z/OS
3.2

DFSMS Using the New Functions

SC23-6857-70

Note

Before using this information and the product it supports, read the information in “Notices” on page 109.

This edition applies to IBM® z/OS® 3.2 (5655-ZOS) and to all subsequent releases and modifications until otherwise indicated in new editions.

Last updated: 2026-01-26

Contents

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, see Appendix A, “Accessibility,” on page 107.

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 Upgrade Workflow*, which covers the required tasks for migrating to each enhancement.

z/OS information

This content 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*.

How to provide feedback to IBM

We welcome any feedback that you have, including comments on the clarity, accuracy, or completeness of the information. For more information, see How to send feedback to IBM.

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.

Note: IBM z/OS policy for the integration of service information into the z/OS product documentation library is documented on the z/OS Internet Library under IBM z/OS Product Documentation Update Policy (www.ibm.com/docs/en/zos/latest?topic=zos-product-documentation-update-policy).

Summary of changes for z/OS 3.2

The following content is new, changed, or no longer included in z/OS 3.2.

New

The following content is new.

January 2026 refresh

- Added Chapter 1, “OAM’s REST API Support ,” on page 3.

December 2025 refresh

- Added Chapter 2, “Optimization Mode for Data Set Copy,” on page 5.

October 2025 refresh

- Note added in Chapter 4, “Using the z/OS data set encryption enhancements,” on page 15.

September 2025 release

- None.

Changed

The following content is changed.

September 2025 release

- None.

Deleted

The following content is deleted.

September 2025 release

- None.

Summary of changes for z/OS 3.1

This information contains no technical changes for this release.

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 Upgrade Workflow* 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 V3R2

This topic describes how to use new DFSMS functions in z/OS V3R2

Chapter 1. OAM's REST API Support

Separate from the OSREQ assembler API, OAM also provides REST API support. AWS S3 support is provided for storing, retrieving, deleting, querying, and changing the management characteristics (policies) of objects managed by OAM. This support was initially introduced in z/OS 3.1 with APAR OA64282: NEW FUNCTION - OAM RESTFUL API (www.ibm.com/support/pages/apar/OA64282).

An OAM Web Application is provided along with an OAM Bridge Program (OAMREST) running in WebSphere Liberty. OAM will invoke the OSREQ interface, on behalf of the REST request, to have the request processed by OAM. An object stored through OAM's REST API support can be natively retrieved using the existing OSREQ API and vice versa. All existing functions and capabilities of OAM will continue to be supported through the REST interface. See, *z/OS DFSMS OAM Application Programmer's Reference* and *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support*

Chapter 2. Optimization Mode for Data Set Copy

z/OS DFSMS V3R2 provides the following enhancement to optimize making a copy of an encrypted and/or compressed format data set by using the bypass encryption interface and the bypass compression interface with extended format data sets, which applies to z/OS V2.5 and z/OS V3.1 as per OA63434: NEW FUNCTION (www.ibm.com/support/pages/apar/OA63434).

You can write an assembler program the does these things with extended format data set:

- Read and write an encrypted data set without the access method decrypting or re-encrypting data. The blocks read and written remain encrypted.
- Read and write a compressed format data set without the access method decompressing or re-compressing data. The blocks read and written remain compressed.
- Read and write an encrypted and compressed format data set without the access method decrypting and decompressing data or re-compressing and re-encrypting data. The blocks read and written remain encrypted and compressed.

This type of program is using bypass mode.

This support delivered via PTFs for these APARs:

- OA63434: NEW FUNCTION (www.ibm.com/support/pages/apar/OA63434) (BAM)
- OA63828: NEW FUNCTION - DFSMS OPTIMIZATION MODE FOR DATA SET COPY (www.ibm.com/support/pages/apar/OA63828) (OPEN/CLOSE/EOV)
- OA63439: NEW FUNCTION - DFSMS OPTIMIZATION MODE FOR DATA SET COPY (www.ibm.com/support/pages/apar/OA63439) (VSAM)
- OA63441: NEW FUNCTION - DFSMS OPTIMIZATION MODE FOR DATA SET COPY (www.ibm.com/support/pages/apar/OA63441) (IDCAMS)
- OA64055: NEW FUNCTION - DFSMS OPTIMIZATION MODE FOR DATA SET COPY (www.ibm.com/support/pages/apar/OA64055) (VSAM RLS)
- OA67110: NEW FUNCTION - DFSMS OPTIMIZATION MODE FOR DATA SET COPY (www.ibm.com/support/pages/apar/OA67110) (SMS)
- OA67112: NEW FUNCTION - DFSMS OPTIMIZATION MODE FOR DATA SET COPY (www.ibm.com/support/pages/apar/OA67112) (Catalog)

This new function is in alignment with IBM's support of pervasive encryption.

Use of this new function does not require any system option. In order for a copying program to exploit this new function, source code changes are required. The IDCAMS REPRO command exploits it.

If you upgrade a program to use the new function and run the program on a system in which the new function is not installed, it will have no effect. That means that the DCBE or ACB option and the AMIA will have no effect on OPEN.

A program can test whether the bypass function has been installed on the system by testing the DFABypassCmpEnc bit in the DFA, data facilities area.

Opening a data set for output with bypass mode requires RACF CONTROL authority to the data set unless the program is authorized (APF-authorized, supervisor state or system key). This level of authority has not previously been applied to data sets. Most of the documentation for this new function is in *z/OS DFSMSdfp Advanced Services*.

Some are documented in *z/OS DFSMS Using Data Sets* and *z/OS DFSMS Macro Instructions for Data Sets*.

See *z/OS MVS System Management Facilities (SMF)* for changes to the type 14, 15 and 62 records.

- SMF 62 will reflect the new MACRF option to indicate that the data set was opened with bypass encryption processing within the existing SMF ACBMACR4 field.

- SMF62DSEB was redefined to mean that an AMIA was passed on the ACB.

Part 2. Using new DFSMS functions in z/OS V2R5

These topics describe how to use new DFSMS functions in z/OS V2R5.

Chapter 3. Using the Object Access Method (OAM) enhancements

z/OS DFSMS V2R5 provides the following enhancements to the Object Access Method (OAM):

Cloud as a tier (as of z/OS V2R3 and z/OS V2R4 OA55700 and base z/OS V2R5)

- A new DFSMS component, CDA (cloud data access), is introduced which provides a common interface to cloud providers that support the S3 API. CDA also manages cloud access keys.
- Is able to store and manage primary copies of OAM objects on cloud storage with the same capabilities for access, transition, and backup as currently exist for primary object copies stored on a file system. Backups can continue to be stored to removable media (optical or tape).
- Is able to recall an object that is stored in the cloud to the disk level of the storage hierarchy in any of the same ways that primary object copies on tape or optical can currently be recalled. Cloud objects can be recalled to both disk sublevels (Db2® or file system).
- Is able to define a user configurable mapping between OAM object storage groups and user-defined cloud containers. For new objects, each storage group is associated with a single container, but multiple storage groups can share a container. When an object is stored, it remains in the same container and continues to be accessible even if the container associated with its storage group is changed.
- Is able to store an object with the same storage group name, collection name, and object name from multiple unrelated OAM instances in the same cloud container and ensure that each instance accesses its own object. However, OAM typically recommends using a different container for better management/ logical partitioning of the data.
- File names for OAM objects that are stored in the cloud have a naming schema (by using forward slashes) that enables customers to create pseudo hierarchical folders for information lifecycle management (ILM) within the cloud.

Cloud and file system backup support (as of z/OS V2R3 and z/OS V2R4 OA59615 and base z/OS V2R5)

- Is able to store immediate backup copies of OAM objects on cloud storage and file system (disk level 2) storage in addition to removable media storage (optical or tape) today.
- Enables the OSMC storage management cycle function to create and manage back up copies on cloud storage and file system (disk level 2) storage in addition to removable media storage (optical or tape) today.
- Is able to retrieve backup copies of an object by using OSREQ RETRIEVE command and by using the Auto Access to Backup facility from cloud storage and file system (disk level 2) storage.
- Is able to delete the associated backup copies from cloud storage and file system (disk level 2) storage when a primary object is being deleted by using OSREQ DELETE command.
- Has single object recovery from backup copies on cloud or file system.
- The volume recovery utility can recover volumes containing primary objects from backup copies on cloud and file system. It can recover backup volumes to new backup location of cloud or file system.
- The MOVEVOL utility can move a backup volume to a new backup location of cloud or file system.
- New displaying cloud task and cloud task cancel operator commands can be used to display cloud task information and cancel a cloud task.

New generic variable OAMVAR on CBROAMxx ONLYIF statement (as of z/OS V2R3 and z/OS V2R4 OA59711 and base z/OS V2R5)

- This new ONLYIF keyword provides another option for more granularity/filtering of OAM statements based on user controlled literal values and system_symbols. It performs a comparison and if the values match, the ONLYIF is enabled, otherwise the ONLYIF is ignored. Both the system_symbol and literal_value are required for the OAMVAR keyword to be processed.

New OAM provided commands for lookup of OAM macro and OSREQ error codes (as of z/OS V2R3 and z/OS V2R4 OA58344 and base z/OS V2R5)

- You can issue a DISPLAY command and pass a reason code to it as input. The returned display gives the error description that is commonly found in this documentation. Therefore, a user is allowed to see that information without having to reference an outside source.
- From a console environment, the following commands are now available:

  DISPLAY OAM,OAMRC,*xxyy*
  DISPLAY OAM,OSREQRC,*wwxxyyzz*
- From a TSO environment, the following commands are now available:

  OAMUTIL OAMRC *xxyy*
  OAMUTIL OSREQRC *wwxxyyzz*

OSMC CBR9370I message enhanced to now display deletion attempted and done statistics (as of z/OS V2R3 and z/OS V2R4 OA58002 and base z/OS V2R5)

- This update helps aid users in diagnosing slow OSMC run times, where OSMC might be attempting to expire many objects (with management class criteria with expiration criteria met) while CBRHADUX exit is set to deny expiration function.

DB2ID restriction lift for multiple OAM configuration with only one object subsystem (as of z/OS V2R3 and z/OS V2R4 OA57842 and base z/OS V2R5)

- The OSREQ DB2ID keyword is now optional for Multiple OAM configurations with only one object subsystem. This allows for users to migrate from a Classic to a Multiple configuration with one object subsystem without API changes.

Multiple OAM object address space limit raised 2 - 5 (as of z/OS V2R3 and z/OS V2R4 OA56291 and base z/OS V2R5)

- Before this support, a Multiple OAM configuration was capped at two object instances and one tape instances per LPAR. With this new support, the cap is raised to five object instances and one tape instance per LPAR. This allows for more flexibility in splitting data workloads between more Db2 subsystems.

OAM Db2 connection management enhancements

- A new configurable timer is added for situations when the OAM address space detects that Db2 entered maintenance mode. Before V2R5, OAM attempts a reconnect to Db2 every 5 minutes. With this support, OAM introduces the DB2RECONNECTWAITTIME keyword for the SETOPT statement within CBROAMxx PARMLIB member which allows the user to configure how many seconds (10-300) for OAM to wait before attempting a reconnect to Db2.
- In addition to the configurable timer, an immediate connect function is added with the F OAM,DB2CONN operator command. Use this when the operator does not want to wait for OAM to attempt an auto reconnect, but rather would like to attempt a reconnect immediately. This eliminates unnecessary downtime of the OAM address space when Db2 is brought back up to normal operating capacity.

- The F OAM,RESTART operator command now accepts a new keyword DB2SSID. The DB2SSID keyword is used to force the OAM address space restart to issue CBR0006D which prompts the operator to explicitly specify the DB2SSID that is used for this OAM address space. This allows the ability for an OAM address space to switch between non-object/object classic configurations and switch between different Db2 subsystems.
- Various OAM address space initialization improvements are made when Db2 is detected to be in an inoperable state. Before V2R5, a classic OAM address space failed to initialize. With this support, a classic OAM address space now gives the operator the option to continue address space initialization with no object support (SMStape support only). This allows the operator to have higher OAM accessibility for SMStape related activities during situations when Db2 maintenance is occurring or other Db2 issues are being resolved.

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

Part 3. Using new DFSMS functions in z/OS V2R4

These topics describe how to use new DFSMS functions in z/OS V2R4.

Chapter 4. Using the z/OS data set encryption enhancements

Introduction

With z/OS data set encryption, you can encrypt data without requiring application changes. z/OS data set encryption, through SAF controls and RACF® or equivalent function along with SMS policies, allows you to identify new data sets or groups of data sets to be encrypted. You can specify encryption key labels to identify encryption keys to be used to encrypt selected data sets. The specified key label and encryption key must exist in the ICSF key repository (CKDS). With data set encryption, you are able to protect data residing on disk from being viewed by unauthorized users in the clear. Authorization is based on access to the key label that is associated with the data set and used by the access methods to encrypt and decrypt the data.

z/OS data set encryption provides the ability to encrypt the following types of data sets:

- Sequential extended format data sets, accessed through BSAM and QSAM
- VSAM extended format data sets (KSDS, ESDS, RRDS, VRRDS, LDS), accessed through base VSAM and VSAM RLS
- Version 2 PDSEs, accessed through BSAM, QSAM or BPAM.
- Sequential basic format data sets, accessed through BSAM, QSAM and EXCP. (Note: EXCP access does require application changes.)
- Sequential large format data sets, accessed through BSAM, QSAM and EXCP. (Note: EXCP access does require application changes.)

Encrypted data sets must be SMS-managed. Encrypted extended format data sets can also be compressed format.

Importance of host based compression
It is recommended that clients using host based encryption also exploit host based compression prior to the encryption of the data. If the data is not compressed prior to the encryption there can be consequences to other parts of the client infrastructure. For example, replicated data that is being compressed in the SAN infrastructure by DWDM technology will no longer be effective trying to compress encrypted data. Without compressing prior to the encryption of the data, additional bandwidth might be required. Another example might be that tape systems can require additional capacity in terms of disk space, in the case of virtual tape, or tape cartridges. If deduplication of data is supported, host encryption can prevent deduplication from working. Therefore, where possible, use compressed format data sets. With encrypted compressed format data sets, the access methods perform compression before encryption. See Considerations when planning for data set encryption for more information.

To create an encrypted data set, a key label must be supplied on new data set allocation. The key label must point to an AES-256 bit encryption DATA key within the ICSF key repository (CKDS) to be used to encrypt or decrypt the data. For each encrypted data set, its key label is stored in the catalog. The key label is not sensitive information; it identifies the encryption key, which is sensitive; therefore, IBM recommends only using secure keys. For more information, see *z/OS Cryptographic Services ICSF System Programmer's Guide*.

