualization Engine. See “Enable JES3 allocation assist tape TS7700 in DEVSUPxx” on page 33.

Validate and remove an incorrect DEB address from the DEB table with new new PURGE,PURGE=FORCE option on the DEBCHK macro: This function introduces the new PURGE,PURGE=FORCE option for the DEBCHK macro that tells the system to validate and remove an incorrect DEB address from the DEB table. This is used when a DEB is freed, but, for some reason the DEB table was not updated to remove that DEB address from the table. For example:

1. An incorrect length in subpool 230 was FREEMAINed, including one or more DEBs
2. An incorrect address was passed to FREEMAIN, including one or more DEBs.

- DEB storage was incorrectly overlaid, which destroys the next DEB pointer in that DEB. The system follows the DEB chain in the TCB for the terminating task and calls CLOSE for each DEB that the task neglected to close. If a CLOSE fails, the DEB is removed from the TCB DEB chain and from the DEB table. However if it gets a program check while following the DEB chain, it abandons the rest of the DEBs on the current TCB chain.

The new PURGE,PURGE=FORCE option on the DEBCHK macro can prevent these problems by removing the DEB pointer from the DEB table without checking the DCB (or ACB). The caller must be in system key, supervisor state, hold the local lock, and the passed DEB pointer must exist in the DEB table but not represent a valid DEB. See Ensuring Data Security by Validating the Data Extent Block (DEBCHK macro) in *z/OS DFSMSdfp Advanced Services*.

IDCAMS support for large block interface (LBI): This enhancement allows IDCAMS REPRO and PRINT commands to perform on data sets with a blocksize larger than 32K, up to the maximum that the LBI interface supports, if the LBI feature is enabled. The blocksize is still limited to 32K when the LBI feature is not enabled. See the following:

- To enable LBI, see *z/OS DFSMS Using Data Sets*
- For the IDCAMS PRINT and REPRO commands, see *z/OS DFSMS Access Method Services Commands*.

IDCAMS support for record level sharing (RLS): This enhancement provides RLS support for several IDCAMS operations, allowing users to copy, print and backup VSAM data sets while sharing the data set with other applications. A new RLSSOURCE parameter is added to the REPRO, PRINT, and EXPORT commands, and an RLSTARGET parameter is added to the REPRO and IMPORT commands.

- For details on the command updates, see *z/OS DFSMS Access Method Services Commands*.

Catalog contention detection enhancements: The new MODIFY CATALOG,CONTENTION command can be used to specifies a new wait time or action (or both) for one of the reason classes or Catalog resources for which contention detection is available (ALLOCLK, SYSIGGV2, SYSZTIOT, and SYSZVVDS). See *z/OS DFSMS Access Method Services Commands* for information on the MODIFY CATALOG,CONTENTION command.

Generation data group enhancements: You can now specify the order in which the generation data set list is to be returned for data set allocation when the generation data group (GDG) name is supplied on the DD statement. GDG entries can now be returned in either FIFO (oldest GDS defined to the newest GDS) or LIFO (newest GDS defined to the oldest GDS) order for concatenation. See *z/OS DFSMS Access Method Services Commands* for information on the new FIFO and LIFO parameters for the ALTER and DEFINE GENERATIONDATAGROUP commands. Also see *z/OS DFSMS Managing Catalogs* for information on activating the new GDG FIFO function using the IGGCATxx keyword GDGFIFOENABLE.

Catalog alias enhancements:

- IDCAMS now resolves the symbolic related name for an alias to make sure requests are oriented to the correct catalog. Previously, orientation was to the master catalog, which could cause unexpected results. The restriction on the IDCAMS DEFINE ALIAS command that the resolved value for entryname must be a catalog entry that is located in the same catalog that contains the value for

aliasname has been removed. See *z/OS DFSMS Access Method Services Commands* for information on the IDCAMS DEFINE ALIAS command.

- IDCAMS DEFINE ALIAS command will record the alias creation date. This date can be helpful when cleaning up obsolete high level qualifiers. If an alias has no associated data sets, the alias creation date can be used to determine whether this is a new alias for which no data sets have been created yet or this is an obsolete alias that should be deleted.
- IDCAMS will now check when deleting a catalog entry that has an associated alias to verify that the alias is related to the entry being deleted, before deleting the alias record. For example, non-VSAM record A has alias association C, but alias C has association D in its X record. In this case, the alias C should not be deleted when data set A is deleted. This check is done for all non-VSAM, GDS, and UCON records.

The following table lists the overall tasks to perform, to fully use the catalog enhancements.

Tasks	Procedures you must perform:
Administering	<ul style="list-style-type: none">• “Do Not Allocate a State Data Set as a GDS PDSE”• “Enable JES3 allocation assist tape TS7700 in DEVSUPxx”• “Set Up VSAM RLS Directory Only Caching”

Administering

Do Not Allocate a State Data Set as a GDS PDSE

Although you can now define a PDSE as a generation data set, you should not allocate a state data set as a generation data set PDSE, because it might cause unpredictable results and interfere with the use of XRC Performance Monitor (XPM) or other utility programs that access the STATE data set. See *z/OS DFSMS Using Data Sets*.

Enable JES3 allocation assist tape TS7700 in DEVSUPxx

Enable JES3 to direct the allocations to candidate clusters for scratch mounts or to particular distributed library clusters for specific mounts in the TS7700 Virtualization Engine by specifying JES3_ALLOC_ASSIST={YES} in the DEVSUPxx parmlib member. See DEVSUPxx (device support options) in *z/OS MVS Initialization and Tuning Reference*.

Set Up VSAM RLS Directory Only Caching

Using SMS DATACLAS RLSCFCACHE: VSAM RLS allows multiple levels of coupling facility caching for XCF cache structures that are defined in the active SMS configuration. The value of the SMS DATACLAS RLSCFCACHE keyword determines the level of CF caching. Specify new DIRONLY parameter to DATACLAS RLSCFCACHE to specify that RLS not cache the data or index part of the VSAM data set in the coupling facility cache structure. In this case, RLS will use the XCF cache structure to keep track of data that resides in permanent storage (DASD) and in local storage but is not stored in the cache structure itself. This lets each SMSVSAM in the sysplex keep track of whether locally cached copies of the

data are valid and contain the latest changes without actually writing any data to the cache structure. See Using VSAM Record-Level Sharing in *z/OS DFSMS Using Data Sets*.

Using ISMF: To specify that RLS not cache data or index parts of the VSAM data set in the coupling facility cache structure, specify new DIRONLY parameter in the RLS CF Cache Value data class attribute in ISMF. In this case, RLS will use the XCF cache structure to keep track of data that resides in permanent storage (DASD) and in local storage but is not stored in the cache structure itself. This lets each SMSVSAM in the sysplex keep track of whether locally cached copies of the data are valid and contain the latest changes without actually writing any data to the cache structure. See Defining Shareoptions and RLS attributes for data class in *z/OS DFSMSdfp Storage Administration*.

Chapter 11. Using the SMS enhancements

In z/OS V2R1, SMS introduces these enhancements:

Exploiting clusters and extent pools in SMS volume selection: SMS improves volume selection to exploit clusters and extent pools. SMS now prefers volumes that are in the same cluster when:

- Allocating or extending a multi-volume data set if the accessibility attribute is CONTINUOUS or CONTINUOUS PREFERRED
- Allocating the target data set for the data set fast replication function.

When allocating a striped data set, SMS now attempts to allocate the stripes across separate extent pools.

Providing accurate space statistics: You can now use a VARY SMS command to update space statistics in the ACDS for a pool storage group or a DASD volume.

Providing an option to suppress SMS messages: SMS provides a new keyword in the IGDSMSxx PARMLIB member that allows an installation to suppress specific SMS messages.

Providing an option to guarantee that a PDS is created: SMS provides a new keyword in the IGDSMSxx PARMLIB member and SETSMS command that allow you to direct SMS to honor any value for DSNTYPE that specifies that a PDSE is to be allocated, regardless of whether directory blocks have been requested.

Exploiting clusters and extent pools in SMS volume selection

Before z/OS V2R1, SMS, when allocating or extending a multi-volume data set, preferred the candidate volumes in the same storage facility image (SFI) if the storage class accessibility attribute was set to CONTINUOUS or CONTINUOUS PREFERRED. Beginning with z/OS V2R1, SMS prefers volumes that are in the same cluster. Similarly, when allocating the target data set for the data set fast replication function, SMS now prefers volumes that are in the same cluster as the source data set. If it is not possible to honor the preference of volumes in the cluster, SMS reverts to preferring volumes in the same SFI.

Data set fast replication operation is more efficient when the entire data set resides in the same cluster. If you wish SMS to select volumes in the same cluster, you should configure your clusters accordingly.

Before z/OS V2R1, SMS, when allocating a striped data set, allocated the stripes across logical control units (LCUs). SMS now attempts to allocate the stripes across separate extent pools, to ensure more uniform performance. If this is not possible, SMS continues to allocate the stripes across LCUs. If you wish SMS to separate the stripes across extent pools, you should configure your storage groups accordingly.

For details on SMS volume selection, refer to the topic about SMS volume selection in *z/OS DFSMSdfp Storage Administration*.

Providing accurate space statistics

The new form of the VARY SMS,STORGRP|VOLUME command allows you to update space statistics in the ACDS for a pool storage group or a DASD volume. Because of the potential for the command to involve thousands of volumes, no immediate LSPACE is issued in response to the VARY command. Instead, SMS issues LSPACE and updates the space statistics when it is processing a call that requests information about the storage group or volume.

For details on the VARY SMS,STORGRP|VOLUME command, refer to the topic about Changing the SMS Status of a storage group or volume in *z/OS MVS System Commands*.

Providing option to suppress SMS messages

The new SUPPRESS_SMSMSG parameter in PARMLIB member IGDSMSxx provides an option to suppress certain messages. The messages that may be suppressed are IGD17054I, IGD17227I and IGD17395I.

For details on IGDSMSxx, refer to the topic about IGDSMSxx in *z/OS MVS Initialization and Tuning Guide*.

Providing an option to guarantee that a PDS is created

The new HONOR_DSNTYPE_PDSE parameter in PARMLIB member IGDSMSxx provides an option to allow you to ensure that a partitioned data set is created. If HONOR_DSNTYPE_PDSE=YES, then creating a data set using a value for DSNTYPE of LIBRARY or HFS (both of which indicate that a PDSE should be created) results in a partitioned data set being created, regardless of the values for data set organization or directory blocks. If HONOR_DSNTYPE_PDSE=NO (the default), then data set creation is not changed, and the value for DSNTYPE may be ignored, depending on the values for the other attributes.

You can also set the value for HONOR_DSNTYPE_PDSE with the SETSMS command.

For details on IGDSMSxx, refer to the topic about IGDSMSxx in *z/OS MVS Initialization and Tuning Guide*.

For details on the SETSMS command, refer to the topic about SETSMS in *z/OS MVS System Commands*.

Changes to LISTCAT output

To find the various changes to LISTCAT output, see Interpreting LISTCAT Output Listings in *z/OS DFSMS Access Method Services Commands*.

Chapter 12. Using the Open/Close/End of Volume enhancements in z/OS V2R1

z/OS V2R1 includes the following enhancement for Open/Close/End of Volume.

Partial release enhancement: z/OS V2R1 allows more space on multivolume data sets to be released when RLSE is specified in JCL. The Partial Release (RLSE parameter in JCL) requests that the CLOSE output function release unused space. Before V2R1, for a multivolume sequential data set, only unused space on the current volume was released -- space on subsequent volumes was not released. Starting in V2R1, Open/Close/End of Volume releases allocated space for the data set on the subsequent volumes as well.