Hardware requirements

Exploitation of this function requires IBM Enterprise z196 or later. It also requires the following cryptographic hardware features:

- Crypto Express3 Coprocessor or later
- Feature 3863, CP Assist for Cryptographic Functions (CPACF)

Operating System requirements

- ICSF is installed and configured with a CKDS and AES master key loaded. See Chapter 12, “Using the z/OS data set encryption enhancements ,” on page 47 for more information.

Coexistence requirements

- On a z/OS V2R3 or z/OS V2R4 system with OA56622, or on a later system, you can create and access encrypted basic and large format data sets.
- On a z/OS V2R2 system with OA50569, you can create encrypted data sets as well as access encrypted data sets created on a z/OS V2R2 (with OA50569) or later system.
- On a z/OS V2R1 system with OA50569, you cannot create encrypted data sets. However, you can access encrypted data sets created on a z/OS V2R2 (with OA50569) or later system.

Note: The minimum software release that can support encrypted data sets is z/OS V2R1 with OA50569. An attempt to access an encrypted data set on a lower release will result in loss of access to the data. Ensure that all systems are at the minimum hardware and software levels before encrypting any data sets.

Note: The minimum software release that can support PDSEs is z/OS V2R2 with OA56324. An attempt to access an encrypted data set on a lower release will result in loss of access to the data. Ensure that all systems are at the minimum hardware and software levels before encrypting any data sets.

Note: The minimum software release that can support encrypted basic and large format data sets is z/OS V2R3 with OA56622. An attempt to access an encrypted basic or large format data set on a z/OS V2R2 system without OA60160 or a lower system, will have unpredictable results. With OA60160 on a z/OS V2R2 system, an attempt to access an encrypted basic or large format data set will result in an expected ABEND0C1. Ensure that all systems are at the minimum hardware and software levels before encrypting any data sets.

Before enabling this function

Because data set encryption has both hardware and software requirements, you must consider all systems that share data with a system on which you plan to enable data set encryption before creating an encrypted data set. This includes backout software levels, backup systems, read-only systems, replication target systems and disaster recovery systems. Before encrypting data sets other than those used for testing, be sure that all the systems that must access encrypted data sets are capable of doing so by meeting the required hardware and software requirements. In addition to the hardware and software requirements that must be available on every system that will access the encrypted data sets, all key labels and encryption keys associated with the encrypted data sets must also be available.

Take these steps to make data set encryption unavailable to users who are not explicitly authorized to use it:

- Define the STGADMIN.SMS.ALLOW.DATASET.ENCRYPT profile in the FACILITY class, and set the universal access to NONE:

```
RDEFINE FACILITY STGADMIN.SMS.ALLOW.DATASET.ENCRYPT UACC(NONE)
```

- To add PDSE as a supported data set type for encrypted data set allocation, define the STGADMIN.SMS.ALLOW.PDSE.ENCRYPT profile in the FACILITY class.

```
RDEFINE FACILITY STGADMIN.SMS.ALLOW.PDSE.ENCRYPT UACC(NONE)
```

- If the FIELD class is active, check for any profile that would allow any user without SPECIAL attribute access to the DATASET.DFP.DATAKEY. If there are none, no additional action is needed. If there is any profile that would allow access to DATASET.DFP.DATAKEY, create a DATASET.DFP.DATAKEY profile in the FIELD class with a UACC of NONE:

```
RDEFINE FIELD DATASET.DFP.DATAKEY UACC(NONE)
```

Taking those steps is intended to assure that only authorized users are allowed to use data set encryption. Such users should be made aware that until the decryption functions are available on all sharing systems, backup systems, and disaster recovery systems, access to encrypted data can be lost at any time.

Tasks for setting up data set encryption

Performing the following tasks is necessary to begin using data set encryption.

Create key labels and encryption keys

To create an encrypted data set, a key label must be assigned to the data set. The key label and its associated AES-256 bit encryption key must exist in the CKDS by the time the data set is opened. To create keys in the CKDS, refer to *z/OS Cryptographic Services ICSF System Programmer's Guide*. IBM recommends the use of secure keys.

Set up CSFKEYS

Important: The following setup is required regardless of whether CHECKAUTH(YES) or CHECKAUTH(NO) is specified in the ICSF installation options data set. CHECKAUTH(NO) is the default.

To control which users can use which keys, protect resources CSFKEYS class. The CSFKEYS class controls access to cryptographic keys identified by the key label.

To enable the use of ICSF keys:

1. Grant the user READ authority to the resource key-label in the CSFKEYS class. This authority can be granted for all uses of the key-label or only when the use of the key label is permitted when using the CRITERIA value of DSENCRYPTION for SMS.

   - if the class has not already been enabled for generics

   ```
   SETROPTS GENERIC(CSFKEYS)
   ```

   - Define profiles in the CSFKEYS class to protect key labels

   ```
   RDEFINE    CSFKEYS  profile-name UACC(NONE)
              ICSF(SYMCPACFWRAP(YES)   SYMCPACFRET(YES))
   ```

   - Give the users (preferably groups) access to the profiles

   ```
   PERMIT profile-name CLASS(CSFKEYS)
          ID(name)    ACCESS(READ)
   ```

   or

   ```
   PERMIT profile-name CLASS(CSFKEYS)
          ID(name)    ACCESS(READ)
          WHEN(CRITERIA(SMS(DSENCRYPTION)))
   ```

2. Define the ICSF information for the key with SYMCPACFWRAP(YES) and SYMCPACFRET(YES), if it was not specified on the DEFINE

   ```
   RALTER CSFKEYS profile-name
          ICSF(SYMCPACFWRAP(YES)   SYMCPACFRET(YES))
   ```

3. Set RACF options

   - if the CSFKEYS class is not already active

   ```
   SETROPTS CLASSACT(CSFKEYS)
   ```

   - if the CSFKEYS class has not already been RACLISTed

   ```
   SETROPTS RACLIST(CSFKEYS)
   ```

- or if the CSFKEYS class has already been RACLISTed

```
SETROPTS RACLIST(CSFKEYS) REFRESH
```

Set up CSFSERV

Important: The following setup is required only if CHECKAUTH(YES) is specified on the ICSF installation options data set. CHECKAUTH(NO) is the default.

Protect the resources CSFSERV class. ICSF controls access to cryptographic services through the CSFSERV resource class. For more information, see *z/OS Cryptographic Services ICSF Administrator's Guide* .

The following table summarizes the CSFSERV resource required for processing encrypted data sets.

Table 1. CSFSERV resource required for data set encryption

Function	ICSF callable service	Resource
CKDS Key Record Read2	CSNBKRR2	CSFKRR2

If the user does not have sufficient access, open processing will fail. An informational message ICH408I (which indicates insufficient authorization) might be issued.

1. Grant the user READ authority to the resource CSFSERV class:

   - if the class has not already been enabled for generics

   ```
   SETROPTS GENERIC(CSFSERV)
   ```

   - Define profiles in the CSFSERV class to protect key labels

   ```
   RDEFINE CSFSERV * UACC(NONE)
   ```

   - Define profile CSFKRR2 if it does not exist

   ```
   RDEFINE CSFSERV CSFKRR2 UACC(NONE)
   ```

   - Give the users (preferably groups) access

   ```
   PERMIT CSFKRR2 CLASS(CSFSERV) ID(groupid) ACCESS(READ)
   ```

2. Set RACF options:

   - if the CSFSERV class is not already active

   ```
   SETROPTS CLASSACT(CSFSERV)
   ```

   - if the CSFSERV class has not already been RACLISTed

   ```
   SETROPTS RACLIST(CSFSERV)
   ```

   - or if the CSFSERV class has already been RACLISTed

   ```
   SETROPTS RACLIST(CSFSERV) REFRESH
   ```

Create an encrypted data set

To create an encrypted data set, you must assign a key label to the data set when it is newly allocated (data set create). A key label can be specified through any of the following methods:

- RACF® data set profile
- JCL, dynamic allocation, TSO ALLOCATE, IDCAMS DEFINE

- SMS data class

To specify a key label using the DFP segment in the RACF data set profile, use keyword DATAKEY(Key-Label). The system will use this key label for data sets that are created after DATAKEY is added to the data set profile. Use keyword NODATAKEY to remove a key label, if defined, from the RACF DFP segment. The key label is ignored for a data set that is not a DASD data set.

To specify a key label using JCL, dynamic allocation, or TSO allocate, use JCL keyword DSKEYLBL='key-label', dynamic allocation text unit DALDKYL, or TSO allocate DSKEYLBL(key-label). DSKEYLBL parameter is effective only if the new data set is on DASD. The key label is ignored for a data set that is not a DASD data set.

See details about the DSKEYLBL parameter(key-label) keyword on the JCL DD statement in *z/OS MVS JCL Reference*.

To specify a key label using SMS data class, use the *Data Set Key Label* field on the ISMF DEFINE/ALTER panel. The system will use this key label fordata sets that are created after the data set key label is added to the data class. The key label is ignored for a data set that is not a DASD data set.

See details on using the new Data Set Key Label field in the ISMF panels in *z/OS DFSMS Using the Interactive Storage Management Facility*.

To specify a key label using the DEFINE CLUSTER command for a VSAM CLUSTER, use the KEYLABEL parameter; for example, KEYLABEL(MYLABEL). Any alternate index associated with the CLUSTER will also be encrypted and use the same key label as specified for the CLUSTER. The key label is ignored for a data set that is not a DASD data set.

For more information on using the DEFINE CLUSTER command for a VSAM CLUSTER, see *z/OS DFSMS Access Method Services Commands*.

When a key label is specified on more than one source, the key label is derived from one of the above sources only on the initial data set allocation (on data set create). The key label is derived in the following order of precedence:

1. From the DFP segment in the RACF data set profile.
2. Explicitly specified on the DD statement, dynamic allocation text unit, TSO ALLOCATE command, or DEFINE CLUSTER control statement.
3. From the data class that applies to the current DD statement.

   Note: The REFDD and LIKE JCL DD statement keywords do not cause a key label from the data set referred to be used when allocating a new data set.

On successful allocation of an encrypted data set, the following message is issued:

```
IGD17150I DATA SET dsname IS ELIGIBLE FOR ACCESS METHOD ENCRYPTION
KEY LABEL IS (key_label)
```

Specifying a key label for a non-extended format data set

If an encryption key label is specified for a DASD data set that is not extended format and the FACILITY class resource STGADMIN.SMS.ALLOW.DATASET.SEQ.ENCRYPT is not defined, the key label is ignored and the data set is successfully created as non-encrypted non-extended format data set. In addition, SMS message IGD17156I is issued (or if using IDCAMS DEFINE and data set is non-SMS managed, message IDC3040I is issued) indicating that the key label is ignored. Instead to have the system fail the allocation, the user must have at least READ authority to the resource in the FACILITY class: STGADMIN.SMS.FAIL.INVALID.DSNTYPE.ENC

If this facility class resource exists and the user has at least read authority, SMS will fail the allocation and issue message IGD17151I (or if using IDCAMS DEFINE and non-SMS managed data set request, message IDC3039I is issued).

Encrypted data sets must be SMS-managed. Encrypted extended format data sets can also be compressed format.

Specifying a key label for a basic or large format data set

With z/OS V2R3 and z/OS V2R4 systems (with OA56622) or higher levels, data set encryption supports basic and large format data sets. To indicate that the specification of a key label should result in the creation of an encrypted basic or large format data set (that is, indicate if the basic and large format data sets type should be regarded as a supported data set type for data set encryption), the data set must be SMS-managed and the following resource in the RACF FACILITY class must be defined: STGADMIN.SMS.ALLOW.DATASET.SEQ.ENCRYPT.

The system checks whether this resource is fully defined when the data set is first allocated (created). The system does not require any authorization to be specified for this resource in order for it to take affect. Therefore, the resource must not be defined until basic and large format data sets are to be created as encrypted.

Note: Restrictions and considerations must be evaluated before enabling encryption of basic and large format data sets.

The physical format of an encrypted basic or large format data set is enhanced with block prefixes that are transparent to BSAM and QSAM programs. The access method adds an 8-byte prefix to each physical block of the data set on output. Each block prefix will contain a value that uniquely identifies the block. The content of this prefix is relevant only to application programs that use EXCP. The prefix will not be included in the physical block size of the data set stored in the block size fields in the DSCB, DCB, DCOLLECT and SMF records. You must take the length of the block prefixes into consideration when requesting DASD space.

- The prefix will not be allowed on WRITE or PUT requests, nor returned on READ and GET requests.
- The system-determined block size algorithms will take the block prefix length into consideration.

Note: In rare cases, there may be encrypted basic and large format data sets which are created without a block prefix. Such a data set can only be opened for EXCP. A flag in the catalog encryption cell will indicate whether the data set has prefixes. After open, the ISITMGD macro can be used to determine the length of the prefix.

Applications using standard BSAM and QSAM APIs require no, or minimal changes, to access encrypted basic and large format data sets. Minimal changes may be required if the application performs DASD space calculations that must take into account the block prefix. Applications using EXCP **require** changes to access encrypted basic and large format data sets by performing encryption and decryption of data. The IGGENC macro can be used to simplify encryption and decryption of data while ensuring compatibility with the access methods. (Note: All references to EXCP apply equally to the EXCPVR and XDAP macros, unless otherwise stated.)

The restrictions identified under “Restrictions for encrypted data sets” on page 25 also apply to encrypted basic and large format data sets. **Additional** restrictions associated with encrypted basic and large format data sets are identified in “Additional restrictions for basic and large format encrypted data sets” on page 25.

Candidates for encrypted basic and large format data sets

SMF records can be used to view activity over a period of time. This information can be valuable when evaluating which basic and large format data sets may be eligible for converting to encrypted basic and large format data sets. DCOLLECT records may also be used for some of this determination. This section describes fields within SMF Type 14, SMF Type 15, and SMF Type 42-6 records that can help you identify whether your data sets would be candidates for data set encryption based on the list of restrictions for encrypted basic and large format data sets.

- SMF Type 14 and SMF Type 15 records are mapped by macro IFGSMF14.
- SMF Type 42-6 records are mapped by macro IGWSMF.

You must determine that the data set is a basic or large format data set. You can do this by testing a field in the JFCB which is found in SMFJFCB1. Test JFCDSORG for bit JFCORGPS being on and test for bit SMF14STR being off. This combination indicates the data set is a sequential data set but not extended format.