Chapter 13. Using the PDSE enhancements

z/OS V2R1 adds the following PDSE enhancements:

- Increased PDSE member size limits. The maximum PDSE member size increases from 15 728 639 records to 2 146 435 071 records. The larger size limit applies to PDSEs being accessed with various sets of characteristics, including DSORG and MACRF values, whether they are open for input or output, and whether BLOCKTOKENSIZE=LARGE is specified. For a list of the PDSE access characteristics that support the larger member size limit, see the information on PDSE member size limits in *z/OS DFSMS Using Data Sets*.
- New PDSE version. z/OS V2R1 introduces a new version of PDSE data sets that can provide for improved performance, reduced path lengths, and improved index searches. New data sets can be allocated as belonging to the new version (version 2) by specifying a new positional parameter in the DSNTYPE keyword of the DD statement or TSO/E ALLOCATE command, or by specifying a new PARMLIB option (PDSE_VERSION) in IGDSMSnn members. Unless version 2 is specified, new allocations continue to create the current version 1 PDSE data sets. Externally, version 1 and version 2 PDSEs look the same, and both versions can be open for input/output with no changes for the users. For more information about the new PDSE version and how to specify it, see PDSE Version in *z/OS DFSMS Using Data Sets*.

Chapter 14. Using the Utility enhancements

z/OS V2R1 adds the following DFSMSdfp utility enhancements:

- The IEBCOPY utility is enhanced with user exit capabilities for specifying control statements and for specific member selection.
- IEBCOPY's group copy function is expanded to include PDS to PDS member group copies. A new statement, COPYGROUP, provides the same functions as the existing COPYGRP statement, and expands it to support group copies when both the input and the output data set are PDSs.
- IEBCOPY's SELECT statement also has been enhanced to allow wild card characters in the specification of member names, when used with the COPYGROUP statement. Member name filter pattern masking, using the asterisk (*) and percent (%) characters, allows you to specify a wide range of similar member names. If you use member name filter pattern matching on a SELECT statement with COPYGROUP, you can also code a corresponding EXCLUDE statement with member name filter pattern matching.

The IEBCOPY utility is also enhanced to provide an ABEND code and the associated reason code in a structure returned in register 0, for certain ABENDs. Starting in z/OS V2R1, IEBCOPY returns this information if an ABEND occurs in the FAMS subcomponent. For details, see the IEBCOPY return code information in Appendix A of *z/OS DFSMSdfp Utilities*.

Starting in z/OS V2R1, the IEBCOPY utility supports user exits supplied by the program that invokes IEBCOPY. (User exits are not supported when IEBCOPY is invoked directly using JCL). IEBCOPY supports the following user exits:

- Control statement user exit – if provided by the invoking program, IEBCOPY calls this exit each time IEBCOPY needs to read a control statement. On each call to the exit, the exit can supply one or more control statements, and optionally, lines of text to be printed to SYSPRINT. IEBCOPY calls the control statement exit for each of the following functions:
 - Initialize control statements exit
 - Return control statement
 - End control statements exit.

Note:

1. An input record returned to IEBCOPY may be a SYSIN statement record or a SYSPRINT record report.
 2. If the control statement user exit is provided, IEBCOPY does not use SYSIN or a DD substituted for it.
- Member selection user exit – if a member selection user exit is supplied in the caller's parameter list, IEBCOPY will call the member-selection user exit to substitute all member selection decision logic pertaining to SELECT and EXCLUDE control statements. IEBCOPY will pass each member name to the exit in alphanumeric order. The user exit can choose to include or exclude that name from the current operation.

For complete details on these user exits and their parameter lists, see Appendix A in *z/OS DFSMSdfp Utilities*.

Also in z/OS V2R1, IEBCOPY introduces a new COPYGROUP statement that provides the same function as the COPYGRP statement but also performs that same function for a PDS to PDS copy. (With COPYGRP, when the input and output data sets are both PDSes, the operation is treated as a simple COPY operation, not a group copy.)

Use the COPYGROUP statement to begin a group copy, unload, or load. A group consists of a member and all of its aliases. COPYGROUP treats the group as a single entity.

You can use COPYGROUP to copy a data set in the following situations:

- PDS to PDS
- PDS to PDSE
- PDSE to PDS
- PDSE to PDSE.

You can use COPYGROUP for unloading groups in the following situations:

- PDS to PS
- PDSE to PS

You can use COPYGROUP for loading groups in the following situations:

- PS to PDSE
- PS to PDS

When using the COPYGROUP statement:

- All aliases in a group will be copied with the member or neither the aliases nor the member in a group will be copied.
- There can be only one INDD per copy operation.
- You can use the SELECT statement to selectively copy members. Either the member name or an alias can be specified to copy the member and all of its aliases.
- Do not indicate replace (R) on the SELECT statement.
- The EXCLUDE statement is not supported unless a SELECT MEMBER statement uses pattern filter matching and the EXCLUDE statement does also.

For complete details on using the new COPYGROUP statement, see the IEBCOPY chapter in *z/OS DFSMSdfp Utilities*.

Chapter 15. Using the Object Access Method (OAM) enhancements

z/OS DFSMS V2 provides the following enhancements to the object access method (OAM):

- OAM now supports tape block sizes larger than 32760. A new TAPESDB keyword on the SETOAM statement in the CBROAMxx PARMLIB member can be set to enable larger block sizes. When the first object is written to an OAM tape volume, the maximum block size for the volume is established. If support for larger tape block sizes is enabled, that maximum block size is set to the optimal system-determined block size for the device. Otherwise, a maximum block size of 32760 is used. The maximum block size for all objects written to a tape volume is the maximum block size for the volume established when the first object was written regardless of the current SETOAM TAPESDB setting.
- OAM now provides a new ALLOCRETRYMINUTES keyword on the SETOAM statement in the CBROAMxx PARMLIB member, which can be set to control how long OAM makes retry attempts or to bypass retry processing entirely and issue CBR6400D immediately. This can be used with ATAM (Automated Tape Allocation Manager) processing.
- The minimum object size required to utilize Store Sequence processing (STOREBEG, STOREPRT, STOREEND) has been reduced from the previous limit of 256MB+1 to 50MB+1 for all objects except those being written to an optical volume.
- Previously, when OSMC moved an object to a different management class, existing backup copies of the object were kept. If the object moves to a management class that requires fewer (or no) OAM backup copies, the extra backup copies remained, but were no longer needed and waste storage space. OAM now provides a new BACKUPDELETE keyword on the SETOSMC statement in the CBROAMxx PARMLIB member which can be set to indicate that OSMC should delete all unneeded OAM backup copies when processing an object.
- OAM now provides a new SETTLIB statement for tape library settings. The optional SETTLIB statement and its associated keywords in the CBROAMxx PARMLIB member can be used to override the default behavior for some of the main cartridge entry messages in a system managed tape library environment. The SETTLIB statement and keywords are processed when the OAM address space is started and cannot be updated by operator command. The SETTLIB statement can be used to specify:
 - How OAM will display volume entry ignore messages during cartridge entry processing (DETAIL, SUMMARY, or SUPPRESS).
 - Where OAM will display volume entry ignore messages during cartridge entry processing (on both the console and system log or only on the system log).
 - Where OAM will display successful volume entry messages (CBR3610I) during cartridge entry processing (on both the console and system log or only on the system log).

Chapter 16. Using VSAM enhancements

z/OS V2R1 introduces the following enhancements for VSAM and VSAM RLS:

- The restriction that VSAM data sets must be extended-format, and therefore SMS-managed, in order to use extended addressability, is removed for VSAM linear data sets (LDSes). A new parameter on the IDCAMS ALTER command, EXTENDEDADDRESSABLE, lets you make an existing non-SMS VSAM LDS eligible for extended addressability, so it can grow beyond the four GB size limit of SMS-managed data sets. You can also define a data class for new data sets, with EA set to YES, DSORG set to LS, and Extended Format values set to blanks. Using non-SMS managed VSAM LDSes with extended addressability can help installations migrate their z/OS UNIX version root from HFS to zFS while continuing to use non-SMS managed data packs.
- The restriction that data sets accessed by VSAM RLS could not use dynamic volume count is removed. Dynamic volume count enables the dynamic addition of volumes to a DASD data set without increasing the number of candidate volumes stored in the catalog.
- The SHOWCB macro has added two new sub-parameters, BUFNOL and BUFUSE. These enhancements are used to display fields of an Access Method Control Block (ACB).
- A new data set attribute identifies whether a data set is eligible for VSAM replication.
- Expanded and new keywords for specifying record access bias and ACB RMODE31 values for a data class.

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

Tasks	Procedure that you must perform:
“Administering”	<ul style="list-style-type: none">• “Specifying a value for dynamic volume count” on page 46• “Specifying eligibility for VSAM replication” on page 46• “Specifying VSAM SMB access method bias using a data class” on page 46
“Programming” on page 46	<ul style="list-style-type: none">• “Using SHOWCB enhancements to display an Access Control Block” on page 47• “Using new attributes collected by DCOLLECT” on page 47

Administering

This section describes the administration tasks you must perform to use the new functions.

Specifying a value for dynamic volume count

You specify a value for dynamic volume count for a data class, using ISMF. For more information, refer to the topic about defining volume and data set attributes for a data class in *z/OS DFSMSdfp Storage Administration*.

Specifying eligibility for VSAM replication

You can specify eligibility for VSAM replication with a new data set attribute. You can set the attribute with the:

- IDCAMS DEFINE CLUSTER or ALTER commands
- The Data Class Define or Alter panels in ISMF.

The IDCAMS LISTCAT command displays the value of the attribute with LOGREPLICATE YES or NO in the RLS section for the data set.

For more information about specifying the attribute with IDCAMS commands, refer to the topics about the ALTER command and the DEFINE CLUSTER command in *z/OS DFSMS Access Method Services Commands*.

For more information about specifying the attribute with ISMF, refer to the topic about Specifying attributes for backup-while-open (BWO) and recovery in *z/OS DFSMSdfp Storage Administration*.

Specifying VSAM SMB access method bias using a data class

The Record Access Bias keyword for data class definitions in ISMF is expanded to include all the possible values which could previously be specified using the JCL AMP parameter's ACCBIAS keyword. In addition, the JCL AMP parameter's ACB RMODE31 keyword can now be specified in ISMF data class definitions. You can set these attribute values with the Data Class Define or Alter panels in ISMF.

The expanded list of options in the Record Access Bias keyword in the ISMF Data Class panels includes all the mutually exclusive SMB options (SYSTEM, USER, SO, SW, DO, DW, with a default of USER). Using a data class definition, you can modify these record access bias values without having to edit individual JCL statements, and the modification is no longer limited to one single job step.

The new ACB RMODE31 keyword lets you specify all the same values as the JCL AMP parameter supports: ALL, BUFF, CB, or NONE, with NONE as the default.

The JCL AMP specifications continue to take precedence over the specifications in the data class, and the data class specifications take precedence over the ACB specifications.

For more information about these new data class attributes and about VSAM system-managed buffering in general, see the topic Tuning for System-Managed Buffering in *z/OS DFSMS Using Data Sets*.

Programming

This section describes the programming tasks you must perform to use the new functions.

Using SHOWCB enhancements to display an Access Control Block

You can now specify new options for displaying an Access Control Block and buffers. For more information, refer to the topic SHOWCB—Display fields of an access method control block in *z/OS DFSMS Macro Instructions for Data Sets*.

Using new attributes collected by DCOLLECT

The IDCAMS Data Collection Facility (DCOLLECT) supports eligibility for VSAM replication with new attributes in the Data Class (DC) record. For more information, refer to the topic DCOLLECT in *z/OS DFSMS Access Method Services Commands*.

Chapter 17. Using the zEnterprise Data Compression (zEDC) enhancements

This enhancement allows storage administrations to exploit zEnterprise data compression (zEDC) for non-VSAM compressed format data sets. Unlike previous compression types, no separate dictionary needs to be created, as zEDC compression hides the dictionary in the data stream. A new dictionary starts in each compression unit. The system can decompress the segment as is.

zEDC compression can be requested in the same way that existing types of compression (generic and tailored compression) are requested. The COMPACTON value in data class is used to allocate a data set in compressed format. The type of compression can be specified at the data set level, system level, or both. At the data set level, it can be specified using the data class via the new COMPACTON options of ZR ("zEDC Required") and ZP ("zEDC Preferred"). It can also be specified at the system level via the new options of ZEDC_R ("zEDC Required") and ZEDC_P ("zEDC Preferred") for the COMPRESS parameter in the IGDMSxx member of SYS1.PARMLIB, when the data class COMPACTON value is specified as Y.

System requirements

- To support the creation of zEDC compressed format data sets, the system must support the zEDC function by supporting a minimum level of software and hardware, as follows:
 - z/OS V2R1 operating system and supplied PTFs, as of March 31, 2014.
 - IBM zEnterprise EC12 (with GA2 level microcode) or IBM zEnterprise zBC12.
- To compress data in a zEDC compressed format data set, the following must also be available at the time the data set is being written:
 - zEDC Express Adapters with the March 31, 2014 Firmware MCL release.
 - zEDC Express software feature must be enabled in an IFAPRDxx parmlib member.

To allocate a zEDC compressed format data set, the allocation amount must also meet the existing minimum space requirement for compressed format data sets. The minimum primary space amount must be 5 MB if secondary amount is specified, or 8 MB if a secondary amount is not specified. If the minimum space requirement is not met, the allocation request may result in a non-compressed extended format data set, depending on whether zEDC is preferred or required. Whereas generic and tailored compressed data sets can be defined as extended format version 1 or version 2 data sets, zEDC compressed data sets will be defined as extended format version 2 data sets, regardless of the user's specification. See Processing Extended-Format Sequential Data Sets in *z/OS DFSMS Using Data Sets* for more information.

Allocation processing

When creating a zEDC-compressed format data set, you can specify how the allocation should proceed if the system does not support the zEDC function, as follows:

- “zEDC Required” indicates that the system should fail the allocation request if the zEDC function is not supported by the system, or if the minimum allocation amount requirement is not met.
- “zEDC Preferred” indicates that the system should *not* fail the allocation request, but rather create either a tailored compressed data set if the zEDC function is not supported by the system, or a non-compressed extended format data set if the minimum allocation amount requirement is not met.

SMS allocation processing determines if a data set can be allocated as compressed format. The type of compression to be used for the data set is not determined until the first OPEN for output of the data set. The following two tables summarize the system behavior during SMS allocation processing for a new data set based on system levels and the user's allocation request, the first table on a z/OS V2R1 system, and the second table on a z/OS V1R12 or V1R13 system.

Table 1. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R1)

z/OS Level	z/OS V2R1			
Processor Level	Supports zEDC compression, as per “System requirements” on page 49		Does <i>not</i> support zEDC compression, as per “System requirements” on page 49	
Meets minimum compression space requirements? (5MB primary if no secondary)	Meets space requirement	Does <i>not</i> meet space requirement	Meets space requirement	Does <i>not</i> meet space requirement
“zEDC Required” request	Allocation successful, create as compressed format (extended format v2)	Allocation request fails with IGD17168I	Allocation request fails with IGD17168I	Allocation request fails with IGD17168I
“zEDC Preferred” request	Allocation successful, create as compressed format (extended format v2)	Allocation successful, create as non-compressed extended format	Allocation successful, create as compressed format (extended format v2)	Allocation successful, create as non-compressed extended format

Table 2. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V1R12 or V1R13)

z/OS Level	z/OS V1R12 or V1R13	
Meets minimum compression space requirements? (5MB primary if no secondary)	Meets space requirement	Does <i>not</i> meet space requirement
Data Class COMPACTON option specified as other than N¹	Allocation successful, create as compressed format	Allocation successful, create as non-compressed extended format
Data Class COMPACTON option specified as N, or not specified	Allocation successful, create as non-compressed extended format	Allocation successful, create as non-compressed extended format
¹ On z/OS V1R12 and V1R13, SMS allocation does not differentiate between the different COMPACTON options.		

The first OPEN for output of the data set determines the compression type for the data set based on the data class and PARMLIB specifications, as shown in “Specifying zEDC compression at the system level” on page 53 and “Specifying zEDC compression at the data set level” on page 54. For a zEDC request, it must also determine the level of the system. When running on a multi-system sysplex, it is possible for the data set to be allocated on one system but opened on a different system.

When running in a JES2 environment, this function does not affect how JES2 selects the system on which to run the job.

When running in a JES3 environment, SMS is invoked to identify a list of one or more target systems where the job should be scheduled. The following table identifies how SMS would select systems when the data set is to be allocated with zEDC compression. For new data sets, SMS will look at the requested compression type (required or preferred). For existing data sets, SMS will look at the compression type found in the dictionary token stored in the data set's extended format cell, located in the catalog.

Table 3. SMS Scheduling System Selection for JES3 Environment Based on zEDC Feature

Priority	System Description Based on zEDC Capability
1	System capable of zEDC compression (as per “System requirements” on page 49). Devices are available on the system.
2	System capable of zEDC compression (as per “System requirements” on page 49). No devices currently available on the system, but devices were available during this IPL.
3	System capable of zEDC compression (as per “System requirements” on page 49). No devices currently available on the system, and no devices were available during this IPL.
4	System not capable of zEDC compression (as per “System requirements” on page 49).

Since the systems can be at different z/OS and processor levels, on the first OPEN for output of the data set, OPEN processing must once again determine the level of system before determining the type of compression to use for the data set. The following tables summarize the system behavior during OPEN processing for a new data set based on system levels and the zEDC compression request, the first table on a z/OS V2R1 system, and the second table on a downlevel z/OS V1R12 or V1R13 system.

Table 4. OPEN Allocation Processing Based on System Levels and Allocation Request (z/OS V2R1)

z/OS Level	z/OS V2R1			
Processor Level	Supports zEDC compression, as per "System requirements" on page 49		Does <i>not</i> support zEDC compression, as per "System requirements" on page 49	
DS1COMPR setting¹	DS1COMPR=on	DS1COMPR=off	DS1COMPR=on	DS1COMPR=off
"zEDC Required" request	Create as zEDC compressed format (extended format v2) ⁴	N/A ²	Create as zEDC compressed format (extended format v2) ^{3 4}	N/A ²
"zEDC Preferred" request	Create as zEDC compressed format (extended format v2) ⁴	Create as non-compressed extended format	Create as tailored compressed format (extended format v2) ⁴	Create as non-compressed extended format

¹ DS1COMPR is the flag in the Format-1/Format-8 DSCB that identifies a data set as being compressed format.

² Since the allocation is set to DS1COMPR=off, OPEN will not check the data class for a compression type.

³ Since the allocation was successful for this data set, the system will avoid failing OPEN, but instead create the data set in zEDC compressed format. In this situation, all data will be written non-compressed.

⁴ When the compressed format data set was created at allocation with a specification of zEDC Required or Preferred, SMS allocation created the compressed format data set as an extended format v2 data set. Therefore, even if the compression type is set to tailored compression at OPEN time, it remains extended format v2.

Table 5. OPEN Allocation Processing Based on System Levels and Allocation Request (z/OS V1R12 or V1R13)

z/OS Level	z/OS V1R12 or V1R13	
DS1COMPR setting¹	DS1COMPR=on	DS1COMPR=off
“zEDC Required” request⁴	Create as zEDC compressed format (extended format v2) ³	N/A ²
“zEDC Preferred” request⁴	Create as tailored compressed format	Create as non-compressed extended format

¹ DS1COMPR is the flag in the Format-1/Format-8 DSCB that identifies a data set as being compressed format.

² Since the allocation is set to DS1COMPR=off, OPEN will not check the data class for a compression type.

³ Since the allocation was successful for this data set, the system will avoid failing OPEN, but instead create the data set in zEDC compressed format. In this situation, all data will be written non-compressed.

⁴ Since the values for zEDC compression are not available for the COMPRESS parameter found in the IGDSMSxx member of SYS1.PARMLIB on downlevel systems, only the Data Class COMPACTION option will be used to determine if zEDC compressed format is requested.

For more information

For more information on zEDC compression, see *z/OS MVS Programming: Callable Services for High-Level Languages*.

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

Tasks	Procedure that you must perform:
“Administering”	<ul style="list-style-type: none"> • “Specifying zEDC compression at the system level” • “Specifying zEDC compression at the data set level” on page 54

Administering

This section describes the administration tasks you must perform to use the new functions.

Specifying zEDC compression at the system level

In addition to the existing TAILORED and GENERIC options, you can now use the ZEDC_R (“zEDC Required”) and ZEDC_P (“zEDC Preferred”) options on the COMPRESS parameter found in the IGDSMSxx member of SYS1.PARMLIB.

For more information, see *z/OS MVS Initialization and Tuning Reference*

Specifying zEDC compression at the data set level

In addition to the existing T (tailored) and G (generic) options, you can now use the ZR ("zEDC Required") and ZP ("zEDC Preferred") options when specifying the **Compaction** on the data class. Note that if COMPACTION=Y is specified for a data set, the system level value will be used.

For more information, see *z/OS DFSMSdfp Storage Administration*.

Chapter 18. Using DADSM/CVAF enhancements

The LSPACE macro now allows the caller to have obtained the SYSVTOC ENQ resource on the volume prior to calling LSPACE for that volume. A new ENQHELD keyword on the LSPACE macro specifies whether or not the LSPACE caller's address space has already obtained the SYSVTOC resource.

For more information, refer to the topic about LSPACE in *z/OS DFSMSdfp Advanced Services*. For new diagnostic codes, refer to the topic about DADSM LSPACE return and diagnostic codes in *z/OS DFSMSdfp Diagnosis*.

Chapter 19. Using the DFSMSdss enhancements

z/OS V2R1 introduces the following enhancements for DFSMSdss:

- A RESET keyword has been added to the RESTORE FULL and RESTORE TRACKS commands. It specifies whether the data-set-changed indicator is reset for the data sets on the volume being restored. You can protect the use of RESET with the RESTORE command. In addition, you can now protect the use of RESET with the DUMP command.
- A DEBUG(SMSMSG) option has been added to the CONVERTV command, the COPY command for logical and physical data sets and the RESTORE command for logical and physical data sets. It instructs DFSMSdss to include ACS WRITE statements in the job output.
- The FCCGVERIFY keyword on the CGCREATED command now accepts multiple volume serials.
- The REPLACEUNCONDITIONAL keyword on the RESTORE command now works for physical data sets.
- The RENAMEUNCONDITIONAL keyword on the RESTORE command, which previously worked only on non-VSAM physical data sets, will now work on VSAM physical data sets, as well.
- The ZCOMPRESS keyword on the DUMP command specifies that DFSMSdss writes the dumped data in compressed format to the output medium using zEDC Services.