To determine the application that accesses the data set, you can use these fields in the SMF 14 and 15 records:

- SMF14JBN for the jobname
- SMFJOBID for the job identifier
- SMF14SPN for the step name
- SMF14PGN for the program name

To determine the application that accesses the data set, you can use these fields in the SMF 42-6 records:

- S42JDJNM for the jobname

To determine if your data set is processed by EXCP:

- Scan your programs for use of EXCP, EXCPVR or XDAP macros
- Review SMF Type 14 and SMF Type 15 records:
  - You can use SMFDCBMF to identify data sets opened with EXCP. This is a 2 byte field which is a copy of DCBMACRF, where the high order bit indicates MACRF=E. You can find DCBMACRF in the DCBD mapping macro.
  - You can then also use SMF14EXCPBAM bit to identify data sets opened with BSAM or QSAM, but then accessed via EXCP.
- Review SMF 42-6 records:
  - You can use S42DSEXC to identify data sets opened with EXCP.

  Note: Use the z Batch Network Analyzer tool to identify data sets accessed by EXCP, as it uses the SMF records to make this determination.
- To determine if your data set is accessed by an EXCP application that can support encrypted data sets:
  - You can use flag SMF14DSENCRYPTOK to identify whether the application program used EXCP and the program is enabled to handle data set encryption for basic and large format data sets. This does not imply that the data set is DASD.

    Note: The system detects this condition by testing DCBE DSENCRYPT=OK is specified regardless of the data set type.

Determine if your data sets meet the restrictions for encrypted basic/large format data sets

To determine if your data set meets the minimum block size requirement:

- You can use the SMFJFCB1 field to find the block size of the data set. Using the JFCB mapping macro, IEFJFCBN, test field JFCBLKSI to determine if the block size is at least the minimum required for encryption, which is 16 bytes.
- You can use S42DSBSZ in SMF 42-6 for block size.

To determine if your data set meets the minimum record length requirement:

- You can use the SMFJFCB1 field to find the record length of the data set. Using the JFCB mapping macro, IEFJFCBN, test field JFCLRECL to determine if the record length is at least the minimum required for encryption, which is 16 bytes for record format F(B(S)) and 12 bytes for record format V(B(S)). Use SMFDCBRF to determine the record format, which is mapped the same as DCBRECFM.
- You can also use the following to view the record format, block size and record length:
  - TSO LISTDSI command for REXX
  - IEHLIST LISTVTOC
  - ISPF option 3.2

To determine if your data set uses hardware keys (DCBKEYLE):

- You can use the SMFJFCB1 field to determine if hardware keys are used with the data set. Using the JFCB mapping macro, IEFJFCBN, test field JFCKEYLE for zeros to ensure no hardware keys are in use.

To determine if your data set is accessed by BDAM:

- You can use the SMF14DDA bit to determine if BDAM has been used to access the data set. If the flag is on, this data set is not a candidate for encryption.

The SMF records indicate processing based on a span of time. You can use the IDCAMS DCOLLECT command to create records for data sets that might not have been accessed recently. These records help with analysis of the attributes of data sets, but do not show how the data sets were used. The mapping macro for DCOLLECT records is IDCDOUT.

- To determine the block size, you can use DCDBKLNG.
- To determine the record length, you can use DCDLRECL.
- To determine the record format, you can use DCDRECRD. This is a copy of DCBRECFM.
- To determine the data set organization, you can use DCDDSORG.

Enable data set encryption

An enablement action is required to allow the creation of encrypted data sets when the key label is specified through a method outside of the DFP segment in the RACF data set profile.

To allow the system to create encrypted data sets using a key label specified through a method other than through the DFP segment in the RACF data set profile, the user must have at least READ authority to the following resource in the FACILITY class:

```
STGADMIN.SMS.ALLOW.DATASET.ENCRYPT
```

Note: IBM recommends that you define STGADMIN.* with UACC(NONE).

The system checks the user's authority to access this resource when a data set to be encrypted is first allocated (that is, created). It is not checked again before encrypting a data set.

Allocation processing

SMS allocation processing determines if a data set can be allocated as an encrypted data set. Under certain conditions, you can specify how the allocation should proceed. The following tables summarize the system behavior during SMS allocation processing for a new data set based on specific FACILITY class resources and the user's authorization to the resource.

On a z/OS V2R2 system (with OA50569) and later, the following table describes the result of an allocation request for an extended format data set when a key label has been specified. On a successful allocation, the resulting data set will be an encrypted extended format data set. Other factors not described in this table (such as lack of space) might cause the allocation to fail.

Table 2. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R2 with OA50569)

FACILITY class resource	**STGADMIN.SMS.ALLOW.DATASET.ENCRYPT**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
Key label from DFP segment of RACF DS profile	Allocation continues with IGD17150I	Allocation continues with IGD17150I	Allocation continues with IGD17150I	Allocation continues with IGD17150I

Table 2. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R2 with OA50569) (continued)

FACILITY class resource	**STGADMIN.SMS.ALLOW.DATASET.ENCRYPT**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
Key label from a source other than DFP segment of RACF DS profile	Allocation fails with IGD17155I	Allocation fails with IDC3038I	Allocation continues with IGD17150I	Allocation continues with IGD17150I

On a z/OS V2R2 system (with OA50569) and above, the following table describes the result of an allocation request for a DASD non-extended format data set when a key label has been specified from any source. On a successful allocation, the resulting data set will be a non-encrypted non-extended format data set. Other factors not described in this table (such as lack of space) might cause the allocation to fail.

Table 3. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R2 with OA50569)

FACILITY class resource	**STGADMIN.SMS. FAIL.INVALID.DSNTYPE.ENC**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
SMS mgd	Allocation continues with IGD17156I	Allocation continues with IGD17156I	Allocation fails with IGD17151I	Allocation fails with IGD17151I
non-SMS mgd	Allocation fails with IGD17156I	Allocation fails with IDC3040I	Allocation fails with IGD17151I	Allocation fails with IDC3039I

On a z/OS V2R1 system (with OA50569), the following table describes the result of an allocation request for a DASD data set (extended format and non-extended format) when a key label has been specified from any source. On a successful allocation, the resulting data set will be a non-encrypted data set since you cannot create new encrypted data sets on a z/OS V2R1 system. Other factors not described in this table (such as lack of space) might cause the allocation to fail.

Table 4. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R1 with OA50569)

FACILITY class resource	**STGADMIN.SMS. FAIL.INVALID.DSNTYPE.ENC**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
Key label specified on JCL (DSKEYLBL)	Job fails with IEFC630I	N/A	Job fails with IEFC630I	N/A
Key label specified on IDCAMS DEFINE (KEYLABEL)	N/A	Allocation fails with IDC3211I	N/A	Allocation fails with IDC3211I

Table 4. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R1 with OA50569) (continued)				
FACILITY class resource	**STGADMIN.SMS. FAIL.INVALID.DSNTYPE.ENC**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
Key label from DFP segment of RACF DS profile or from data class)	Refer to the following two rows.	Refer to the following two rows.	Refer to the following two rows.	Refer to the following two rows.
SMS mgd	Allocation continues with IGD17156I	Allocation continues with IDC3040I	Allocation fails with IGD17154I	Allocation fails with IDC0017I
non-SMS mgd	Allocation continues with IGD17156I	Allocation continues with IDC3040I	Allocation fails with IGD17154I	Allocation fails with IDC0017I

Accessing encrypted data sets

Applications that use standard BSAM, QSAM, and VSAM APIs do not require changes to access encrypted data sets. The user data transferred between the application and the access methods is in the unencrypted form. The access method encrypts the data when writing to DASD and decrypts the data when reading from DASD. For encrypted compressed format data sets, the access method compresses the data before encrypting it on output. On input, the access method decrypts the data before decompressing it.

Considerations regarding ICSF startup and shutdown

If you plan to encrypt SMF data sets or other data sets used during z/OS initialization, you must ensure that ICSF is started early in the IPL process to avoid delays in z/OS initialization and termination.

The IEASYSxx ICSF= parameter specifies the suffix of the ICSF on z/OS, CSFPRM*xx*, to be used by the system. The two characters represented by *xx* are appended to CSFPRM during ICSF initialization to form the name of the CSFPRM*xx* member. When specified in conjunction with the ICSFPROC parameter, ICSF on z/OS starts at IPL-time. The IEASYSxx ICSFPROC= parameter specifies the name of the ICSF procedure you want the system to use to automatically start ICSF on z/OS at IPL-time. The procedure must reside in a MSTJCL*xx* IEFPDSI PROCLIB data set. When specified in conjunction with the ICSF parameter, ICSF on z/OS starts at IPL-time.

Furthermore, during z/OS system shutdown, ICSF must be one of the last features to stop so that dependent functions are not impacted. It is highly recommended that you shut down ICSF after terminating the JES address space and after initiating SMF halt processing. Note when ICSF is stopped after SMF is halted, that there might not be an SMF record cut for the termination of ICSF. (The ability to start ICSF with SUB=MSTR is available on all supported releases of ICSF.)

Considerations regarding backup/migration/replication functions

Key label authorization

The following system functions maintain data in the encrypted form. Therefore, users performing these functions do not require authorization to the key label associated with the data sets being processed with these functions:

- DFSMSdss functions: COPY, DUMP, and RESTORE

- DFSMShsm functions: migrate/recall, backup/recover, abackup/arecover, dump/data set restore, FRBACKUP/FRRECOV DSNAME
- Track based copy (PPRC, XRC, FlashCopy®, concurrent copy) operations

Other considerations

- DFSMSdss REBLOCK keyword is ignored on COPY and RESTORE functions for encrypted data sets.
- DFSMSdss ADRREBLK installation exit will not be called for encrypted data sets.
- DFSMSdss does not support VALIDATE processing when backing up encrypted indexed VSAM data sets. VALIDATE will be ignored.
- Backup and migration of encrypted data sets may impact expected savings with disk or tape device compression. Where possible, convert to compressed format data sets. When data set level compression requested, access methods handle compression before encryption for compressed format encrypted data sets.

Considerations when planning for data set encryption

- Encrypted data sets must be extended format. Refer to *Extended-Format VSAM Data Sets* and *Processing Extended-Format Sequential Data Sets* for information on allocating extended format data sets, including guidelines and restrictions.
- Sequential extended format data sets can be version 1 or 2. When allocating an encrypted sequential extended format data set, the system creates the data set as an extended format version 2 data set, regardless of the user's specification for version number on DSNTYPE or the PS_EXT_VERSION keyword in the IGDSMSxx member in PARMLIB.
- Encrypted data do not compress. Therefore, encrypted data sets might not benefit from storage savings when relying on disk and tape device compression, as well as other areas in the infrastructure that attempt to compress data. Where possible, consider using compressed format data sets. (Note that only sequential extended format data sets and VSAM extended format KSDS data sets can be compressed format.) When data set level compression is requested along with encryption, access methods handle compression before encryption for encrypted compressed format data sets. Refer to *Compressed Data* and *Allocating Compressed-Format Data Sets* for information on allocating compressed format data sets, including guidelines and restrictions.
- When processing encrypted data sets, the access methods may use additional internal system buffers during read and write processing. For optimal performance, consider increasing memory to your user address space as your use of encrypted data sets increases.

Restrictions for encrypted data sets

- System data sets (such as Catalogs, SMS control data sets, SHCDS, HSM data sets) must not be encrypted, unless otherwise specified in product documentation.
- Data sets used during IPL must not be encrypted.
- Encrypted data sets only supported on 3390 device types.
- Sequential (non-compressed) data sets with a BLKSIZE of less than 16 bytes cannot be encrypted.
- Program objects cannot reside in encrypted PDSEs.

Additional restrictions for basic and large format encrypted data sets

- The minimum size of any user block written to an encrypted data set is 16 bytes. Using BSAM or QSAM, an attempt to write a block less than 16 bytes will fail with ABEND 002-F2.
- The minimum DCB LRECL is 16 bytes for RECFM (F(B(S)) and 12 bytes for RECFM V(B(S)). An attempt to open a data set with LRECL less than the minimum supported will fail with ABEND 213-89.
- Encrypted basic and large format data sets do not support hardware keys (DCBKEYLE). An attempt to open a data set with a non-zero key length will fail with 213-99. Diagnostic information in the message will clearly identify the reason for this failure.

- BDAM does not support processing encrypted basic and large format data sets. An attempt to open an encrypted basic or large format data set with BDAM will fail with 213-99. Diagnostic information in the message will clearly identify the reason for this failure.
- Checkpoint/Restart does not support open encrypted basic or large format data sets. Also, checkpoint data sets cannot be encrypted. An attempt to take a checkpoint when an encrypted basic or large format data set is open or when the checkpoint data set is encrypted, will fail with return code 08 and reason code 568.

For potential restrictions associated with other products, consult the corresponding product documentation.

See the following publications for more details on data set encryption support:

References:

z/OS DFSMS Using Data Sets

z/OS DFSMS Access Method Services Commands

z/OS DFSMSdfp Storage Administration

z/OS DFSMSdss Storage Administration

z/OS DFSMShsm Storage Administration

z/OS Cryptographic Services ICSF Administrator's Guide

z/OS Cryptographic Services ICSF System Programmer's Guide

Chapter 5. Using the catalog enhancements

z/OS V2R4 includes the following new functions for DFSMS catalogs.

- **IDCAMS enhancement**:

  The DEFINE CLUSTER command has a new zFS parameter, which specifies that the cluster being defined is for linear data, and the linear data set is a z/OS UNIX file system. When zFS is specified, the linear data set is defined as extended addressable. LISTCAT will display

  ```
  ZFS
  ```

  whenever the data set is LINEAR and zFS indicator flag is on, or the zFS initialized flag is on.

- **Catalog Forward Recovery Enhancement**: There are multiple enhancements for DFSMS catalogs:

  This enhancement allows for a complete point-in-time recovery of a catalog, where all data is restored with consistency. MVS system logger REDO log records are used for logging updates. Recovery can be automated with the CICSVR utility. For more information, refer to the Procedure for Forward Recovering a Catalog in *z/OS DFSMS Managing Catalogs*.

Chapter 6. Using the Object Access Method (OAM) enhancements

z/OS DFSMS V2R4 provides the following enhancements to the object access method (OAM):

- The LIBRARY DISABLE,CATCOUNT command can be used to disable the category count I/O call to the library for transitions from private to scratch. Other transitions from scratch to private (during job processing) continue to issue the category count call to the library. Today, this I/O is done for every volume that is being returned to scratch. By default there is a monitoring task that obtains the scratch category counts every 10 minutes. As job processing continues to do this I/O, the elimination of this extra I/O might improve performance during return to scratch processing. To re-enable this I/O call LIBRARY RESET,CATCOUNT can be issued.

  This support was introduced with APAR OA48240.
- The LIBRARY DISPDRV command is enhanced with new MOUNTED and MOUNTED,ALL parameters to optionally display:
  - Status information for volumes that are mounted in the TS7700 Virtualization Engine for a specified composite or distributed library. Information pertaining to the distributed library that owns the device for the mount as well as distributed library information associated with the primary and the secondary tape volume cache (TVC) is displayed.
  - For a distributed library in a TS7700 Virtualization Engine, additional drives that are not owned by the specified distributed library can be displayed using MOUNTED,ALL. The additional drives are displayed if the distributed library specified is the primary or secondary TVC for the mounted volume. This option can be useful when a distributed library is going into service and can be used to determine if the specified distributed library is associated with a mount request from another distributed library.

  This support was introduced with OA47487 (z/OS V1R13 and above).

  For more information, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* .
- The IGDSMSxx member of PARMLIB keywords DB2SSID, OAMTASK, and OAMPROC are enhanced in support of a new optional multiple OAM configuration. In a classic OAM configuration, the DB2SSID, OAMTASK, and OAMPROC keywords can continue to be specified in the IGDSMSxx member of PARMLIB as in the existing environment. However, in a multiple OAM configuration, changes are made to the existing OAM related keywords and a specification of the DB2SSID keyword is ignored.
- The IEFSSNxx member of PARMLIB is enhanced in support of a new optional multiple OAM configuration. In a classic OAM configuration, a single OAM subsystem can continue to be configured in the IEFSSNxx member of PARMLIB. While in a multiple OAM configuration, multiple OAM Object subsystems can be configured in addition to a single OAM Tape Library subsystem. A new D= keyword is used to specify the SSID or Group Attachment Name of the Db2 subsystem associated with the OAM subsystem being defined , and is required in a multiple OAM configuration. The IEFSSNxx member of PARMLIB keyword TIME is also enhanced. Either GMT or LOC can be specified as values for the TIME keyword. However, LOC is the default and indicates that local time should be used.
- The CBROAMxx member of PARMLIB is enhanced in support of a new optional OAM configuration. The ONLYIF statement can be used as before to specify whether specific statements within the CBROAMxx PARMLIB member are to be processed on a given system and can now also be used to specify whether statements are to be processed by a given OAM address space. In a multiple OAM configuration, optical related keywords on the SETOPT statement are validated but not used. Similarly, the CBROAMxx member of PARMLIB keyword SETOSMC is also enhanced so that optical related keywords on the statement are ignored in a multiple OAM configuration. And, in the CBROAMxx member of PARMLIB, optical related keywords on the SETOSMC statement are ignored in a multiple OAM configuration.

- The OAM Startup Procedure is enhanced in support of a new optional multiple OAM configuration. A new D= parameter identifies the SSID or Group Attachment Name of the Db2 subsystem to be associated with an OAM Object address space or indicates that the address space is a Tape Library address space by specifying D=NONE. D= is required in a multiple OAM configuration and not allowed in a classic OAM configuration. Additionally, multiple individual OAM procedures can be created (OAMA, OAMB), one for each OAM address space to be started in a multiple OAM configuration. Guidance on using the new D= parameter is added to the CBRAPROC member of SAMPLIB.
- The OAM Operator commands MVS MODIFY and MVS DISPLAY are enhanced in support of a new optional multiple OAM configuration. Guidance is provided on interacting with a specific OAM address space (F OAMA) using the MVS MODIFY command as well as multiple OAM address spaces using wildcarding. Guidance is also provided on the commands that can be used in a multiple OAM configuration as well as commands that can be used in a classic OAM configuration. OTIS supports new keywords on the MODIFY command to allow the operator to remove one or more OAM subsystems from the OAM configuration or to re-establish a connection to a Db2 subsystem. The MVS DISPLAY command is enhanced to display the overall OAM topology to assist with operations.
- The OSREQ API changes are enhanced in support of a new optional multiple OAM configuration. The OSREQ API is changed to provide a new DB2ID keyword to specify the Db2 subsystem (for a single OAM) or Db2 Group Attachment Name (for an OAM that is part of an OAMplex) to be used. The OSREQ sample program CBROSR2 remains unchanged for use with a classic OAM configuration, and a new sample program CBROSR3 is added for a multiple OAM configuration allowing for the specification of the DB2ID value to be used for a set of requests in a multiple OAM configuration.
- The TSO/E OSREQ Command processor is enhanced to allow specification of the DB2ID value in a multiple OAM configuration. DB2ID specifies the Db2 subsystem ID (for a single OAM) or Db2 group attachment name (for an OAMplex) that an OAM subsystem (and optionally a started OAM address space) has been configured to use. In a multiple OAM configuration, the DB2ID keyword is required for all OSREQ TSO/E Command Processor functions.
- The CBRSAMUT and CBRAPROC members of PARMLIB are enhanced to provide guidance on using the new DB2ID keyword in a multiple OAM configuration. The new CBROSR3 member of PARMLIB is created to show how to use OSREQ with the new DB2ID keyword in a multiple OAM configuration.
- The OAM SMF records are enhanced in support of a new optional multiple OAM configuration. OAM SMF records are updated to also include the actual OAM subsystem ID (OAM1,OAM2) as well as the associated Db2 subsystem ID (DB2A, DB2B).
- The OAM collection related function OAMUTIL CHGCOL is enhanced to include a new DB2ID keyword to identify the specific Db2 subsystem containing the Db2 Collection Name Table to be modified. Changes to the MVS catalog are no longer done since OAM collections are no longer maintained in the catalog.

Chapter 7. Using the new VSAM functions

z/OS V2R4 introduces the following enhancements for VSAM:

- DFSMS supports the storage of BSON and JSON objects (documents) into VSAM data sets. The changes to VSAM RLS are as follows:
  - You can specify NoSQL databases (known as VSAMDBs) to contain such objects using the new DATABASE parameter on the DEFINE CLUSTER command.
  - A new KEYNAME parameter on the DEFINE CLUSTER command, only specifiable with DATABASE, lets you specify a primary key for indexing documents in the database.
  - New ALTKEYS and ALTKEYSU parameters on the DEFINE ALTERNATEINDEX command allow you to specify alternate key names for the AIX against a base cluster with DATABASE, and the sorting order (ascending or descending) for each alternate key name.
  - New MACRF keyword values (DB or NODB) on the ACB, GENCB ACB, and MODCB ACB macros also give you the option to have RLS process the contents of a VSAM database as a regular KSDS, or as a database.
  - New OPTCD keyword values ARA31 or ARA64 (mutually exclusive) on the RPL, GENCB RPL, and MODCB RPL macros allow VSAM RLS records or VSAMDB documents to be passed to or retrieved from VSAM RLS to an area above the 2GB address bar in the user's address space.
  - New OPTCD=FRD on the RPL, GENCB RPL, and MODCB RPL will position to the first record in a regular VSAM data set or the first document in a VSAMDB.

For more information, see Defining a VSAM database in *z/OS DFSMS Using Data Sets*. See also *z/OS DFSMS Access Method Services Commands* and *z/OS DFSMS Macro Instructions for Data Sets*.

Chapter 8. Using the SMS enhancements

In z/OS V2R4, SMS introduces the following enhancements.

Virtual storage constraint relief

Prior to z/OS V2R4, each online DASD volume obtained a volume statistics block (VSB) which consumed 616 bytes of 31-bit common storage. In z/OS V2R4 (or earlier releases with APAR OA55711), each online DASD volume consumes 48 bytes in 31-bit common storage and an additional 896 bytes above the 2 G bar. This change provides virtual storage constraint relief.

New SMS ACS read-only variables

SMS provides several new read-only variables for z/OS V2R4 and later. They are used with the ACS routines, which allows more flexibility and control when writing ACS routines. The new ACS variables are as follows:

- &TIME
- &DATE
- &STEP
- &DAYOFWEEK
- &DSN_VERSION
- &MAXGENS

For more information about ACS routines and the new ACS read-only variables, see Defining ACS routines and Writing ACS routines in *z/OS DFSMSdfp Storage Administration*.

Multi-release toleration:

Do not reference these new read-only variables in releases earlier than z/OS V2R4. The ACS routines programmer needs to add logic to jump around references of these new R/O variables to prevent earlier releases from failing during ACS processing. If failure occurs on an invalid ACS R/O variable, existing return code 20 and reason code 2035 is issued.

Chapter 9. Using the DFSMSrmm enhancements

The functional enhancements available with z/OS V2R4 DFSMSrmm provide you with these benefits:

RMM Mandatory Defaults Table Enhancements

In V2R4, several improvements were made to the defaults table. It is now possible to combine several defaults table entries using the CONTINUE keyword. This keyword allows to create sections of the defaults table dedicated to setting a single parameter - "CONTINUE" ensures that the other defaults table entries are still processed to set the remaining parameters.

In addition, support for scratch pooling has been added to the defaults table. The new defaults table POOL parameter works exactly the same way as the POOL parameter in the UXTABLE, and allows to convert setups that still rely on the UXTABLE due to its pooling capabilities over to the defaults table. Scratch pooling is useful for non-SMS managed tape libraries or for manual tape libraries, and allows RMM to select which pool of volumes should be used to satisfy a particular scratch mount request. For automatic and virtual SMS-managed libraries, scratch pooling is done by having the ACS routines select an appropriate storage group, and there is no need to use the defaults table.

The PGMNAME keyword has been added to the defaults table. It allows to set retention parameters of tape volumes and data set based on the program name that is used to write the data set.

The new EDM keyword has also been added to the defaults table. It allows to fine tune which volumes are to be marked with the EDM flag, which means that the volume is managed by an External Data Manager, and that RMM is not to scratch the volume until the External Data Manager releases it. It is possible to specify programs that manage their own tape data by using a combination of the PGMNAME and EDM keywords in the defaults table. It is also possible to use EDM(NO) to make exceptions, and mark certain volumes which are written by HSM, OAM and TSM, and would have been marked as EDM volumes by default, as non-EDM managed.

RMM RAS Enhancements

A new ONLYIF command is now available to be used in the EDGRMMxx and EDGDEFxx PARMLIB members, when defining the settings for RMM. ONLYIF makes it easy to share a single PARMLIB data set among multiple instances of RMM on a SYSPLEX. The ONLYIF command causes the command that follows ONLYIF to only be executed on the system that has the same system name as that specified in the ONLYIF command.

The report generator in DFSMSrmm can now work with dates that are in European and American formats, and allows them to be sorted correctly. To use fields with these dates, first mark the fields that contain a date in the report definition, by pressing R next to the field and setting Y next to "Field contains date".

In V2R4, search support for the EDM bit has been added. EDM is an attribute that shows that a volume is managed by an external program, and prevents RMM from scratching the volume. It is now possible to search for EDM volumes using the RMM SEARCHVOLUME EDM subcommand. It is also possible to see EDM information in the report extract data set that is created by RMM housekeeping.

Also, new messages have been added to alert the users when the RMM subsystem interface is down, to prevent data loss. For RMM to be able to function, both the RMM started task needs to be active and the RMM subsystem interface needs to be initialized. If for any reason, RMM is not fully active, warning messages are issued whenever tape activity occurs. To completely stop RMM (and prevent warning messages from being shown), both the RMM Started task must be stopped, and the subsystem interface uninitialized. For example, commands such as the following can be used: "S DFRMM,OPT=RESET" and "P DFRMM".

A new warning message has been added (EDG2120W) warning the user that the RMM CDS is about to fill up. It is possible to set a threshold for the CDS utilization using the new CDSFULL PARMLIB command. Once the specified percentage is reached, the warning message will be issued. It will also be issued for

each additional percent that is increased. The CDS utilization is calculated as the High Used RBA divided by the High Available RBA. This is the best possible indicator of possible shortage of space in the CDS, even though it is still possible to insert records into a CDS that has 100% utilization.

To help clients comply with the new European GDPR legislation, that has a focus on forgetting data, several new options were added to DFSMSrmm. The FORCEEXPIRE parameter of the RMM CHANGEDATASET subcommand allows to manually block access to selected data sets. The DSNEXPIRE RMM option allows to block access to all data sets that are expired and not retained by a Vital Record Selection policy, even if the volume on which they reside is not yet moved to scratch status.

A new sample has been added to SYS1.SAMLPIB, which can be used to create a backup of the currently defined Vital Record Selection policies. The REXX script is called EDGMKVRS, and it can be invoked using the sample JCL EDGJMVRS, which is also located in SYS1.SAMPLIB. The script reads the RMM report extract data set, which is created by RMM housekeeping. It outputs a list of ADDVRS commands that can be executed to recreate the policies.

Enhancements to EXPDT retention method

DFSMSrmm has been enhanced with these new functions:

- *Specify expiration time for added volumes and data sets:* The ADDDATASET, ADDVOLUME, and GETVOLUME subcommands 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.
- *Specify that an added data set not expire while cataloged:* A new WHILECATALOG option has been added to the EDGRMMxx parmlib member OPTION command and to the ADDDATASET TSO subcommand to enable the user to specify either that:
  - the added 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 added dataset will be kept as long as it is cataloged, but no later than the expiration date.

Chapter 10. Using the DFSMStvs enhancements

In z/OS V2R4, DFSMStvs is enhanced with the following new function:

- **Automatic commit support**

  DFSMStvs provides a new parameter on the JCL EXEC statement that you can use to enable the automatic issuing of commit points on behalf of a batch application. The TVSAMCOM parameter lets you specify how many updates (an exact number or a range of numbers) should occur before a commit point is issued. You can also specify a default value for this parameter in the **IGDSMSxx** member of SYS1.PARMLIB (the EXEC statement value takes precedence over values in IDGSMS*xx*).

  For more information about using automatic commit, including the types of batch jobs that are good candidates for it, see *z/OS DFSMStvs Planning and Operating Guide*.

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

Tasks	Procedure that you must perform:
"Programming" on page 77	• "Using the TVSAMCOM parameter on the JCL EXEC statement to enable automatic commits in batch jobs" on page 77 • "Specifying a TVSAMCOM value in IGDSMSxx" on page 77

Part 4. Using new DFSMS functions in z/OS V2R3

These topics describe how to use new DFSMS functions in z/OS V2R3.