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

Tasks	Procedure that you must perform:
“Adminstering”	<ul style="list-style-type: none">• “Protecting the use of the RESET keyword” on page 58
“Operating” on page 58	<ul style="list-style-type: none">• “Using the RESET keyword with DFSMSdss commands” on page 58• “Using DEBUG(SMSMSG) with the CONVERTV, COPY and RESTORE commands” on page 58• “Using FCCGVERIFY with the CGCREATED command” on page 58• “Using REPLACEUNCONDITIONAL with the RESTORE command for physical data sets” on page 59• “Specifying zEDC compression for dumps” on page 59

Adminstering

The new V2R1 functions require the following administration tasks.

Protecting the use of the RESET keyword

You protect the use of the RESET keyword with the DUMP command, and the new RESET keyword with the RESTORE command, using profiles in the FACILITY class, shown in Table 6.

Table 6. FACILITY Class Profiles

Keyword	Profile
RESET with DUMP	STGADMIN.ADR.DUMP.RESET
RESET(YES) with RESTORE	STGADMIN.ADR.RESTORE.RESET.YES

Operating

The new V2R1 functions require the following operation tasks.

Using the RESET keyword with DFSMSdss commands

The data-set-changed indicator (DS1DSCHA) in the VTOC indicates whether or not the data set has changed since its last backup. Prior to z/OS V2R1, during full-volume restore, DFSMSdss unconditionally reset (turned off) the data-set-changed indicator for each data set restored to the target volume. During tracks-restore, if any VTOC track was restored, DFSMSdss unconditionally reset the data-set-changed indicator for all data sets on the volume.

With z/OS V2R1, a RESET keyword has been added to the RESTORE FULL and RESTORE TRACKS commands. It allows you to specify whether the data-set-changed indicator is to be reset for the data sets on the volume being restored. If you omit the RESET keyword on the RESTORE command, DFSMSdss resets the data-set-changed indicator only if the RESET keyword was specified on the DUMP command.

The RESET keyword on the DUMP command is not new. It specifies that the data-set-changed indicator in the VTOC is reset for all data sets that are serialized and dumped successfully. This applies to both a full dump and a data set dump operation.

For more information, refer to the topics about the RESTORE command and DUMP command in *z/OS DFSMSdss Storage Administration*.

Using DEBUG(SMSMSG) with the CONVERTV, COPY and RESTORE commands

A new DEBUG(SMSMSG) option has been added to the CONVERTV command, the COPY command for logical and physical data sets and the RESTORE command for logical and physical data sets. It instructs DFSMSdss to display ACS WRITE statements to the job output.

For details, refer to the topics about the CONVERTV command, COPY command and RESTORE command in *z/OS DFSMSdss Storage Administration*.

Using FCCGVERIFY with the CGCREATED command

The FCCGVERIFY keyword of the CGCREATED command now accepts multiple volume serials. FCCGVERIFY specifies the verification volumes that DFSMSdss will use to validate the state of the FlashCopy® Consistency Group before thawing all the volumes.

For details, refer to the topic about the CGCREATED command in *z/OS DFSMSdss Storage Administration*.

Using REPLACEUNCONDITIONAL with the RESTORE command for physical data sets

You can now use the REPLACEUnconditional keyword with the RESTORE command when restoring and renaming physical data sets, with either the RENAME or RENAMEUnconditional keyword. If you specify REPLACEUnconditional and RENAMEUnconditional, DFSMSdss may replace preallocated target data sets with the new name. If a new name data set exists, it will be used to restore to it if it is usable. Preallocated target data sets must have the same characteristics as the source data sets. Otherwise, they will be deemed unusable.

For details, refer to the topic about the RESTORE command in *z/OS DFSMSdss Storage Administration*.

Specifying zEDC compression for dumps

With z/OS V2R1, a new ZCOMPRESS keyword has been added to the DUMP command to specify that DFSMSdss writes the dumped data in compressed format to the output medium using zEDC Services. This decreases the space occupied by the dumped data.

For more information, refer to the DUMP command in *z/OS DFSMSdss Storage Administration*.

For more information on zEDC, see Chapter 17, “Using the zEnterprise Data Compression (zEDC) enhancements,” on page 49.

Chapter 20. Using the DFSMShsm enhancements

In z/OS V2R1, DFSMShsm is enhanced with the following new functions:

- **Storage tiers**

In previous releases, DFSMShsm treated all data in Level 0 (L0) as being in one single tier in the overall storage hierarchy, with no policies to enable automated data movement within that L0 tier. In this release, DFSMShsm is enhanced to move data from one class of devices to another within the L0 hierarchy. This process is called class transition. Because of this complexity, the full overview of this new enhancement is described in detail in “An overview of storage tiers” on page 62.

- **Increased tape limit**

To allow DFSMShsm to migrate and back up larger data sets, the DFSMShsm limit of the number of tapes that a DFSMShsm migration or backup data set can span has been increased from 40 to 254. RECYCLE will now also process connected sets of up to 254 volumes.

The DFSMShsm MCD, MCC, and FSR records have been extended to contain up to 254 volumes. FSR records that list more than 144 tape volsers will be truncated when written to the DFSMShsm log data sets. This will affect the ARCPRLLOG and ARCPEDIT output. The formatted dump of each FSR in the ARCPRLLOG output will include only the portion of the FSR that was written to the log. When the output volsers included in an FSR for RECYCLE have been truncated, the ARCPRLLOG and ARCPEDIT output will display “TOVOL=*****”.

- **Migration subtasking**

A MIGRATIONSUBTASKS(YES | NO) parameter has been added to the SETSYS command. It allows DFSMShsm to run multiple subtasks concurrently under each migration task for primary space management, on-demand migration, and interval migration on level 0 volumes that migrate data sets to ML1 or ML2 volumes.

The ADDITIONALSUBTASKS(*nn*) subparameter allows you to dynamically change the number of additional subtasks that the system can use, running under each migration task. These additional subtasks add to the number of subtasks that the system already uses when the MIGRATIONSUBTASKS parameter is specified. Note that the actual number of total subtasks used can vary. In general, the total migration subtasks used will be lower if a large value is specified with the maximum migration tasks (MAXMIGRATIONTASKS) parameter. Conversely, the total migration subtasks used will be higher, up to 15, if a smaller value is specified with the maximum migration tasks (MAXMIGRATIONTASKS) parameter.

By processing data sets in migration subtasks for a level 0 volume migration task, the aggregate throughput of all the migration tasks is improved.

- **Fast replication enhancements:**

- **Recovering a data set to a new name during fast replication data set recovery**

A NEWNAME(*newdsname*) parameter has been added to the FRRECOV command. It allows DFSMShsm to use a new, fully-qualified data set name for the recovered backup version or dump copy.

- **Recovering a data set to any volume during fast replication data set recovery**

DFSMShsm fast replication data set recovery will no longer be restricted to recovering data sets back to the original volumes. If DFSMShsm is not able to recover a data set to the original volumes, it will instead select the most eligible volumes with the most free space within the storage group. There are no changes to the DFSMShsm commands.

- **FlashCopy consistency groups**

DFSMShsm fast replication backup will now support FlashCopy consistency groups. A new FlashCopy consistency group option will be added to the SMS copy pool definition. If set to 'Yes' for the copy pool, it indicates that the copy pool backup version must be data-consistent. If the FlashCopy consistency group function fails, the FRBACKUP command will be terminated and the new or in-process copy pool backup version will be invalidated. The FlashCopy consistency group option can be used in combination with other FlashCopy options.

For more information, refer to the topic covering Defining copy pools in *z/OS DFSMSdfp Storage Administration*.

- **zEDC compression:**

- **Specifying whether to use zEDC Services on the dump data**

A ZCOMPRESS parameter has been added to the DEFINE DUMPCLASS command. It allows DFSMShsm to specify whether to use zEDC Services on the dump data.

- **Specifying when compression with zEDC should be done**

A ZCOMPRESS parameter has been added to the SETSYS command. It allows DFSMShsm to specify the type of compression used during migration or backup for all data sets.

An overview of storage tiers

DFSMS is enhanced with this new support to provide policy-based data movement between storage classes and storage groups defined within a Primary (L0) Storage hierarchy. This data movement will be performed during the existing space management processing performed by primary space management, on-demand migration, and interval migration, which are performed at the volume level. When a volume is selected for space management processing due to being over threshold, in addition to analyzing each data set on the volume to determine if it is eligible for migration or expiration, these functions will additionally determine if a data set is eligible to be transitioned, based on new management class criteria.

The existing management class *Class Transition Criteria* will now be applied to all data sets, not just data objects managed with OAM. The class transition criteria enable a policy to be assigned which will then determine when a data set should be transitioned, based on its age, the date that it was last referenced, or a periodic value. The class transition criteria are as shown:

CDS Name : USER.MYSCDS
 Management Class Name . . . : MC1

Class Transition Criteria
 Time Since Creation Years :
 Months :
 Days . :
 Time Since Last use Years :
 Months :
 Days . :
 Periodic
 Monthly on Day . . . :
 Quarterly on Day . . . :
 in Month . . . :
 Yearly on Day . . . :
 in Month . . . :

Similar to data set migrations, class transitions will occur during the second phase of volume-level space management processing. The first phase will attempt to free enough space on the volume by expiring and reconnecting data sets. If the first phase does not free enough space on the volume, then the second phase of space management will move eligible data off of the volume. Data sets that are eligible for a class transition will be processed first. If a data set is eligible for both a class transition and migration, then it will be migrated. This prevents the data from being transitioned to another volume, only to be later migrated. After all data sets that are eligible for a class transition have been processed, then standard migration processing will be performed. Class transitions and migrations occur until the volume reaches its low threshold. The DFSMShsm MD exit can be used to bypass the transition of specific data sets.

When a data set is eligible for a class transition, then DFSMShsm invokes the SMS ACS routines to determine how the data set should be transitioned. A new ACS environment of SPMGCLTR is defined to support class transitions. The ACS routines with an ACEROENV of SPMGCLTR may assign any of the following:

Storage Class

When a new storage class is assigned, DFSMShsm will attempt to move the data set to the appropriate storage class. Similar to new allocations, the specified storage class is the “preferred” storage class. If DFSMShsm is unable to move the data set to the new storage class, then the data set will not be transitioned, unless a new storage group has also been specified.

Management Class

When a new management class is assigned to a data set, DFSMShsm will begin using the policies of the newly assigned management class to manage the data set. For example, when a data set is newly allocated, it may be assigned to a management class that specifies that a data set should have two backup copies and that it is not eligible to be migrated. This may be appropriate for the first 180 days of the data sets existence. After 180 days, only a single backup copy is required and the data set is eligible for migration. This may be accomplished by assigning the data set to one management class upon data set creation, and then transitioning that data set to another management class after 180 days from creation.

If the ACS routines indicate that only the management class should be transitioned, then no actual data movement is performed. DFSMShsm just assigns the data set to the new management class.

Storage Group

When a new storage group is assigned, DFSMShsm will attempt to move the data set to the appropriate storage group. Up to fifteen storage groups may be specified. DFSMShsm will attempt to allocate the data set on one of the fifteen storage groups, excluding the storage group to which the data set is already assigned. It is up to the storage administrator to write the ACS routines such that meaningful transitions occur.

Note: The new SMS parameters for class transition are set as follows, and in this specific order:

1. The new **storage class** parameter value is set according to the existing management class parameter and the existing storage group parameter.
2. The new **management class** parameter value is set according to the existing management class parameter and the *new* storage class parameter.
3. The new **storage group** parameter value is set according to the *new* management class parameter and the *new* storage class parameter.