Chapter 11. Using DFSMS transparent cloud tiering

Cloud computing is a term that describes a new model for delivering services to consumers. One such service is data storage and retrieval for use of long-term storage. Using a cloud storage service, consumers are able to store and retrieve data from anywhere in the world on any device using internet protocols. Providers of a cloud storage service are able to provide broad network access, pool storage resources, and rapidly respond to growing storage requirements. Because of the value that cloud service delivery provides, enterprise IT organizations are adopting this model within their own data centers, which are known as private clouds, to deliver storage services to their own internal organizations. Release 7.5 of the IBM System Storage DS8000 introduces the ability to store data to and retrieve data from a cloud.

For information about the IBM Object Cloud Storage offering, see IBM Cloud Object Storage (www.ibm.com/cloud/object-storage).

z/OS DFSMS introduces transparent cloud tiering, to enable data storage on a cloud in combination with support provided by the IBM System Storage DS8000. The following DFSMS cloud storage functions are included in transparent cloud tiering:

- Cloud connection definition using ISMF and SMS
- Migration and recall using DFSMShsm
- Full volume dump and restore using DFSMShsm
- DFSMShsm Audit support
- Command-based backup and recovery to a cloud using DFSMSdss.

To prepare your system for Cloud data storage, perform the following steps:

1. Enable cloud functions on your DS8000, as specified in the DS8000 information. For information about setting up your DS8000, see Transparent Cloud Tiering (www.ibm.com/docs/en/ds8870/7.5.0?topic=reference-transparent-cloud-tiering).

   Note: Initially, the DS8000 only supports a single account. With only a single account, there is no way to separate test data from production data. Additionally, there is no way to separate HSM data from DSS data. It is recommended that using both DFSMShsm and DFSMSdss for the same account should only be done in a test environment. DFSMSdss and DFSMShsm should not share a single account in a production environment.
2. Ensure that TCP/IP is enabled and active on your z/OS system.
3. Obtain the following information from your cloud provider or administrator:
   - Provider: SWIFT (if your cloud is an Openstack-based Swift object storage cloud using the built-in Tempauth authorization system) or SWIFT-KEYSTONE (if your cloud is an Openstack-based Swift object storage cloud using the Keystone Identity service).
   - Identity: a concatenation of the tenant name and a user name in the form *tenant_name:user_name*.

     In order to secure the cloud password, DFSMShsm data must have its own account and should not be intermixed with other data.
   - Endpoint: the Uniform Resource Identifier (URI) that DFSMS should use for authentication with your object storage cloud.
   - Port number: the port on which DFSMS is to communicate with the endpoint.
   - SSL version: the lowest acceptable level of SSL/TLS that should be used when establishing a secure connection.
4. Digital certificate setup for SSL/TLS: if your cloud provider uses SSL/TLS communication, then you must also obtain the necessary digital certificates and understand the type of SSL/TLS authentication the cloud performs. For more information about these steps, see
5. Perform SMS setup:

- (optional) Allocate a backup SMS configuration data set (BACKUP.SCDS), a backup active configuration data set (BACKUP.ACDS), and a new active configuration data set (NEW.ACDS) that you will use for a new SMS configuration.
- Make a backup of your existing SMS configuration, using the SETSMS SAVESCDS command.
- Make a backup of your existing active SMS configuration data set, using the SETSMS SAVEACDS command.
- Use new ISMF Cloud panels to define cloud construct(s) in the active SMS source configuration data set (required information: cloud name, provider, identity (credentials), endpoint, port, SSL version, SSL key). Select 'S' on the **ISMF Primary Option Menu** panel to access the cloud definition panels.
- Validate the new SMS configuration in the active source configuration data set, using ISMF.
- Copy the active SCDS into the new ACDS, using the SETSMS COPYSCDS(active.scds , new.acds) command, where active.scds is the SCDS with defined cloud constructs.
- Activate the new SMS active configuration data set, using the SETSMS ACDS(new.acds) command.

6. Use z/OS Cloud Data Access Authorization Utility to protect cloud credentials (**Optional**)
   - Create a gdk directory and copy setup files from /u/lpp/dfsms/samples to your working directory:
     - mkdir gdk; cd gdk;
     - cp /usr/lpp/dfsms/samples/gdkconfig.json config.json
     - cp /usr/lpp/dfsms/samples/gdkkeyf.json gdkkeyf.json
   - Create a 'providers' directory and an empty file within that directory that corresponds to the SMS cloud network connection name.
   - From TSO, execute SYS1.SAREXEC(GDKAUTHP) to reveal the z/OS Cloud Data Access Authorization Utility:
     - Setup credentials for the desired provider:
       - In the Select Cloud Provider section, choose provider by selecting from the enumerated list and use option O to open credential entry panel.
       - In the Authorization Parameters section, populate the key and secret key and use option S to save the resource authorization information.
   - In DFSMSdss commands, provide the credentials using CDACREDSTORE instead of the CLOUDCREDENTIALS keyword.
   - In the DFSMShsm SETSYS CLOUD command, specify CDACREDENTIALS instead of the CLOUDCREDENTIALS keyword to request DFSMShsm to use a password that has been secured within CDA.
7. SDM setup: Define SAF resources to control access to the OBJSTORE ILK on the ANTRQST API. It is recommended that the following FACILITY class profile be defined with a universal access of NONE:
   - STGADMIN.ANT.ESS.OBJSTORE
8. DFSMSdss setup:

   Define SAF resources to control access to the CLOUD keyword on the DFSMSdss DUMP and RESTORE commands, by granting READ access to FACILITY class profiles:

   a. STGADMIN.ADR.DUMP.CLOUD

   b. STGADMIN.ADR.RESTORE.CLOUD
9. DFSMShsm setup:

   Enable the CP Assist for Cryptographic Functions (required by DFSMShsm).

   Enable the CP Assist for Cryptographic Functions.

   Define DFSMShsm to z/OS UNIX System Services as a superuser, and the RACF user ID must have a default RACF group that has an OMVS segment with a group ID (GID). The user ID must also have an OMVS segment with the following parameters:

- UID(0) HOME('/')

Define SAF resources to control access to the CLOUD keyword on the HMIGRATE end-user command, by granting READ access to the following FACILITY class profile:

- STGADMIN.ARC.ENDUSER.HMIGRATE.CLOUD

Define SAF resources to control access to the DDELETE command by granting READ access to FACILTY class profile:

- STGADMIN.ARC.DDELETE

Set the password that DFSMShsm uses when communicating with the Cloud Storage account.

a. CLOUDCREDENTIAL option:
   - Issue the following operator command:
     - SETSYS CLOUD(NAME(*cloud_name*) CCREDS)
   - The system will prompt for the cloud password in a WTOR and the response will not be echoed back to the console. If the cloud password is case sensitive, use the System Command Extension from the SDSF log when replying to the WTOR. The System Command Extension can be started by typing a forward slash (/) on the Command Input line in the SDSF system log.
   - The cloud password must be kept very secure. IBM recommends that the Security Administrator who is administering the cloud storage accounts be the one who enters this password.

b. CDACREDENTIALS option:
   - Issue the following operator command:
     - SETSYS CLOUD(NAME(*cloud_name*) CDACREDS)

If you plan to use DFSMShsm with Cloud storage, DFSMShsm must have a RACF user ID with an OMVS segment associated with it.

- DFSMShsm must be defined to z/OS UNIX System Services as a superuser, and the RACF user ID must have a default RACF group that has an OMVS segment with a group ID (GID). The user ID must also have an OMVS segment with the following parameters:
  - UID(0) HOME('/')

Define SAF resources to control access to the CLOUD keyword on the HMIGRATE end user command, by granting READ access to FACILITY class profile:

a. STGADMIN.ARC.ENDUSER.HMIGRATE.CLOUD

Set the password that DFSMShsm uses when communicating with the Cloud Storage account:

a. Issue the following operator command:
   - SETSYS CLOUD(NAME(*cloud_name*) CCREDS)
b. The system will prompt for the cloud password in a WTOR and the response will not be echoed back to the console. If the cloud password is case-sensitive, use the System Command Extension from the SDSF log when replying to the WTOR. (The System Command Extension can be started by typing a forward slash (/) on the Command Input line in the SDSF system log.)
c. The password must be kept very secure. IBM recommends that the Security Administrator who is administering the Cloud Storage accounts be the one who enters this password.

Note: RACF user ID and/or user ID refers to DFSMShsm Started Task user ID.

10. Enable status messages: ENABLE(AOM496I) in DEVSUP*xx*, when exceptions need to be looked at.

    Note: The default is DISABLE(AOM496I).
11. Inform users about cloud storage functions, including the new DFSMSdss and DFSMShsm commands.

Using Cloud storage functions on DFSMS

SMS provides a new construct called a cloud, which contains a list of cloud attributes and their values. DFSMShsm and DFSMSdss use the attributes of a specified cloud definition on their commands to manage data set migration. A new ISMF option 'S' on the ISMF Primary Option Menu panel leads to a new Cloud Application Selection panel where storage administrators can select functions to define a cloud, display or alter a cloud, or generate a list of clouds. For details, see *z/OS DFSMSdfp Storage Administration*.

DFSMShsm commands -- for details, see *z/OS DFSMShsm Storage Administration*:

- The MIGRATE and HMIGRATE commands include a new optional keyword, CLOUD(*cloudname*). When this keyword is specified, the named data set, or data sets on the namedvolume is migrated to the requested cloud. The *cloudname* specified must match an existing SMS Cloud definition.
- The DEFINE DUMPCLASS command includes new optional keywords, TAPE | CLOUD(*cloud_network_connection_name*) and UNASSIGNTAPE. TAPE| CLOUD specifies whether the volumes should be dumped to tape or offloaded to the designated cloud object store. UNASSIGNTAPE specifies that DFSMShsm unassign any dump volumes that are empty and are currently added to this (existing) dump class.
- The BACKVOL DUMP, RECOVER FROMDUMP, FRBACKUP DUMP | DUMPONLY, FRRECOV FROMDUMP, and automatic dump functions are enhanced to dump to or restore from cloud object store when designated by a CLOUD dump class. Currently, if multiple dump copies are made concurrently, each copy is associated with a different dump class. With this enhancement, DFSMShsm continues to support up to 5 TAPE dump classes in a dump generation. The new function will support a single CLOUD dump class in a dump generation. Multiple CLOUD dump classes and mixed CLOUD and TAPE dump classes within a dump generation will not be supported. Data set restore from a full-volume CLOUD dump copy is not supported.
- The DDELETE command can be used to delete specified non-copy pool dump copies residing in cloud storage. This command is intended for occasional cloud dump copy deletion and is not designed for bulk deletion. Use of automatic dump expiration to manage dumps and empty dump container deletion is recommended.
- The LIST command processing is updated to display information about data sets migrated to cloud storage, volumes dumped to cloud storage, as well as the cloud storage that DFSMShsm has used. The CLOUD optional keyword for DATASET SELECT will cause DFSMShsm to list only those data sets that have been migrated to cloud storage.
- The REPORT command is updated to include information about data sets migrated to cloud storage. New TOCLOUD and FROMCLOUD optional parameters on MIGRATION and RECALL (respectively) allow selection of records where cloud storage was involved.
- The SETSYS command is updated with optional parameters CLOUDMIGRATION, to allow specification of Fast Subsequent Migration for data set migrated to storage in the cloud, and CLOUD, to specify properties for a particular Cloud Storage that DFSMShsm uses to store migration copies of data sets. The SETSYS CLOUD command is updated with the CDACREDENTIALS option to request that DFSMShsm obtain the password for the cloud from the z/OS Cloud Data Access (CDA).
- AUDIT command processing is modified to include migrated data sets and full volume dumps that are stored in the Cloud. AUDIT MEDIACONTROLS adds a new CLOUD(*cloudname*) parameter. This function audits control information contained in migration copies that reside in the Cloud. The *cloudname* refers to a defined SMS cloud construct name.

DFSMSdss commands -- for details, see *z/OS DFSMSdss Storage Administration*:

- The DUMP and RESTORE commands have new keywords CDACREDSTORE, CLOUD, CLOUDCREDENTIALS, CONTAINER, and OBJECTPREFIX to support the placement of a DFSMSdss backup in cloud storage. The credentials specified in the CLOUDCREDENTIALS keyword must be kept secure. If there are batch JCL jobs that specify this keyword, then those jobs should be in a data set or library that has limited access and controlled by a security product. If the credentials cannot be kept secure, then this keyword should not be used. Using the new CDACREDSTORE keyword instead of the CLOUDCREDENTIALS keyword is recommended,

- A new command, CLOUDUTILS, provides LIST and DELETE operations which can be used to manage containers and backups created by DFSMSdss. Like the DUMP and RESTORE commands, it requires the CLOUD and either CLOUDCREDENTIALS or CDACREDSTORE keyword.

Tiering data to a TS7700

DFSMS can use a DS8000 to transparently tier data to a TS7700. To enable transparent cloud tiering to a TS7700, define a cloud construct using SWIFT authentication with the DS8000 as the endpoint. All object requests originating from DFSMS are routed through the DS8000 onto the TS7700. See *z/OS DFSMSdfp Storage Administration* for information on how to setup a DS8000 as an object proxy server. See hardware RPQ 8B3867 FC 0005 DS8000 Offload for more information about the support being provided.

Tiering data to an S3 compatible object storage cloud

DFSMS does not natively support clouds that expose an S3 API, however, DFSMS can use a DS8000 to transparently tier data to an object storage cloud that exposes an S3 compatible API. To enable transparent cloud tiering to an S3 cloud, define a cloud construct using SWIFT authentication with the DS8000 as the endpoint. All object requests originating from DFSMS are routed through the DS8000 onto the S3 cloud. See *z/OS DFSMSdfp Storage Administration* for information on how to setup a DS8000 as an object proxy server.

Chapter 12. Using the z/OS data set encryption enhancements

Introduction

With z/OS data set encryption, you can encrypt data without requiring application changes. z/OS data set encryption, through SAF controls and RACF® or equivalent function along with SMS policies, allows you to identify new data sets or groups of data sets to be encrypted. You can specify encryption key labels to identify encryption keys to be used to encrypt selected data sets. The specified key label and encryption key must exist in the ICSF key repository (CKDS). With data set encryption, you are able to protect data residing on disk from being viewed by unauthorized users in the clear. Authorization is based on access to the key label that is associated with the data set and used by the access methods to encrypt and decrypt the data.

z/OS data set encryption provides the ability to encrypt the following types of data sets:

- Sequential extended format data sets, accessed through BSAM and QSAM
- VSAM extended format data sets (KSDS, ESDS, RRDS, VRRDS, LDS), accessed through base VSAM and VSAM RLS

Encrypted data sets must be SMS-managed extended format. They also can be compressed format.