Figure 1 shows sample ACS routines for assigning new classes and groups during a class transition.

Storage Class:

```
IF &ACSENVIR = 'SPMGCLTR' THEN
  SELECT (&STORCLAS)
    WHEN ('SSD')      SET &STORCLAS = 'EASYTIER'
    WHEN ('EASYTIER') SET &STORCLAS = 'SCSATA'
    OTHERWISE        SET &STORCLAS = &STORCLAS
  END
ELSE ...
```

Management Class:

```
IF &ACSENVIR = 'SPMGCLTR' THEN
/* SPACE MANAGEMENT CLASS TRANSITION */
  SELECT (&MGMTCLAS)
    WHEN ('NOML2')   SET &MGMTCLAS = 'ML20K'
    WHEN ('DB2NEW')  SET &MGMTCLAS = 'DB2AGED'
    OTHERWISE        SET &MGMTCLAS = &MGMTCLAS
  END
ELSE ...
```

Storage Group:

```
IF &ACSENVIR = 'SPMGCLTR' THEN
  SELECT (&STORCLAS)
    WHEN ('EASYTIER') SET &STORGRP = 'EASYTIER'
    WHEN ('SCSATA')   SET &STORGRP = 'SGSATA'
    OTHERWISE        SET &STORGRP = &STORGRP
  END
ELSE ...
```

Figure 1. Sample ACS routines for new classes and groups during a class transition

Last successful class transition date (LSCTD)

A new field called the last successful class transition date (LSCTD), will be set and used by DFSMShsm to manage class transitions. DFSMShsm will use this field to determine if a data set has already been transitioned and should not be reprocessed.

- All existing and newly allocated data sets are assigned an LSCTD value of zero

- If the management class is changed during a transition, then the LSCTD is set to zero. This indicates that the data set should be transitioned based on the class transition policy of the newly assigned management class.
- If the management class is not changed during a transition, then the LSCTD is set to the date of the transition. A non-zero date will prevent a data set from a repeating another transition based on creation or last referenced date. A data set with a nonzero transition date is still eligible for a periodic transition.

LISTCAT and DCOLLECT have been updated to include this field.

Class transition movement

When DFSMShsm determines that a data set is eligible to be physically moved for a class transition, due to either the storage class and/or storage group having been reassigned, then DFSMShsm invokes DFSMSdss to physically move the data set. If DFSMSdss cannot allocate the data set to the newly assigned storage group, then the transition will be failed. If the data set is successfully moved, then the original data set will be deleted. After this data movement, the data set retains all of its existing attributes, including its name, and can be immediately accessed. The catalog is updated to reflect the new location of the data set.

Copy technique

The new management class *Transition Copy Technique* setting specifies which copy technique should be used for the class transitions or data set moving. Possible values are:

STANDARD (STD)

specifies that only standard I/O should be used to perform the data movement. This is the default.

FAST REPLICATION PREFERRED (FRP)

specifies that fast replication should be used when possible. If fast replication cannot be used, then it is acceptable to use standard I/O to perform the data movement.

FAST REPLICATION REQUIRED (FRR)

specifies that fast replication is required. If fast replication cannot be used, then the transition fails.

PRESERVE MIRROR PREFERRED (PMP)

specifies that a Metro Mirror primary volume is allowed to become a FlashCopy target and it would be preferable that the Metro Mirror pair does not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, then the FlashCopy operation is still to be attempted. If the intended FlashCopy target volume is not a Metro Mirror primary volume, then the rules for FAST REPLICATION PREFERRED (FRP) are followed.

PRESERVE MIRROR REQUIRED (PMR)

specifies that a Metro Mirror primary volume is allowed to become a FlashCopy target, and the Metro Mirror pair must not go into a duplex pending state as a result of the FlashCopy operation when the target volume is a Metro Mirror primary volume. If the preserve mirror operation cannot be accomplished, then the FlashCopy should not be attempted. If the target volume is not a Metro Mirror primary volume, then the rules for FAST REPLICATION REQUIRED (FRR) are followed.

Note that if a copy technique other than NONE or Preserve Mirror Required is specified, a valid backup copy must exist before the data set is transitioned.

Data set accessibility

In order for a data set to be moved for a class transition, it must be closed and unallocated so that DFSMSdss can serialize on the data set and move it with full data integrity. Since class transitions are desirable for many types of data sets that are always open, the new management class *Serialization Error Exit* setting specifies which action DFSMSdss should take if it cannot initially serialize a data set.

Possible values for this setting are:

NONE

specifies that the transition should fail with no additional action. This is the default. As a default, DFMSHsm does not issue error messages for data sets that fail serialization. If you wish to ensure that these types of error messages are issued, use PATCH .MGCB.+EF BITS(...1....)

DB2

specifies that DB2 should be invoked to close the data set. If there are no DB2 transactions in progress and the data set is successfully closed and unallocated, then the data set will be exclusively serialized. If exclusive access is obtained, then the data set will be transitioned. After the data set has moved, DB2 is reinvoked to allocate and open the data set. If the serialization cannot be obtained or there are active DB2 transactions, then the transition will fail.

Only DB2 objects should be assigned to a management class with this setting.

CICS

specifies that CICS should be invoked to make the data set unavailable for use by CICS and to close all files open to the data set. If these steps are successful, then the data set will be exclusively serialized and then transitioned. After the data set has moved, CICS is reinvoked to enable the CICS files to use the data set and make available the data set to be used by CICS. If the serialization cannot be obtained a second time, then the transition will fail.

Only CICS data sets should be assigned to a management class with this setting.

ZFS

specifies that zFS should be invoked to unmount the data set. If the data set is successfully unmounted, then the data set will be exclusively serialized. The unmount will fail if the file system is currently accessing the data set. If exclusive access is obtained, then the data set will be transitioned. After the data set has moved, zFS is reinvoked to mount the data set.

Only zFS data sets should be assigned to a management class with this setting.

EXIT

specifies that a user exit should be invoked to unserialize the data set. The exit will be invoked twice: initially to unserialize the data set, and a second time after the transition in order to reserialize the data set. The transition will be performed if the data set can be exclusively serialized after the user exit has been initially invoked. A valid exit should be in place before this option is specified. No default exit is provided.

Storage group processing priority

Class transition processing moves data sets from one tier of L0 volumes to another. For this strategy to work, enough available space must exist on the L0 volumes that are the targets of the class transition processing. For this reason, DFSMS has

provided a new storage group processing priority setting. The value of this priority is from 1 to 100, with a default value of 50. For space management processing, DFSMShsm will process storage groups in the order of their processing priority. Storage groups that will have data sets moved to them as a part of class transition processing should have a relatively higher processing priority such that free space is created for the transitions.

Recall

A data set may miss one or more class transitions while it is migrated. Before recalling a data set, DFSMShsm determines if a class transition was missed. If a class transition was missed, then DFSMShsm invokes the ACS routines with ACS Environment SPMGCLTR to determine any new SMS construct names. Using the newly retrieved management class name, DFSMShsm will repeat the checking to see if a class transition was missed until the appropriate management, storage class, and storage group(s) are determined. DFSMShsm will perform the recall using these values.

Reporting

The DFSMShsm REPORT function and QUERY STATISTICS output have been enhanced to include basic class transition statistics, such as the number of data sets that transitioned.

A new DFSMShsm FSR record type of 24 will be created for each data set transition. These FSR records may be used to analyze DFSMShsm class transition processing. A new class transition related sample report for the Report Generator has been provided.

Related tasks

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

Tasks	Procedure that you must perform:
“Planning and Installation”	<ul style="list-style-type: none">“Planning to use the DFSMShsm enhancements” on page 68“Planning to use storage tiers” on page 68
“Administering” on page 69	<ul style="list-style-type: none">“Administration tasks for storage tiers” on page 69
“Operating” on page 69	<ul style="list-style-type: none">“Operation tasks for storage tiers” on page 70“Using the RECOVERRESET keyword with the DFSMShsm DEFINE DUMPCLASS command” on page 71“Operation tasks for zEDC compression” on page 71

Planning and Installation

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

Planning to use the DFSMShsm enhancements

Before you begin: Be familiar with information about the following topics:

- Space management of SMS-managed storage; see *z/OS DFSMShsm Storage Administration*.

Software requirements: These enhancements require z/OS V2R1 and supplied PTFs.

Hardware requirements: None.

Coexistence requirements: APARS OA36576 and OA37582.

Planning to use storage tiers

The first step for implementing class transition processing in your environment is to determine which data sets would benefit from this type of processing. Two possible candidates are:

- Data sets that are not eligible for migration because they always need to reside on L0 storage. These data sets could be initially allocated on a particular class of storage and then transitioned to a lower class of storage using class transitions.
- Data sets that are currently eligible for migration in order to reduce the cost of storing them, but would improve application performance if they could be kept online. The management class for these data sets could be changed such that instead of migrating the data after a certain period of time, they are transitioned to lower cost L0 storage.

For example, say that you have Easy Tiers devices and SATA devices. Data sets are initially allocated to an Easy Tiers device that provides real time data placement on the appropriate storage tier based on usage. You know that after 45 days, however, certain data set types are rarely accessed and you do not want this set of data mixed with your active data. This set of data could be assigned to a management class that after 45 days will automatically move the data sets from a storage group comprised of Easy Tiers devices to a storage group comprised of just SATA devices.

The next step would be to update the SMS environment to support class transitions:

- Storage devices with differing characteristics must be segregated by storage class and storage group. For example, Easy Tiers devices must be assigned to different storage groups than SATA devices. If the devices are placed within the same storage groups, then SMS is unable to distinguish the appropriate device that is desired. SSD devices may reside within the same storage group since the storage class can be used to specify that an SSD device is preferred over another type of storage.
- ACS routines must be written for the new ACS environment SPMGCLTR. Determine the actions you wish to occur for the class transition and write the appropriate routine. For example, if you wish the class transition to move the data set from an EASYTIER storage group to a SATA storage group, you could define an ACS routine as seen in the “**Storage Group:**” section of Figure 1 on page 64.
- Define or update management classes with the new class transition characteristics. Remember that only the appropriate data set type should be assigned to management classes with a SERIALIZATION EXIT value other than NONE.

Note that no changes are required to the DFSMShsm space management functions. They will automatically examine management classes for the class transition settings and process data sets accordingly. If you expect class transition processing to increase the amount of work done during space management, then you may wish to increase the number of tasks that are assigned to that processing.

Administering

The new V2R1 functions require the following administration tasks.

Administration tasks for storage tiers

This topic describes the administration tasks for using storage tiers.

Specifying class transition processing for interval migration and on-demand migration

By default, interval migration and on-demand migration will perform class transition processing. If this is *not* desired, specify:

```
SETSYS CLASSTRANSTION(EVENTDRIVENMIGRATION(N))
```

Specifying the SERIALIZATION EXIT setting for interval migration and on-demand migration

By default, the management class SERIALIZATION EXIT setting is ignored for interval migration and on-demand migration. This prevents data sets from being closed during prime processing hours. If you wish the SERIALIZATION EXIT to be honored during these functions, then specify:

```
SETSYS CLASSTRANSTION(EVENTDRIVENMIGRATION(Y SERIALIZATIONEXIT(Y)))
```

Using the MIGRATIONSUBTASKS parameter with the DFSMShsm SETSYS command

With z/OS V2R1, a MIGRATIONSUBTASKS(YES | NO) parameter has been added to the SETSYS command. It allows DFSMShsm to run multiple subtasks concurrently under each migration task for primary space management, on-demand migration, and interval migration on level 0 volumes that migrate data sets to ML1 or ML2 volumes.

The ADDITIONALSUBTASKS(*nn*) subparameter allows you to dynamically change the number of additional subtasks that the system can use, running under each migration task. These additional subtasks add to the number of subtasks that the system already uses when the MIGRATIONSUBTASKS parameter is specified. Note that the actual number of total subtasks used can vary. In general, the total migration subtasks used will be lower if a large value is specified with the maximum migration tasks (MAXMIGRATIONTASKS) parameter. Conversely, the total migration subtasks used will be higher, up to 15, if a smaller value is specified with the maximum migration tasks (MAXMIGRATIONTASKS) parameter.

For more information, refer to the topic covering the SETSYS command in *z/OS DFSMShsm Storage Administration*.

Operating

The new V2R1 functions require the following operation tasks.

Operation tasks for storage tiers

This topic describes the operating tasks for using storage tiers.

Tracking Class Transition Processing

QUERY STATISTICS and REPORT output can be used to track, at a high level, how much class transition processing is occurring. For detailed analysis of class transition processing, it is recommended that the DFSMS Report Generator be used to analyze the new DFSShsm FSR type 24 records. A sample report has been provided.

Message ARC0734I has a new insert of 'CLASS-TR' to indicate that the action was a class transition.

Using the existing primary space management, interval migration, and on-demand migration processing commands

Class transition processing occurs within the existing primary space management, interval migration and on-demand migration processing. The existing commands that currently apply to these functions (HOLD, RELEASE, QUERY, SETSYS, etc.) also apply during the new class transition processing of these functions.

Operation tasks for serviceability and usability enhancements

This topic describes the operating tasks for using the serviceability and usability enhancements.

Using the RECYCLETAKEAWAYRETRY(YES | NO) parameter with DFSShsm SETSYS command

With z/OS V2R1, a RECYCLETAKEAWAYRETRY(YES | NO) parameter has been added to the SETSYS command. It allows DFSShsm to automatically generate a new RECYCLE command for a tape when the original recycle must terminate due to the takeaway process, or when the tape is in use by another DFSShsm task. Two additional subparameters, MAXRETRYATTEMPTS(nn) and DELAY(ssss), allow you to set the maximum number of recycle retry attempts and to set the delay interval in seconds between recycle attempts, respectively.

For more information, refer to the topic covering the SETSYS command in *z/OS DFSShsm Storage Administration*.

Using the SELECT(RECYCLETAKEAWAY) subparameter with the DFSShsm LIST command

With z/OS V2R1, a SELECT(RECYCLETAKEAWAY) subparameter has been added to the LIST TAPETABLEOFCONTENTS command. It displays the volumes that were not completely recycled because they were taken away by recall or another DFSShsm task.

For more information, refer to the topic covering the LIST command in *z/OS DFSShsm Storage Administration*.

Using the (TAPECOPY | RECYCLE) subparameter with the DFSShsm SETSYS command

With z/OS V2R1, a (TAPECOPY | RECYCLE) subparameter has been added to both the BACKUP and the MIGRATION parameters of the SETSYS DUPLEX command. It specifies whether, after an error occurs on the duplex alternate tape and the alternate tape is demounted and discarded so that DFSShsm can continue to write to the original tape, a tapecopy or a recycle will be immediately attempted.

For more information, refer to the topic covering the SETSYS command in *z/OS DFSMShsm Storage Administration*.

Using the RECOVERRESET keyword with the DFSMShsm DEFINE DUMPCLASS command

With z/OS V2R1, a RECOVERRESET keyword has been added to the DEFINE DUMPCLASS command. It allows you to specify whether or not the data-set-changed indicator in the VTOC is reset for all data sets that are restored during full-volume recover processing.

For more information, refer to the topic about the DEFINE DUMPCLASS command in *z/OS DFSMShsm Storage Administration*.

The enhancement to the DEFINE DUMPCLASS command is consistent with a related enhancement to DFMSDss. For more information, refer to Chapter 19, “Using the DFMSDss enhancements,” on page 57.

Using the NEWNAME parameter with the DFSMShsm FRRECOV command

With z/OS V2R1, a NEWNAME(*newdsname*) parameter has been added to the FRRECOV command. It allows DFSMShsm to use a new, fully-qualified data set name for the recovered backup version or dump copy.

For more information, refer to the topic covering the FRRECOV command in *z/OS DFSMShsm Storage Administration*.

Operation tasks for zEDC compression

This topic describes the operating tasks for using zEDC compression.

Using the ZCOMPRESS parameter with the DFSMShsm DEFINE DUMPCLASS command

With z/OS V2R1, a ZCOMPRESS parameter has been added to the DEFINE DUMPCLASS command. It allows DFSMShsm to specifying whether to use zEDC Services on the dump data.

For more information, refer to the topic about the DEFINE DUMPCLASS command in *z/OS DFSMShsm Storage Administration*.

For more information on zEDC, see Chapter 17, “Using the zEnterprise Data Compression (zEDC) enhancements,” on page 49.

Using the ZCOMPRESS parameter with the DFSMShsm SETSYS command

With z/OS V2R1, a ZCOMPRESS parameter has been added to the SETSYS command. It allows DFSMShsm to specifying the type of compression used during migration or backup for all data sets.

For more information, refer to the topic covering the SETSYS command in *z/OS DFSMShsm Storage Administration*.

For more information on zEDC, see Chapter 17, “Using the zEnterprise Data Compression (zEDC) enhancements,” on page 49.

Chapter 21. Using the DFSMSrmm enhancements

The functional enhancements available with z/OS V21R1 DFSMSrmm provide you with these benefits:

- **DFSMSrmm operational enhancements**

- *Retain data sets based on the number of days since they were last referenced:* Data sets managed by the EXPDT retention method can be retained or expired based on the number of days since the data set was last referenced. A new *LastReferenceDays* attribute is added to the data set record as a binary number. The value is taken from the new LASTREF suboperand of the DFSMSrmm parmlib OPTION RM(EXPDT) operand.

If the volume set is retained by SET or VOLUME the LastReferenceDays data set attribute will be kept equal for all files of a multi-volume data set. The latest *LastReferenceDays* update to a single file in a multivolume data set is propagated to all files that belong to the same multi-volume data set.

However, for volume set retained by FIRSTFILE the *LastReferenceDays* data set attribute will be set but not equalized across the multi-volume data set because the expiration date depends only of the first file of the first volume, and its *LastReferenceDays*.

The *LastReferenceDays* value can be:

- Set by the ADDDATASET subcommand when the data set record is created
- Changed by the CHANGEDATASET subcommand any time after the data set record has been created.
- *EXPDT retention method now allows retention to be based on a single volume or volume set, or on a controlling first file:* You now have the option of retaining volumes with the EXPDT retention method based on a single volume or volume set, or on a controlling first file. The parmlib OPTION RM(EXPDT(RETAINBY(VOLUME/SET/FIRSTFILE))) can specify:

RETAINBY(VOLUME)

DFSMSrmm expires volumes in a multi volume set at the volume level. Each volume has its own expiration date. RETAINBY(VOLUME) is the default value.

RETAINBY(SET)

DFSMSrmm expires volumes in a multi volume set at the volume set level. All volumes in the set have the same expiration date, which is the maximal expiration date of all volumes (except if changed by the CV command).

RETAINBY(FIRSTFILE)

DFSMSrmm expires volumes in a multi volume set at the volume set level. All volumes in the set have the same expiration date, which is the expiration date of the first file in the volume set (here a single volume is treated as a volume set with only one volume in it).

You can also set the RETAINBY for a specific volume set by subcommand.

Note: These new options apply only to the EXPDT retention method, not to the VRSEL expiration method. The processing of volume sets managed by the VRSEL retention method is unchanged.

- *DFSMS Management Class attributes:* You can now set the expiration date in DFSMSrmm for a tape data set with a DFSMS Management Class (MC). When you enable use of MC attributes by DFSMSrmm, the MC expiration attributes (except the MC Expiration attribute Retention limit) are retrieved by DFSMSrmm during OPEN for output and used to set the expiration date for the tape data set, and also to set the LASTREF extra days in the tape data set record on retention method EXPDT managed volumes. Regardless of whether the expiration attributes are retrieved from MC, from a DFSMSrmm default parmlib option or from an installation exit, an expiration date is calculated and will be used to manage expiration. Any attributes needed to continue management of retention, such as 'days non-usage' are bound to the data set record in the DFSMSrmm CDS, depending on the retention method. This is a one-time action, thus avoiding any overhead of repeating the policy decisions as part of inventory management. At OPEN for input the MC attributes are not considered for processing. At OPEN for output with Disposition MOD the MC attributes are not considered for processing. The enablement in DFSMSrmm provides options to use or not use these MC attributes for all volumes, and it provides an option for VRSEL managed volumes to exclude the MC attribute 'Expire after Date/Days'. This last option is recommended if it is desired that the processing of VRSEL managed volumes not change with DFSMSrmm V2R1.

The Management Class expiration attributes processed by DFSMSrmm are:

- Expire after days Non-usage, which is equivalent to LASTREF extra days
- Expire after Date/Days, which is equivalent to expiration date / retention period.

Part 3. Appendixes

Appendix. Accessibility

Accessible publications for this product are offered through IBM Knowledge Center (<http://www.ibm.com/support/knowledgecenter/SSLTBW/welcome>).

If you experience difficulty with the accessibility of any z/OS information, send a detailed message to the "Contact us" web page for z/OS (<http://www.ibm.com/systems/z/os/zos/webqs.html>) or use the following mailing address.

IBM Corporation
Attention: MHVRCFS Reader Comments
Department H6MA, Building 707
2455 South Road
Poughkeepsie, NY 12601-5400
United States

Accessibility features

Accessibility features help users who have physical disabilities such as restricted mobility or limited vision use software products successfully. The accessibility features in z/OS can help users do the following tasks:

- Run assistive technology such as screen readers and screen magnifier software.
- Operate specific or equivalent features by using the keyboard.
- Customize display attributes such as color, contrast, and font size.

Consult assistive technologies

Assistive technology products such as screen readers function with the user interfaces found in z/OS. Consult the product information for the specific assistive technology product that is used to access z/OS interfaces.

Keyboard navigation of the user interface

You can access z/OS user interfaces with TSO/E or ISPF. The following information describes how to use TSO/E and ISPF, including the use of keyboard shortcuts and function keys (PF keys). Each guide includes the default settings for the PF keys.

- *z/OS TSO/E Primer*
- *z/OS TSO/E User's Guide*
- *z/OS V2R2 ISPF User's Guide Vol I*

Dotted decimal syntax diagrams

Syntax diagrams are provided in dotted decimal format for users who access IBM Knowledge Center with a screen reader. In dotted decimal format, each syntax element is written on a separate line. If two or more syntax elements are always present together (or always absent together), they can appear on the same line because they are considered a single compound syntax element.

Each line starts with a dotted decimal number; for example, 3 or 3.1 or 3.1.1. To hear these numbers correctly, make sure that the screen reader is set to read out

punctuation. All the syntax elements that have the same dotted decimal number (for example, all the syntax elements that have the number 3.1) are mutually exclusive alternatives. If you hear the lines 3.1 USERID and 3.1 SYSTEMID, your syntax can include either USERID or SYSTEMID, but not both.

The dotted decimal numbering level denotes the level of nesting. For example, if a syntax element with dotted decimal number 3 is followed by a series of syntax elements with dotted decimal number 3.1, all the syntax elements numbered 3.1 are subordinate to the syntax element numbered 3.