Importance of host based compression
It is recommended that clients using host based encryption also exploit host based compression prior to the encryption of the data. If the data is not compressed prior to the encryption there can be consequences to other parts of the client infrastructure. For example, replicated data that is being compressed in the SAN infrastructure by DWDM technology will no longer be effective trying to compress encrypted data. Without compressing prior to the encryption of the data, additional bandwidth might be required. Another example might be that tape systems can require additional capacity in terms of disk space, in the case of virtual tape, or tape cartridges. If deduplication of data is supported, host encryption can prevent deduplication from working. Therefore, where possible, use compressed format data sets. With encrypted compressed format data sets, the access methods perform compression before encryption. See Considerations when planning for data set encryption for more information.

To create an encrypted data set, a key label must be supplied on new data set allocation. The key label must point to an AES-256 bit encryption DATA key within the ICSF key repository (CKDS) to be used to encrypt or decrypt the data. For each encrypted data set, its key label is stored in the catalog. The key label is not sensitive information; it identifies the encryption key, which is sensitive; therefore, IBM recommends only using secure keys. For more information, see *z/OS Cryptographic Services ICSF System Programmer's Guide*.

Hardware requirements

Exploitation of this function requires IBM Enterprise z196 or later. It also requires the following cryptographic hardware features:

- Crypto Express3 Coprocessor or later
- Feature 3863, CP Assist for Cryptographic Functions (CPACF)

Operating System requirements

- ICSF is installed and configured with a CKDS and AES master key loaded. See “Considerations regarding ICSF startup and shutdown” on page 53 for more information.

Coexistence requirements

- On a z/OS V2R2 system with OA50569, you can create encrypted data sets as well as access encrypted data sets created on a z/OS V2R2 (with OA50569) or later system.
- On a z/OS V2R1 system with OA50569, you cannot create encrypted data sets. However, you can access encrypted data sets created on a z/OS V2R2 (with OA50569) or later system.

  Note: The minimum software release that can support encrypted data sets is z/OS V2R1 with OA50569. An attempt to access an encrypted data set on a lower release will result in loss of access to the data. Ensure that all systems are at the minimum hardware and software levels before encrypting any data sets.

Before enabling this function

Because data set encryption has both hardware and software requirements, you must consider all systems that share data with a system on which you plan to enable data set encryption before creating an encrypted data set. This includes backout software levels, backup systems, read-only systems, replication target systems and disaster recovery systems. Before encrypting data sets other than those used for testing, be sure that all the systems that must access encrypted data sets are capable of doing so by meeting the required hardware and software requirements. In addition to the hardware and software requirements that must be available on every system that will access the encrypted data sets, all key labels and encryption keys associated with the encrypted data sets must also be available.

Take these steps to make data set encryption unavailable to users who are not explicitly authorized to use it:

- Define the STGADMIN.SMS.ALLOW.DATASET.ENCRYPT profile in the FACILITY class, and set the universal access to NONE:

```
RDEFINE FACILITY STGADMIN.SMS.ALLOW.DATASET.ENCRYPT UACC(NONE)
```

- If the FIELD class is active, check for any profile that would allow any user without SPECIAL attribute access to the DATASET.DFP.DATAKEY. If there are none, no additional action is needed. If there is any profile that would allow access to DATASET.DFP.DATAKEY, create a DATASET.DFP.DATAKEY profile in the FIELD class with a UACC of NONE:

```
RDEFINE FIELD DATASET.DFP.DATAKEY UACC(NONE)
```

Taking the steps above is intended to assure that only authorized users are allowed to use data set encryption. Such users should be made aware that until the decryption functions are available on all sharing systems, backup systems, and disaster recovery systems, access to encrypted data can be lost at any time.

Tasks for setting up data set encryption

Performing the following tasks is necessary to begin using data set encryption.

Create key labels and encryption keys

To create an encrypted data set, a key label must be assigned to the data set. The key label and its associated AES-256 bit encryption key must exist in the CKDS by the time the data set is opened. To create keys in the CKDS, refer to *z/OS Cryptographic Services ICSF System Programmer's Guide*. IBM recommends the use of secure keys.

Set up CSFKEYS

To control which users can use which keys, protect resources CSFKEYS class. The CSFKEYS class controls access to cryptographic keys identified by the key label.

To enable the use of ICSF keys:

1. Grant the user READ authority to the resource key-label in the CSFKEYS class. This authority can be granted for all uses of the key-label or only when the use of the key-label is permitted when using the a CRITERIA value of DSENCRYPTION for SMS.

   - if the class has not already been enabled for generics

   ```
   SETROPTS GENERIC(CSFKEYS)
   ```

   - Define profiles in the CSFKEYS class to protect key labels

   ```
   RDEFINE    CSFKEYS  profile-name UACC(NONE)
              ICSF(SYMCPACFWRAP(YES)   SYMCPACFRET(YES))
   ```

   - Give the users (preferably groups) access to the profiles

   ```
   PERMIT profile-name CLASS(CSFKEYS)
          ID(name)    ACCESS(READ)
   ```

   or

   ```
   PERMIT profile-name CLASS(CSFKEYS)
          ID(name)    ACCESS(READ)
          WHEN(CRITERIA(SMS(DSENCRYPTION)))
   ```

2. Define the ICSF information for the key with SYMCPACFWRAP(YES) and SYMCPACFRET(YES), if it was not specified on the DEFINE

   ```
   RALTER CSFKEYS profile-name
          ICSF(SYMCPACFWRAP(YES)   SYMCPACFRET(YES))
   ```

3. Set RACF options

   - if the CSFKEYS class is not already active

   ```
   SETROPTS CLASSACT(CSFKEYS)
   ```

   - if the CSFKEYS class has not already been RACLISTed

   ```
   SETROPTS RACLIST(CSFKEYS)
   ```

   - or if the CSFKEYS class has already been RACLISTed

   ```
   SETROPTS RACLIST(CSFKEYS) REFRESH
   ```

Set up CSFSERV

Important: The following setup is required only if CHECKAUTH(YES) is specified on the ICSF installation options data set. CHECKAUTH(NO) is the default.

Protect the resources CSFSERV class. ICSF controls access to cryptographic services through the CSFSERV resource class. For more information, see *z/OS Cryptographic Services ICSF Administrator's Guide* .

The following table summarizes the CSFSERV resource required for processing encrypted data sets.

Table 5. CSFSERV resource required for data set encryption

Function	ICSF callable service	Resource
CKDS Key Record Read2	CSNBKRR2	CSFKRR2

If the user does not have sufficient access, open processing will fail. An informational message ICH408I (which indicates insufficient authorization) might be issued.

1. Grant the user READ authority to the resource CSFSERV class:

- if the class has not already been enabled for generics

```
SETROPTS GENERIC(CSFSERV)
```

- Define profiles in the CSFSERV class to protect key labels

```
RDEFINE CSFSERV * UACC(NONE)
```

- Define profile CSFKRR2 if it does not exist

```
RDEFINE CSFSERV CSFKRR2 UACC(NONE)
```

- Give the users (preferably groups) access

```
PERMIT CSFKRR2 CLASS(CSFSERV) ID(groupid) ACCESS(READ)
```

2. Set RACF options:

   - if the CSFSERV class is not already active

```
SETROPTS CLASSACT(CSFSERV)
```

   - if the CSFSERV class has not already been RACLISTed

```
SETROPTS RACLIST(CSFSERV)
```

   - or if the CSFSERV class has already been RACLISTed

```
SETROPTS RACLIST(CSFSERV) REFRESH
```

Create an encrypted data set

To create an encrypted data set, you must assign a key label to the data set when it is newly allocated (data set create). A key label can be specified through any of the following methods:

- RACF data set profile
- JCL, dynamic allocation, TSO ALLOCATE, IDCAMS DEFINE
- SMS data class

To specify a key label using the DFP segment in the RACF data set profile, use keyword DATAKEY(Key-Label). The system will use this key label for extended format data sets that are created after DATAKEY is added to the data set profile. Use keyword NODATAKEY to remove a key label, if defined, from the RACF DFP segment. The key label is ignored for a data set that is not a DASD data set.

To specify a key label using JCL, dynamic allocation, or TSO allocate, use JCL keyword DSKEYLBL='key-label', dynamic allocation text unit DALDKYL, or TSO allocate DSKEYLBL(key-label). DSKEYLBL parameter is effective only if the new data set is on DASD. The key label is ignored for a data set that is not a DASD data set.

See details about the DSKEYLBL parameter(key-label) keyword on the JCL DD statement in *z/OS MVS JCL Reference*.

To specify a key label using SMS data class, use the *Data Set Key Label* field on the ISMF DEFINE/ALTER panel. The system will use this key label for extended format data sets that are created after the data set key label is added to the data class. The key label is ignored for a data set that is not a DASD data set.

See details on using the new Data Set Key Label field in the ISMF panels in *z/OS DFSMS Using the Interactive Storage Management Facility*.

To specify a key label using the DEFINE CLUSTER command for a VSAM CLUSTER, use the KEYLABEL parameter; for example, KEYLABEL(MYLABEL). Any alternate index associated with the CLUSTER will also be encrypted and use the same key label as specified for the CLUSTER. The key label is ignored for a data set that is not a DASD data set.

For more information on using the DEFINE CLUSTER command for a VSAM CLUSTER, see *z/OS DFSMS Access Method Services Commands*.

When a key label is specified on more than one source, the key label is derived from one of the above sources only on the initial data set allocation (on data set create). The key label is derived in the following order of precedence:

1. From the DFP segment in the RACF data set profile.
2. Explicitly specified on the DD statement, dynamic allocation text unit, TSO ALLOCATE command, or DEFINE CLUSTER control statement.
3. From the data class that applies to the current DD statement.

   Note: The REFDD and LIKE JCL DD statement keywords do not cause a key label from the data set referred to be used when allocating a new data set.

On successful allocation of an encrypted data set, the following message is issued:

```
IGD17150I DATA SET dsname IS ELIGIBLE FOR ACCESS METHOD ENCRYPTION
KEY LABEL IS (key_label)
```

Specifying a key label for a non-extended format data set

If an encryption key label is specified for a DASD data set that is not extended format, the key label is ignored and the data set is successfully created as non-encrypted non-extended format data set. In addition, SMS message IGD17156I is issued (or if using IDCAMS DEFINE and data set is non-SMS managed, message IDC3040I is issued) indicating that the key label is ignored. Instead to have the system fail the allocation, the user must have at least READ authority to the resource in the FACILITY class: STGADMIN.SMS. FAIL.INVALID.DSNTYPE.ENC

If this facility class resource exists and the user has at least read authority, SMS will fail the allocation and issue message IGD17151I (or if using IDCAMS DEFINE and non-SMS managed data set request, message IDC3039I is issued).

Enable data set encryption

An enablement action is required to allow the creation of encrypted data sets when the key label is specified through a method outside of the DFP segment in the RACF data set profile.

To allow the system to create encrypted data sets using a key label specified through a method other than through the DFP segment in the RACF data set profile, the user must have at least READ authority to the following resource in the FACILITY class:

```
STGADMIN.SMS.ALLOW.DATASET.ENCRYPT
```

Note: IBM recommends that you define STGADMIN.* with UACC(NONE).

The system checks the user's authority to access this resource when a data set to be encrypted is first allocated (that is, created). It is not checked again before encrypting a data set.

Allocation processing

SMS allocation processing determines if a data set can be allocated as an encrypted data set. Under certain conditions, you can specify how the allocation should proceed. The following tables summarize the system behavior during SMS allocation processing for a new data set based on specific FACILITY class resources and the user's authorization to the resource.

On a z/OS V2R2 system (with OA50569) and later, the following table describes the result of an allocation request for an extended format data set when a key label has been specified. On a successful allocation, the resulting data set will be an encrypted extended format data set. Other factors not described in this table (such as lack of space) might cause the allocation to fail.

Table 6. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R2 with OA50569)				
FACILITY class resource	**STGADMIN.SMS.ALLOW.DATASET.ENCRYPT**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
Key label from DFP segment of RACF DS profile	Allocation continues with IGD17150I	Allocation continues with IGD17150I	Allocation continues with IGD17150I	Allocation continues with IGD17150I
Key label from a source other than DFP segment of RACF DS profile	Allocation fails with IGD17155I	Allocation fails with IDC3038I	Allocation continues with IGD17150I	Allocation continues with IGD17150I

On a z/OS V2R2 system (with OA50569) and above, the following table describes the result of an allocation request for a DASD non-extended format data set when a key label has been specified from any source. On a successful allocation, the resulting data set will be a non-encrypted non-extended format data set. Other factors not described in this table (such as lack of space) might cause the allocation to fail.

Table 7. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R2 with OA50569)				
FACILITY class resource	**STGADMIN.SMS. FAIL.INVALID.DSNTYPE.ENC**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
SMS mgd	Allocation continues with IGD17156I	Allocation continues with IGD17156I	Allocation fails with IGD17151I	Allocation fails with IGD17151I
non-SMS mgd	Allocation fails with IGD17156I	Allocation fails with IDC3040I	Allocation fails with IGD17151I	Allocation fails with IDC3039I

On a z/OS V2R1 system (with OA50569), the following table describes the result of an allocation request for a DASD data set (extended format and non-extended format) when a key label has been specified from any source. On a successful allocation, the resulting data set will be a non-encrypted data set since you cannot create new encrypted data sets on a z/OS V2R1 system. Other factors not described in this table (such as lack of space) might cause the allocation to fail.