Certain words and symbols are used next to the dotted decimal numbers to add information about the syntax elements. Occasionally, these words and symbols might occur at the beginning of the element itself. For ease of identification, if the word or symbol is a part of the syntax element, it is preceded by the backslash (\) character. The * symbol is placed next to a dotted decimal number to indicate that the syntax element repeats. For example, syntax element *FILE with dotted decimal number 3 is given the format 3 * FILE. Format 3* FILE indicates that syntax element FILE repeats. Format 3* * FILE indicates that syntax element * FILE repeats.

Characters such as commas, which are used to separate a string of syntax elements, are shown in the syntax just before the items they separate. These characters can appear on the same line as each item, or on a separate line with the same dotted decimal number as the relevant items. The line can also show another symbol to provide information about the syntax elements. For example, the lines 5.1*, 5.1 LASTRUN, and 5.1 DELETE mean that if you use more than one of the LASTRUN and DELETE syntax elements, the elements must be separated by a comma. If no separator is given, assume that you use a blank to separate each syntax element.

If a syntax element is preceded by the % symbol, it indicates a reference that is defined elsewhere. The string that follows the % symbol is the name of a syntax fragment rather than a literal. For example, the line 2.1 %OP1 means that you must refer to separate syntax fragment OP1.

The following symbols are used next to the dotted decimal numbers.

? indicates an optional syntax element

The question mark (?) symbol indicates an optional syntax element. A dotted decimal number followed by the question mark symbol (?) indicates that all the syntax elements with a corresponding dotted decimal number, and any subordinate syntax elements, are optional. If there is only one syntax element with a dotted decimal number, the ? symbol is displayed on the same line as the syntax element, (for example 5? NOTIFY). If there is more than one syntax element with a dotted decimal number, the ? symbol is displayed on a line by itself, followed by the syntax elements that are optional. For example, if you hear the lines 5 ?, 5 NOTIFY, and 5 UPDATE, you know that the syntax elements NOTIFY and UPDATE are optional. That is, you can choose one or none of them. The ? symbol is equivalent to a bypass line in a railroad diagram.

! indicates a default syntax element

The exclamation mark (!) symbol indicates a default syntax element. A dotted decimal number followed by the ! symbol and a syntax element indicate that the syntax element is the default option for all syntax elements that share the same dotted decimal number. Only one of the syntax elements that share the dotted decimal number can specify the ! symbol. For example, if you hear the lines 2? FILE, 2.1! (KEEP), and 2.1 (DELETE), you know that (KEEP) is the

default option for the FILE keyword. In the example, if you include the FILE keyword, but do not specify an option, the default option KEEP is applied. A default option also applies to the next higher dotted decimal number. In this example, if the FILE keyword is omitted, the default FILE(KEEP) is used.

However, if you hear the lines 2? FILE, 2.1, 2.1.1! (KEEP), and 2.1.1 (DELETE), the default option KEEP applies only to the next higher dotted decimal number, 2.1 (which does not have an associated keyword), and does not apply to 2? FILE. Nothing is used if the keyword FILE is omitted.

*** indicates an optional syntax element that is repeatable**

The asterisk or glyph (*) symbol indicates a syntax element that can be repeated zero or more times. A dotted decimal number followed by the * symbol indicates that this syntax element can be used zero or more times; that is, it is optional and can be repeated. For example, if you hear the line 5.1* data area, you know that you can include one data area, more than one data area, or no data area. If you hear the lines 3* , 3 HOST, 3 STATE, you know that you can include HOST, STATE, both together, or nothing.

Notes:

1. If a dotted decimal number has an asterisk (*) next to it and there is only one item with that dotted decimal number, you can repeat that same item more than once.
2. If a dotted decimal number has an asterisk next to it and several items have that dotted decimal number, you can use more than one item from the list, but you cannot use the items more than once each. In the previous example, you can write HOST STATE, but you cannot write HOST HOST.
3. The * symbol is equivalent to a loopback line in a railroad syntax diagram.

+ indicates a syntax element that must be included

The plus (+) symbol indicates a syntax element that must be included at least once. A dotted decimal number followed by the + symbol indicates that the syntax element must be included one or more times. That is, it must be included at least once and can be repeated. For example, if you hear the line 6.1+ data area, you must include at least one data area. If you hear the lines 2+, 2 HOST, and 2 STATE, you know that you must include HOST, STATE, or both. Similar to the * symbol, the + symbol can repeat a particular item if it is the only item with that dotted decimal number. The + symbol, like the * symbol, is equivalent to a loopback line in a railroad syntax diagram.

Notices

This information was developed for products and services offered in the U.S.A. or elsewhere.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan, Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

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

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

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

Site Counsel
IBM Corporation
2455 South Road
Poughkeepsie, NY 12601-5400
USA

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

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

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.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

COPYRIGHT LICENSE:

This information might contain 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Policy for unsupported hardware

Various z/OS elements, such as DFSMS, HCD, JES2, JES3, and MVSTM, contain code that supports specific hardware servers or devices. In some cases, this device-related element support remains in the product even after the hardware devices pass their announced End of Service date. z/OS may continue to service element code; however, it will not provide service related to unsupported hardware devices. Software problems related to these devices will not be accepted.

for service, and current service activity will cease if a problem is determined to be associated with out-of-support devices. In such cases, fixes will not be issued.

Minimum supported hardware

The minimum supported hardware for z/OS releases identified in z/OS announcements can subsequently change when service for particular servers or devices is withdrawn. Likewise, the levels of other software products supported on a particular release of z/OS are subject to the service support lifecycle of those products. Therefore, z/OS and its product publications (for example, panels, samples, messages, and product documentation) can include references to hardware and software that is no longer supported.

- For information about software support lifecycle, see: IBM Lifecycle Support for z/OS (<http://www.ibm.com/software/support/systemsz/lifecycle/>)
- For information about currently-supported IBM hardware, contact your IBM representative.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available at Copyright and Trademark information (<http://www.ibm.com/legal/copytrade.shtml>).

Glossary

This glossary defines technical terms and abbreviations used in DFSMS documentation. If you do not find the term you are looking for, refer to the index of the appropriate DFSMS manual or view the *Glossary of Computing Terms* located at:

<http://www.ibm.com/ibm/terminology/>

This glossary includes terms and definitions from:

- The *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- The *Information Technology Vocabulary* developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1/SC1). Definitions of published part of this vocabulary are identified by the symbol (I) after the definition; definitions taken from draft international standards, committee drafts, and working papers being developed by ISO/IEC JTC1/SC1 are identified by the symbol (T) after the definition, indicating that final agreement has not yet been reached among the participating National Bodies of SC1.
- The *IBM Dictionary of Computing*, New York: McGraw-Hill, 1994.

The following cross-reference is used in this glossary:

See: This refers the reader to (a) a related term, (b) a term that is the expanded form of an abbreviation or acronym, or (c) a synonym or more preferred term.

A

access method services

A multifunction service program that manages both VSAM and non-VSAM data sets and integrated catalog facility catalogs or VSAM catalogs. It defines data sets and allocates space for VSAM data

sets, VSAM catalogues, and ICF catalogs. It converts indexed-sequential (ISAM) data sets to key-sequenced data sets, modifies data set attributes in the catalog, reorganizes data sets, facilitates data portability among operating systems, creates backup copies of data sets and indexes, helps make inaccessible data sets accessible, lists the records of data sets and catalogs, defines and builds alternate indexes, and converts CVOLs and VSAM catalogs to integrated catalog facility catalogs.

access permission

A group of designations that determine who can access a particular AIX or UNIX file and how the user can access the file.

ACS

Automatic class selection.

active control data set (ACDS)

A VSAM linear data set that contains an SCDS that has been activated to control the storage management policy for the installation. When activating an SCDS, you determine which ACDS will hold the active configuration (if you have defined more than one ACDS). The ACDS is shared by each system that is using the same SMS configuration to manage storage. See also *source control data set*, *communications data set*.

alias

An alternative name for an ICF user catalog, a non-VSAM file, or a member of a partitioned data set (PDS) or PDSE.

automatic class selection (ACS)

A mechanism for assigning Storage Management Subsystem classes and storage groups to data sets.

B

base addressing space

On an extended address volume, the cylinders with addresses below 65 536. These cylinder addresses are represented by 16-bit cylinder numbers or by 28-bit cylinder numbers with the high-order 12 bits equal to zero.

basic format

The format of a data set that has a data

set name type (DSNTYPE) of BASIC. A basic format data set is a sequential data set that is specified to be neither large format nor extended format. The size of a basic format data set cannot exceed 65 535 tracks on each volume.

basic partition access method (BPAM)

An access method that can be applied to create program libraries in direct access storage for convenient storage and retrieval of programs.

BPAM See *Basic partitioned access method*.

C

catalog

A directory of files and libraries, with reference to their locations. A catalog may contain other information such as the types of devices in which the files are stored, passwords, blocking factors.

A data set that contains extensive information required to locate other data sets, to allocate and deallocate storage space, to verify the access authority of a program or operator, and to accumulate data set usage statistics. (A) (ISO)

To enter information about a file or a library into a catalog. (A) (ISO)

The collection of all data set indexes that are used by the control program to locate a volume containing a specific data set.

To include the volume identification of a data set in the catalog.

See *VSAM master catalog*, *VSAM user catalog*.

CAS catalog address space.

catalog address space

The area of virtual storage where catalog functions are performed. It contains tables with all user catalog names identified in the master catalog, their aliases, and their associated volume serial numbers. Any changes to the master catalog are automatically reflected in these tables.

coupling facility (CF)

The hardware that provides high-speed caching, list processing, and locking functions in a Parallel Sysplex.

cylinder-managed space

The space on the volume that is managed

only in multicylinder units. Cylinder-managed space begins at cylinder address 65 520. Each data set occupies an integral number of multicylinder units. Space requests targeted for the cylinder-managed space are rounded up to the size of a multicylinder unit. The cylinder-managed space exists only on extended address volumes.

D

DASD volume

A DASD space identified by a common label and accessed by a set of related addresses. See also *volume*, *primary storage*, *migration level 1*, *migration level 2*.

data class

A collection of allocation and space attributes, defined by the storage administrator, that are used to create a data set.

device This term is used interchangeably with unit. For a disk or tape, a unit on which a volume may be mounted. For example, a tape drive is a device; a tape cartridge is a volume. Device also applies to other types of equipment, such as a card reader or a channel-to-channel (CTC) adapter.

Device Support Facilities (ICKDSF)

A program used for initialization of DASD volumes and track recovery.

DFSMS

See *Data Facility Storage Management Subsystem*.

DFSMSdfp

A DFSMS functional component or base element of z/OS, that provides functions for storage management, data management, program management, device management, and distributed data access.

DFSMSdss

A DFSMS functional component or base element of z/OS, used to copy, move, dump, and restore data sets and volumes.

DFSMShsm

A DFSMS functional component or base element of z/OS, used for backing up and recovering data, and managing space on volumes in the storage hierarchy.

DFSMS environment

An environment that helps automate and centralize the management of storage. This is achieved through a combination of hardware, software, and policies. In the DFSMS environment for MVS, this function is provided by DFSMS, DFSORT, and RACF. See also *system-managed storage*.

direct access device space management (DADSM)

A collection of subroutines that manages space on disk volumes. The subroutines are: Create, Scratch, Extend, and Partial Release.

dump A capture of valuable storage information at the time of an error.