Table 8. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R1 with OA50569)				
FACILITY class resource	**STGADMIN.SMS. FAIL.INVALID.DSNTYPE.ENC**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
Key label specified on JCL (DSKEYLBL)	Job fails with IEFC630I	N/A	Job fails with IEFC630I	N/A

Table 8. SMS Allocation Processing Based on System Levels and Allocation Request (z/OS V2R1 with OA50569) (continued)

FACILITY class resource	**STGADMIN.SMS. FAIL.INVALID.DSNTYPE.ENC**			
Access	**Not defined OR not authorized**		**At least authorized for READ**	
Allocation type	**JCL**	**IDCAMS DEFINE**	**JCL**	**IDCAMS DEFINE**
Key label specified on IDCAMS DEFINE (KEYLABEL)	N/A	Allocation fails with IDC3211I	N/A	Allocation fails with IDC3211I
Key label from DFP segment of RACF DS profile or from data class)	Refer to the following two rows			
SMS mgd	Allocation continues with IGD17156I	Allocation continues with IDC3040I	Allocation fails with IGD17154I	Allocation fails with IDC0017I
non-SMS mgd	Allocation continues with IGD17156I	Allocation continues with IDC3040I	Allocation fails with IGD17154I	Allocation fails with IDC0017I

Accessing encrypted data sets

Applications that use standard BSAM, QSAM, and VSAM APIs do not require changes to access encrypted data sets. The user data transferred between the application and the access methods is in the unencrypted form. The access method encrypts the data when writing to DASD and decrypts the data when reading from DASD. For encrypted compressed format data sets, the access method compresses the data before encrypting it on output. On input, the access method decrypts the data before decompressing it.

Considerations regarding ICSF startup and shutdown

If you plan to encrypt SMF data sets or other data sets used during z/OS initialization, you must ensure that ICSF is started early in the IPL process to avoid delays in z/OS initialization and termination.

The ICSF parameter specifies the suffix of the ICSF on z/OS, CSFPRM*xx*, to be used by the system. The two characters represented by *xx* are appended to CSFPRM during ICSF initialization to form the name of the CSFPRM*xx* member. When specified in conjunction with the ICSFPROC parameter, the ICSF on z/OS starts at IPL-time. The ICSFPROC parameter specifies the name of the ICSF procedure you want the system to use to automatically start ICSF on z/OS at IPL-time. The procedure must reside in a SYS1.PROCLIB data set. When specified in conjunction with the ICSF parameter, ICSF on z/OS starts at IPL-time.

Furthermore, during z/OS system shutdown, ICSF must be one of the last features to stop so that dependent functions are not impacted. It is highly recommended that you shut down ICSF after terminating the JES address space and after initiating SMF halt processing. Note when ICSF is stopped after SMF is halted, that there might not be an SMF record cut for the termination of ICSF. (The ability to start ICSF with SUB=MSTR is available on all supported releases of ICSF.)

Considerations regarding backup/migration/replication functions

Key label authorization

The following system functions maintain data in the encrypted form. Therefore, users performing these functions do not require authorization to the key label associated with the data sets being processed with these functions:

- DFSMSdss functions: COPY, DUMP, and RESTORE
- DFSMShsm functions: migrate/recall, backup/recover, abackup/arecover, dump/data set restore, FRBACKUP/FRRECOV DSNAME
- Track based copy (PPRC, XRC, FlashCopy, concurrent copy) operations

Other considerations

- DFSMSdss REBLOCK keyword is ignored on COPY and RESTORE functions for encrypted data sets.
- DFSMSdss ADRREBLK installation exit will not be called for encrypted data sets.
- DFSMSdss does not support VALIDATE processing when backing up encrypted indexed VSAM data sets. VALIDATE will be ignored.
- Backup and migration of encrypted data sets may impact expected savings with disk or tape device compression. Where possible, convert to compressed format data sets. When data set level compression requested, access methods handle compression before encryption for compressed format encrypted data sets.

Considerations when planning for data set encryption

- Encrypted data sets must be extended format. Refer to *Extended-Format VSAM Data Sets* and *Processing Extended-Format Sequential Data Sets* for information on allocating extended format data sets, including guidelines and restrictions.
- Sequential extended format data sets can be version 1 or 2. When allocating an encrypted sequential extended format data set, the system creates the data set as an extended format version 2 data set, regardless of the user's specification for version number on DSNTYPE or the PS_EXT_VERSION keyword in the IGDSMSxx member in PARMLIB.
- Encrypted data do not compress. Therefore, encrypted extended format data sets might not benefit from storage savings when relying on disk and tape device compression, as well as other areas in the infrastructure that attempt to compress data. Where possible, consider using compressed format data sets. When data set level compression is requested along with encryption, access methods handle compression before encryption for encrypted compressed format data sets. Refer to *Compressed Data* and *Allocating Compressed-Format Data Sets* for information on allocating compressed format data sets, including guidelines and restrictions.
- When processing encrypted extended format data sets, the access methods may use additional internal system buffers during read and write processing. For optimal performance, consider increasing memory to your user address space as your use of encrypted data sets increases.

Restrictions for encrypted data sets

- System data sets (such as Catalogs, SMS control data sets, SHCDS, HSM data sets) must not be encrypted, unless otherwise specified in product documentation.
- Data sets used during IPL must not be encrypted.
- Encrypted data sets only supported on 3390 device types.
- Sequential (non-compressed) data sets with a BLKSIZE of less than 16 bytes cannot be extended format.

For potential restrictions associated with other products, consult the corresponding product documentation.

See the following publications for more details on data set encryption support:

References:

z/OS DFSMS Using Data Sets

z/OS DFSMS Access Method Services Commands

z/OS DFSMSdfp Storage Administration

z/OS DFSMSdss Storage Administration

z/OS DFSMShsm Storage Administration

z/OS Cryptographic Services ICSF Administrator's Guide

z/OS Cryptographic Services ICSF System Programmer's Guide

Chapter 13. Using the device support enhancements

DFSMS device support is enhanced to provide read-only access to data sets on specially-defined PPRC secondary devices. Using HCD panels, you can specify a READ-ONLY attribute for DASD devices to be varied online in read-only mode. I/O requests to those devices are limited to search and read commands; write commands to those devices are not supported. For details on defining the read-only devices, see *z/OS HCD Planning*.

A subset of data set types are eligible for read-only access on those devices – for a list of the data set types, their supported access methods, and the conditions and limitations for their use in read-only mode, see *z/OS DFSMS Using Data Sets*.

Applications must indicate that they are prepared to handle the conditions and limitations for read-only data, by issuing specific keywords on dynamic allocation, batch allocation, or open of the data sets. See details, see information about the DALROAC and DALROA2 text units in *z/OS MVS Programming: Authorized Assembler Services Guide* and about the ROACCESS keyword on the JCL DD statement in *z/OS MVS JCL Reference*.

All of the support for read-only PPRC secondary volumes is installed on the system when IHADFA flag DFAROSEC is set to 1.

Chapter 14. Using the DADSM/CVAF enhancements

z/OS DFSMS V2R3 provides the following DADSM/CVAF enhancements:

- **VTOC Update Safe interface**

  The VTOC consists of Data Set Control Blocks (DSCBs) that describe the data sets on the volume. A new CVAFDIR ACCESS=WRITE parameter, VALIDATE=(YES, NO), allows CVAF to read the existing DSCBs and compare them to the modified DSCBs that are passed by the user before updating the VTOC. The VALIDATE=(YES, NO) parameter checks the modified Format 1/8/9/3 DSCBs to be written to ensure the caller does not change fields that can corrupt the VTOC or disable the VTOC Index. If VALIDATE=YES is passed to CVAFDIR, then CVAF reads those DSCBs, and ensures the caller has not changed any fields that are not allowed to be changed.

  For more information about using the CVAFDIR ACCESS=WRITE parameter, see *z/OS DFSMSdfp Advanced Services*

  A related CVAF status code has been added. For more information, refer to *z/OS DFSMSdfp Diagnosis*.

  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, the VTOC audit log records the DSCB, along with the job name, date and time, and other fields to help identify the activity of the data set. This logging is saved in the SMF 42 subtype 27 record. For a description of Record 42, refer to *z/OS MVS System Management Facilities (SMF)*.

Chapter 15. Using the catalog enhancements

z/OS V2R3 includes the following new functions for DFSMS catalogs.

- **IDCAMS enhancement**:

  The DEFINE CLUSTER command has a new zFS parameter, which specifies that the cluster being defined is for linear data, and the linear data set is a z/OS UNIX file system. When zFS is specified, the linear data set is defined as extended addressable. LISTCAT will display

  ```
  ZFS
  ```

  whenever the data set is LINEAR and zFS indicator flag is on, or the zFS initialized flag is on.

- **Catalog Forward Recovery Enhancement**: There are multiple enhancements for DFSMS catalogs:

  This enhancement allows for a complete point-in-time recovery of a catalog, where all data is restored with consistency. MVS system logger REDO log records are used for logging updates. Recovery can be automated with the CICSVR utility. For more information, refer to the Procedure for Forward Recovering a Catalog in *z/OS DFSMS Managing Catalogs*.

Chapter 16. Using the Object Access Method (OAM) enhancements

z/OS DFSMS V2R3 provides the following enhancements to the object access method (OAM):

- The LIBRARY DISABLE,CATCOUNT command can be used to disable the category count I/O call to the library for transitions from private to scratch. Other transitions from scratch to private (during job processing) continue to issue the category count call to the library. Today, this I/O is done for every volume that is being returned to scratch. By default there is a monitoring task that obtains the scratch category counts every 10 minutes. As job processing continues to do this I/O, the elimination of this extra I/O might improve performance during return to scratch processing. To re-enable this I/O call LIBRARY RESET,CATCOUNT can be issued.

  This support was introduced with APAR OA48240.
- The LIBRARY DISPDRV command is enhanced with new MOUNTED and MOUNTED,ALL parameters to optionally display:
  - Status information for volumes that are mounted in the TS7700 Virtualization Engine for a specified composite or distributed library. Information pertaining to the distributed library that owns the device for the mount as well as distributed library information associated with the primary and the secondary tape volume cache (TVC) is displayed.
  - For a distributed library in a TS7700 Virtualization Engine, additional drives that are not owned by the specified distributed library can be displayed using MOUNTED,ALL. The additional drives are displayed if the distributed library specified is the primary or secondary TVC for the mounted volume. This option can be useful when a distributed library is going into service and can be used to determine if the specified distributed library is associated with a mount request from another distributed library.

  This support was introduced with OA47487 (z/OS V1R13 and above).

  For more information, refer to *z/OS DFSMS OAM Planning, Installation, and Storage Administration Guide for Tape Libraries* .
- The IGDSMSxx member of PARMLIB keywords DB2SSID, OAMTASK, and OAMPROC are enhanced in support of a new optional multiple OAM configuration. In a classic OAM configuration, the DB2SSID, OAMTASK, and OAMPROC keywords can continue to be specified in the IGDSMSxx member of PARMLIB as in the existing environment. However, in a multiple OAM configuration, changes are made to the existing OAM related keywords and a specification of the DB2SSID keyword is ignored.
- The IEFSSNxx member of PARMLIB is enhanced in support of a new optional multiple OAM configuration. In a classic OAM configuration, a single OAM subsystem can continue to be configured in the IEFSSNxx member of PARMLIB. While in a multiple OAM configuration, multiple OAM Object subsystems can be configured in addition to a single OAM Tape Library subsystem. A new D= keyword is used to specify the SSID or Group Attachment Name of the Db2 subsystem associated with the OAM subsystem being defined , and is required in a multiple OAM configuration. The IEFSSNxx member of PARMLIB keyword TIME is also enhanced. Either GMT or LOC can be specified as values for the TIME keyword. However, LOC is the default and indicates that local time should be used.
- The CBROAMxx member of PARMLIB is enhanced in support of a new optional OAM configuration. The ONLYIF statement can be used as before to specify whether specific statements within the CBROAMxx PARMLIB member are to be processed on a given system and can now also be used to specify whether statements are to be processed by a given OAM address space. In a multiple OAM configuration, optical related keywords on the SETOPT statement are validated but not used. Similarly, the CBROAMxx member of PARMLIB keyword SETOSMC is also enhanced so that optical related keywords on the statement are ignored in a multiple OAM configuration. And, in the CBROAMxx member of PARMLIB, optical related keywords on the SETOSMC statement are ignored in a multiple OAM configuration.

- The OAM Startup Procedure is enhanced in support of a new optional multiple OAM configuration. A new D= parameter identifies the SSID or Group Attachment Name of the Db2 subsystem to be associated with an OAM Object address space or indicates that the address space is a Tape Library address space by specifying D=NONE. D= is required in a multiple OAM configuration and not allowed in a classic OAM configuration. Additionally, multiple individual OAM procedures can be created (OAMA, OAMB), one for each OAM address space to be started in a multiple OAM configuration. Guidance on using the new D= parameter is added to the CBRAPROC member of SAMPLIB.
- The OAM Operator commands MVS MODIFY and MVS DISPLAY are enhanced in support of a new optional multiple OAM configuration. Guidance is provided on interacting with a specific OAM address space (F OAMA) using the MVS MODIFY command as well as multiple OAM address spaces using wildcarding. Guidance is also provided on the commands that can be used in a multiple OAM configuration as well as commands that can be used in a classic OAM configuration. OTIS supports new keywords on the MODIFY command to allow the operator to remove one or more OAM subsystems from the OAM configuration or to re-establish a connection to a Db2 subsystem. The MVS DISPLAY command is enhanced to display the overall OAM topology to assist with operations.
- The OSREQ API changes are enhanced in support of a new optional multiple OAM configuration. The OSREQ API is changed to provide a new DB2ID keyword to specify the Db2 subsystem (for a single OAM) or Db2 Group Attachment Name (for an OAM that is part of an OAMplex) to be used. The OSREQ sample program CBROSR2 remains unchanged for use with a classic OAM configuration, and a new sample program CBROSR3 is added for a multiple OAM configuration allowing for the specification of the DB2ID value to be used for a set of requests in a multiple OAM configuration.
- The TSO/E OSREQ Command processor is enhanced to allow specification of the DB2ID value in a multiple OAM configuration. DB2ID specifies the Db2 subsystem ID (for a single OAM) or Db2 group attachment name (for an OAMplex) that an OAM subsystem (and optionally a started OAM address space) has been configured to use. In a multiple OAM configuration, the DB2ID keyword is required for all OSREQ TSO/E Command Processor functions.
- The CBRSAMUT and CBRAPROC members of PARMLIB are enhanced to provide guidance on using the new DB2ID keyword in a multiple OAM configuration. The new CBROSR3 member of PARMLIB is created to show how to use OSREQ with the new DB2ID keyword in a multiple OAM configuration.
- The OAM SMF records are enhanced in support of a new optional multiple OAM configuration. OAM SMF records are updated to also include the actual OAM subsystem ID (OAM1,OAM2) as well as the associated Db2 subsystem ID (DB2A, DB2B).
- The OAM collection related function OAMUTIL CHGCOL is enhanced to include a new DB2ID keyword to identify the specific Db2 subsystem containing the Db2 Collection Name Table to be modified. Changes to the MVS catalog are no longer done since OAM collections are no longer maintained in the catalog.