E

ECS Enhanced Catalog Sharing.

extended address volume (EAV)

A volume with more than 65 520 cylinders.

extended addressing space (EAS)

On an extended address volume, the cylinders with addresses that are equal to or greater than 65 536. These cylinder addresses require more than 16 bits to represent.

extended format

The format of a data set that has a data set name type (DSNTYPE) of EXTREQ or EXTPREF. The data set is structured logically the same as a data set that is not in extended format, but the physical format is different. A data set in extended format can be sequential or any type of VSAM data set. An extended format data set can optionally be striped or in compressed format or both. See also *striped data set*, *compressed format*.

extent A file extent is a storage area for records allocated to a file by the server. Extents are not formally architected in DDM.

G

GDG See *generation data group*.

GDS See *generation data set*.

generation data group (GDG)

A collection of historically related

non-VSAM data sets that are arranged in chronological order; each data set is a generation data set.

generation data set

One generation of a generation data group.

I**ICKDSF**

See *Device Support Facilities program*.

initial program load (IPL)

The initialization procedure that causes an operating system to commence operation.

The process by which a configuration image is loaded into storage at the beginning of a work day or after a system malfunction.

The process of leading system programs and preparing a system to run jobs.

Synonymous with system restart, system startup.

Interactive Storage Management Facility (ISMF)

The interactive interface of DFSMS/MVS that allows users and storage administrators access to the storage management functions.

Interactive System Productivity Facility (ISPF)

An interactive base for ISMF.

IPL See *initial program load*.

ISMF See *Interactive Storage Management Facility*.

ISPF See *Interactive System Productivity Facility*.

J

JCL See *Job control language*.

Job control language (JCL)

A problem-oriented language used to identify the job or describe its requirements to an operating system.

L**large format**

The format of a data set that has a data set name type (DSNTYPE) of LARGE. A large format data set has the same characteristics as a sequential (non-extended format) data set, but its size on each volume can exceed 65 535 tracks. There is no minimum size requirement for a large format data set.

Logical partition (LPAR)

An LPAR uses software and firmware to logically partition the resources on a system. An LPAR consists of processors, memory, and I/O slots available in one processor complex.

M

master catalog

A key-sequenced data set or file with an index containing extensive data set and volume information that VSAM requires to locate data sets or files, to allocate and deallocate storage space, to verify the authorization of a program or operator to gain access to a data set or file, and to accumulate usage statistics for data sets or files.

multicylinder unit

A fixed unit of disk space that is larger than a cylinder. For example, a multicylinder unit might be 21 cylinders; in this case, the number of the first cylinder in each multicylinder unit would be a multiple of 21.

multilevel security

A security policy that allows the classification of data and users based on a system of hierarchical security levels (for example: unclassified, secret, top secret) combined with a system of non-hierarchical security categories (for example: Project A, Project B, Project C). In order to access data, a user must have a security level greater than or equal to that of the data, and be authorized to all of the categories assigned to the data.

N

name hiding

Prevents unauthorized users from obtaining names about data sets.

nonVSAM data set

A data set allocated and accessed using one of the following methods: BDAM, BPAM, BISAM, BSAM, QSAM, QISAM.

P

partitioned data set (PDS)

A data set on direct access storage that is divided into partitions, called members, each of which can contain a program, part of a program, or data.

partitioned data set extended (PDSE)

A system-managed data set that contains an indexed directory and members that are similar to the directory and members of partitioned data sets. A PDSE can be used instead of a partitioned data set.

PDS See *Partitioned data set*.

PDSE See *partitioned data set extended*.

performance

A measurement of the amount of work a product can produce with a given amount of resources.

In a system-managed storage environment, a measurement of effective data processing speed with respect to objectives set by the storage administrator. Performance is largely determined by throughput, response time, and system availability.

pool storage group

A type of storage group that contains system-managed DASD volumes. Pool storage groups allow groups of volumes to be managed as a single entity. See also *storage group*.

R

RACF See *Resource access control facility*.

Resource Access Control Facility (RACF)

An IBM licensed program that is included in z/OS Security Server and is also available as a separate program for the z/OS and VM environments. RACF provides access control by identifying and verifying the users to the system, authorizing access to protected resources, logging detected unauthorized attempts to enter the system, and logging detected accesses to protected resources.

S

SCDS See *Source control data set*.

sequential data set

A data set whose records are organized on the basis of their successive physical positions, such as on magnetic tape. Contrast with *direct data set*.

SMS Storage Management Subsystem.

SMS class

A list of attributes that SMS applies to data sets having similar allocation (data

class), performance (storage class), or backup and retention (management class) needs.

SMS-managed data set

A data set that has been assigned a storage class.

source control data set (SCDS)

A VSAM linear data set containing an SMS configuration. The SMS configuration in an SCDS can be changed and validated using ISMF. See also *active control data set*, *communications data set*.

storage administrator

A person in the data processing center who is responsible for defining, implementing, and maintaining storage management policies.

storage class

A collection of storage attributes that identify performance goals and availability requirements, defined by the storage administrator, used to select a device that can meet those goals and requirements.

storage facility

The physical components that comprise a single storage server (DS8000 or DS6000) including the base frame and the optional expansion frames. A storage facility is composed of two processor complexes (servers) and some number of storage devices that are packaged in one or more enclosures with associated power supplies and cooling.

storage facility image

For hosts that use FICON/ESCON I/O commands, a storage facility image contains one or more ESCON or Fibre Channel (FICON) I/O interfaces (ports) that can access one or more control-unit images. Each control-unit image has an associated set of devices. Each device is assigned a unique device address on the control-unit image. Depending upon the model, more than one storage facility image can be configured on a storage facility. (For DS8000, the storage facility can support more than one storage facility image.) A storage facility image might also be referred to as a storage image.

storage group

A collection of storage volumes and

attributes, defined by the storage administrator. The collections can be a group of DASD volumes or tape volumes, or a group of DASD, optical, or tape volumes treated as a single object storage hierarchy. See also *VIO storage group*, *pool storage group*, *tape storage group*, *object storage group*, *object backup storage group*, *dummy storage group*.

storage management

The activities of data set allocation, placement, monitoring, migration, backup, recall, recovery, and deletion. These can be done either manually or by using automated processes. The Storage Management Subsystem automates these processes for you, while optimizing storage resources. See also *Storage Management Subsystem*.

Storage Management Subsystem (SMS)

A DFSMS facility used to automate and centralize the management of storage. Using SMS, a storage administrator describes data allocation characteristics, performance and availability goals, backup and retention requirements, and storage requirements to the system through data class, storage class, management class, storage group, and ACS routine definitions.

stripe

In DFSMS, the portion of a striped data set, such as an extended format data set, that resides on one volume. The records in that portion are not always logically consecutive. The system distributes records among the stripes such that the volumes can be read from or written to simultaneously to gain better performance. Whether it is striped is not apparent to the application program.

sysplex

A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process customer workloads.

system-managed storage

Storage managed by the Storage Management Subsystem. SMS attempts to deliver required services for availability, performance, and space to applications. See also *system-managed storage environment*.

system programmer

A programmer who plans, generates, maintains, extends, and controls the use of an operating system and applications with the aim of improving overall productivity of an installation.

T**threshold**

A storage group attribute that controls the space usage on DASD volumes, as a percentage of occupied tracks versus total tracks. The *low migration threshold* is used during primary space management and interval migration to determine when to stop processing data. The *high allocation threshold* is used to determine candidate volumes for new data set allocations. Volumes with occupancy lower than the high threshold are selected over volumes that meet or exceed the high threshold value.

track address

A 32-bit number that identifies each track within a volume. A track address is in the format hex *CCCCcccH*, where *CCCC* is the low-order 16 bits of the cylinder number, *ccc* is the high-order 12 bits of the cylinder number, and *H* is the four-bit track number. For compatibility with older programs, the *ccc* portion is hex 000 for tracks in the base addressing space.

track-managed space

The space on a volume that is managed in tracks and cylinders. For an extended address volume, track-managed space ends at cylinder address 65 519. Each data set occupies an integral number of tracks.

U

UNIX A highly portable operating system originally developed by Bell Laboratories that features multiprogramming in a multi-user environment. UNIX is implemented in the C language. UNIX was originally developed for use on minicomputers but has been adapted on mainframes and microcomputers. It is especially suitable for multiprocessor, graphics, and vector-processing systems.

V

VLF Virtual lookaside facility

virtual storage access method (VSAM)

An access method for direct or sequential processing of fixed and variable-length records on direct access devices. The records in a VSAM data set or file can be organized in logical sequence by a key field (key sequence), in the physical sequence in which they are written on the data set or file (entry-sequence), or by the relative-record number.

volume

The storage space on DASD, tape, or optical devices, which is identified by a volume label. See also *DASD volume*, *optical volume*, *tape volume*.

VSAM

See *virtual storage access method*.

Z

z/OS z/OS is a network computing-ready, integrated operating system consisting of more than 50 base elements and integrated optional features delivered as a configured, tested system.

z/OS Network File System

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

Index

A

accessibility 77
 contact IBM 77
 features 77
assistive technologies 77

C

catalog enhancements 3, 31
catalog enhancements, R12
 description 31
catalog enhancements, R22
 description 3
contact
 z/OS 77

D

DADSM/CVAF enhancements 17, 55
 overview 17, 55
DFSMS
 ACS enhancements 29
 Advanced Copy Services
 enhancements 29
 DADSM/CVAF enhancements 17, 55
 OAM enhancements 13, 43
 PDSE enhancements 39
 utility enhancements 41
 VSAM enhancements 45
DFSMSdfp
 catalog enhancements 3, 31
 DFSMSdss enhancements 57
 Open/Close/End of Volume
 enhancements 11, 37
 SMS enhancements 7, 35
DFSMShsm
 enhancements 61
DFSMShsm enhancements 61
 administration 69
 planning tasks 68
DFSMShsm usability enhancements
 overview 61
DFSMSrmm
 operating 73
DFSMSrmm enhancements 25, 73

K

keyboard
 navigation 77
 PF keys 77
 shortcut keys 77

N

navigation
 keyboard 77
Notices 81

O

OAM enhancements 13, 43
 overview 13, 43
OCE enhancements
 description 11, 37
Open/Close/End of Volume
 enhancements 11, 37

z/OS V2R1 updates (*continued*)

DFSMSdfp

 Open/Close/End of Volume
 enhancements 37
SMS enhancements 35

DFSMSdss 57

DFSMShsm
 enhancements 61

DFSMSrmm
 operating 73

z/OS V2R2 updates

DFSMS

 DADSM/CVAF enhancements 17
 OAM enhancements 13

DFSMSdfp

 catalog enhancements 3
 Open/Close/End of Volume
 enhancements 11
SMS enhancements 7

S

sending comments to IBM vii
shortcut keys 77
SHOWCB
 programming tasks 47
storage tiers 62
summary of changes ix
 as updated February 2015 ix
 as updated March 2014 x
Summary of changes x

T

trademarks 85

U

user interface
 ISPF 77
 TSO/E 77

V

VASM replication
 programming tasks 47
VSAM
 administration tasks 46
VSAM enhancements 45
 overview 45
VSAM RLS
 administration tasks 46

Z

z/OS V1R12 updates
 DFSMS

 OAM enhancements 43

 DFSMSdfp

 catalog enhancements 31

z/OS V2R1 updates

 DFSMS

 ACS enhancements 29

 Advanced Copy Services

 enhancements 29

 DADSM/CVAF enhancements 55

 PDSE enhancements 39

 utility enhancements 41

 VSAM enhancements 45

IBM[®]

Product Number: 5650-ZOS

Printed in USA

SC23-6857-04