Chapter 17. Using the new VSAM functions

z/OS V2R3 introduces the following enhancements for VSAM:

- DFSMS supports the storage of BSON and JSON objects (documents) into VSAM data sets. The changes to VSAM RLS are as follows:
  - You can specify NoSQL databases (known as VSAMDBs) to contain such objects using the new DATABASE parameter on the DEFINE CLUSTER command.
  - A new KEYNAME parameter on the DEFINE CLUSTER command, only specifiable with DATABASE, lets you specify a primary key for indexing documents in the database.
  - New ALTKEYS and ALTKEYSU parameters on the DEFINE ALTERNATEINDEX command allow you to specify alternate key names for the AIX against a base cluster with DATABASE, and the sorting order (ascending or descending) for each alternate key name.
  - New MACRF keyword values (DB or NODB) on the ACB, GENCB ACB, and MODCB ACB macros also give you the option to have RLS process the contents of a VSAM database as a regular KSDS, or as a database.
  - New OPTCD keyword values ARA31 or ARA64 (mutually exclusive) on the RPL, GENCB RPL, and MODCB RPL macros allow VSAM RLS records or VSAMDB documents to be passed to or retrieved from VSAM RLS to an area above the 2GB address bar in the user's address space.
  - New OPTCD=FRD on the RPL, GENCB RPL, and MODCB RPL will position to the first record in a regular VSAM data set or the first document in a VSAMDB.

For more information, see Defining a VSAM database in *z/OS DFSMS Using Data Sets*. See also *z/OS DFSMS Access Method Services Commands* and *z/OS DFSMS Macro Instructions for Data Sets*.

Chapter 18. Using the SMS enhancements

In z/OS V2R3, SMS introduces the following enhancement.

zHyperLink storage class granularity

In z/OS V2R3, SMS introduces the ability to specify whether data sets associated with a storage class are eligible to use zHyperLinks for reading and writing. zHyperLinks can dramatically reduce latency by interconnecting the z14 Central processor complex directly to the I/O bays in the IBM DS8880. This can improve application response time, without significant application changes. For more information, seeDefining use of zHyperlinks in *z/OS DFSMSdfp Storage Administration*.

Chapter 19. Using the DFSMShsm enhancements

In z/OS V2R3, DFSMShsm is enhanced with the following new functions:

- **DFSMS UNIX File Backup**

  APAR OA52703 provides DFSMS backup of UNIX files. This support gives you the ability to perform availability management of your z/OS UNIX files using DFSMSdss and DFSMShsm commands.

- **Common recover queue (CVQ) enhancements**

  With this enhancement, storage administrators have the ability to distribute the processing of volume restores (RECOVER *, FRRECOV COPYPOOL, and FRRECOV TOVOLUME with FROMDUMP) to all DFSMShsm tape resources available in a group of DFSMShsm hosts. New parameters on the SETSYS COMMONQUEUE command define the common recover group and its members (hosts) 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. Message simplification supports a host running a common recover queue. 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, copy pool auto dump, and FRRECOV TOVOLUME FROMDUMP, DFSMShsm now collects all related DFSMShsm and DFSMSdss messages and records them in a unique fast replication message data set.

  The message simplification function also supports the common dump queue and common recover queue processing.

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 the common recover queue” on page 70	• “Using the SETSYS COMMONQUEUE function(CONNECT(basename)) command to connect hosts as members of a functional group in an HSMplex” on page 70 • “Using the QUERY COMMONQUEUE command to display requests queued or active in the Common Recover Queue (plus other QUERY options)” on page 70 • “Using HOLD for volume restores in a common recover queue” on page 70 • “Using ALTERPRI to change the priority of queued requests in the common recover queue” on page 71 • “Using the DISPLAY command to monitor XCF for the common recover queue” on page 71
“Administering message simplification” on page 71	• “Using MESSAGEDATASET with the FRBACKUP and FRRECOV COPYPOOL/TOVOLUME commands to specify the date and time used in the message data set name” on page 71

Administering

The new V2R3 functions require the following administration tasks.

Administering the common recover queue

This topic describes the administration tasks for using the common recover queue (CVQ) enhancements in V2R3.

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

Use the new RECOVER(CONNECT(*basename*)) subparameter on the SETSYS COMMONQUEUE command to connect hosts as members of the common recover queue (CVQ) group of hosts in an HSMplex. When specified, this system can send and receive volume restores to and from other systems in the HSMplex that start with the same *basename*.

The DFSMShsm host that manages all of the volume restore requests in the CVQ is called the master scheduler. All volume restore requests are sent to the master scheduler, and it assigns work to all of the other hosts eligible in the group to process requests. Each group needs a master scheduler to manage CVQ processing for the group; otherwise, volume restore requests are processed locally.

A system becomes a candidate for CVQ master scheduler when SETSYS COMMONQUEUE (RECOVER(CONNECT(*basename*)) is specified.

Note: MASTERSCHEDULERCANDIDATE(YES) is the default. To prevent this host from becoming a master scheduler for the group, you should specify MASTERSCHEDULERCANDIDATE(NO).

One of the hosts that connects to the group and is a master scheduler candidate becomes the master scheduler.

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 Common Recover Queue (plus other QUERY options)

Use the QUERY COMMONQUEUE with the RECOVER sub-parameter to display to display the status of RECOVER and FRRECOV requests that are being processed and queued in the Common recover queue (CVQ) of which the host issuing the command is a member. This query provides a snapshot of all CVQ activity across all group members.

The QUERY SETSYS command displays the current settings for the common recover queue. For QUERY ACTIVE, QUERY WAITING, QUERY REQUEST, and QUERY USER, in a CVQ, only requests that originate 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 HOLD for volume restores in a common recover queue

Using HOLD for volume restores in a common recover queue.

To prevent the master scheduler host from selecting a CVQ-connected host to process volume restores from tapes, use SETSYS MAXDUMPRECOVERTASKS(0) on hosts that should not be selected for CVQ processing. An alternative is to use the HOLD RECOVER and FRRECOV(TAPE) commands, but this also holds all Master Scheduler assignment duties on a host that is the Master Scheduler or a candidate for one. It is recommended that SETSYS MAXDUMPRECOVERTASKS(0) be used to configure hosts to not process volume restores, while HOLD RECOVER or FRRECOV(TAPE) be reserved for situations where ongoing volume restore processing needs to be held for environmental reasons.

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

For environments where a host is connected to a common recover queue (CVQ), 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 CVQ of which the system issuing the command is a member.

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

Using the DISPLAY command to monitor XCF for the common recover queue

Using the DISPLAY command to monitor XCF for the common recover queue.

The DISPLAY XCF command can be used to monitor the XCF facilities utilized by the common recover queue. Tracking and monitoring RECOVER, and FRRECOV functions remain unchanged.

Administering message simplification

This topic describes the administration tasks for using the message simplification enhancements in V2R3.

Using MESSAGEDATASET with the FRBACKUP and FRRECOV COPYPOOL/ TOVOLUME 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, FRRECOV COPYPOOL request, or FRRECOV TOVOLUME FROMDUMP requests. 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, see the FRRECOV command description in *z/OS DFSMShsm Storage Administration*.

Chapter 20. Using the DFSMSrmm enhancements

The functional enhancements available with z/OS V2R3 DFSMSrmm provide you with these benefits:

Enhancements to EXPDT retention method

The DFSMSrmm has been enhanced with these new functions:

- *Specify expiration time for added volumes and data sets:* The ADDDATASET, ADDVOLUME, and GETVOLUME subcommands 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.
- *Specify that an added data set not expire while cataloged:* A new WHILECATALOG option has been added to the EDGRMMxx parmlib member OPTION command and to the ADDDATASET TSO subcommand to enable the user to specify either that:
  - the added 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 added dataset will be kept as long as it is cataloged, but no later than the expiration date.

Chapter 21. Using the DFSMSdss enhancements

z/OS V2R3 introduces the following enhancements for DFSMSdss:

- Added the new DFSMSdss SPACEREL command, which allows you to release unallocated free space extents from specified extent space efficient volumes at the volume and storage group level. For more information, see *z/OS DFSMSdss Storage Administration*.
- Increased the maximum number of data sets that can potentially reside on a logical data set backup. The previous maximum of 131,070 data sets is increased to allow up to 2,147,483,392 data sets. This number represents the number of data sets that passed the INCLUDE and EXCLUDE filtering checks only, and does not mean that the backup actually contains that amount of data sets.

  If the limit is exceeded, message ADR865E is issued. Several fields in the ADRTAPB data area, in *z/OS DFSMSdss Storage Administration*, have been updated to support this increased limit.
- Added a reason code to message ADR383W, to identify specific causes for a data set not being selected and specific actions to be taken.
- For OA52703, added a new section to describe backing up and restoring individual UNIX files.
- For OA52703, added new section for Unix file filtering.
- For OA52703, added new profile STGADMIN.ADR.DUMP.TOLERATE.WRITERS.
- For OA52703, added new section for Backing up UNIX files.
- For OA52703, added new section for Restoring UNIX files.
- For OA52703, added ADRTAPB version 2 data area.
- For OA52703, added new patch Using RESET with CLONE.
- For OA52703, added new section for DFSMSdss filtering-choosing the z/OS UNIX files you want processed.
- For OA52703, added new DUMP PATH and RESTORE PATH syntax.
- For OA52703, added CLONE, NOTIFYCLONE, PATH, and WORKINGDIRECTORY keywords in both Explanation of Dump and Restore command keywords.
- For OA52703, added Eioption 41-43 to User interaction module exit option descriptions.
- For OA52703, added new chapter z/OS UNIX file attributes.

Chapter 22. Using the DFSMStvs enhancements

In z/OS V2R3, DFSMStvs is enhanced with the following new function:

- **Automatic commit support**

  DFSMStvs provides a new parameter on the JCL EXEC statement that you can use to enable the automatic issuing of commit points on behalf of a batch application. The TVSAMCOM parameter lets you specify how many updates (an exact number or a range of numbers) should occur before a commit point is issued. You can also specify a default value for this parameter in the IGDSMS*xx* member of SYS1.PARMLIB (the EXEC statement value takes precedence over values in IDGSMS*xx*).

  For more information about using automatic commit, including the types of batch jobs that are good candidates for it, see *z/OS DFSMStvs Planning and Operating Guide*.

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

Tasks	Procedure that you must perform:
"Programming" on page 77	• "Using the TVSAMCOM parameter on the JCL EXEC statement to enable automatic commits in batch jobs" on page 77 • "Specifying a TVSAMCOM value in IGDSMSxx" on page 77

Programming

The new V2R3 function requires the following programming tasks.

Using the TVSAMCOM parameter on the JCL EXEC statement to enable automatic commits in batch jobs

For suitable batch jobs, you can code the optional TVSAMCOM parameter on the JCL EXEC statement, as follows:

```
//stepname EXEC positional-parm, TVSAMCOM=({minval|0}),({maxval|0})
```

The *minval* and *maxval* values specify the minimum and maximum for a range of numbers of update requests that DFSMStvs uses to determine when and if to call RRS services to issue a commit point on behalf of the batch application. The default value for *minval* is 1 and the default value for *maxval* is 0. If you do not specify a TVSAMCOM parameter, then an TVSAMCOM value set in the **IGDSMSxx** member of SYS1.PARMLIB will be used.

For details, see the description of the JCL EXEC statement's TVSAMCOM parameter in *z/OS MVS JCL Reference*.

Specifying a TVSAMCOM value in IGDSMS*xx*

You can specify TVSAMCOM values in the IGDSMS*xx* member of SYS1.PARMLIB, for DFSMStvs to provide automatic commit points for batch applications. The IGDSMS*xx* values are used if an application does not specify a TVSAMCOM parameter on the JCL EXEC statement. The default for the TVSAMCOM values in IGDSMS*xx* (both *minval* and *maxval*) is 0, which specifies that no automatic commits be performed.

You can specify the TVSAMCOM values by coding them directly in IGDSMS*xx*, or by setting them with the SETSMS command.

For details on coding the IGDSMS*xx* TVSAMCOM parameter, see IGDSMSxx in *z/OS MVS Initialization and Tuning Reference*.

For details on the SETSMS command's TVSAMCOM parameter, see the SETSMS command in *z/OS MVS System Commands*.

Chapter 23. Using the Advanced Copy Services enhancements

z/OS V2R3 adds the following Advanced Copy Services enhancements.

- XRC device-based management supports XRC primary volumes that are offline and may never have been online to the data mover system. TSO commands for XRC and the associated ANTRQST and REXX interfaces now support adding devices by device number rather than volume serial. The enhancements include:
  - The addition of the PDEVNUM keyword, and the SDEVNUM keyword for add requests, in XADDPAIR, XDELPAIR, XSET, and XSUSPEND commands, as well as the XADD, XDEL, XSET, and XSUSPEND APIs provided with ANTRQST and ANTTREXX.
  - The addition of the DEVNUM keyword to the XQUERY TSO command and related ANTRQST and ANTTREXX APIs. This supports looking for multiple volumes that have the same volume serial.
  - A new MessageVolumeFormat parameter of parmlib member ANTXIN00, which controls whether messages that include volume information show the volume serial, the device number, or both.
  - XQUERY reports indicating volumes that are offline with an apostrophe (') after the volume serial.

  For more information, see *z/OS DFSMS Advanced Copy Services*.
- Remote Pair FlashCopy® (RPFC) is a solution for mirroring the results of a point-in-time copy (FlashCopy) from the primary volumes to the secondary volumes in a remote mirror configuration, without disrupting the mirror or consistency at the remote site. RPFC captures the parameters of a FlashCopy and sends those parameters to the secondary volumes, replicating the operation. RPFC has been available only for Metro Mirror (PPRC) environments, which are synchronous mirroring environments. This support provides RPFC capability in environments where Cascading FlashCopy is available on the XRC secondary volumes, as well as limited RPFC capability for z Global Mirror (zGM or XRC) environments.

  At the local site, you enable RPFC by:
  - Specifying the FCTOXRCPRIMARY keyword on the DFSMSdss COPY command. It is supported on all COPY commands, including FULL, TRACKS, and DATASET, for logical and physical data sets.
  - Specifying FLASHCOPYTOXRC=YES in the DEVSUP*xx* member of parmlib and activating the new parameters

  At the remote site, you enable RPFC by:
  - Specifying the RemotePairFlashCopy parameter in the ANTXIN00 member of parmlib. This parameter takes effect only during XSTART. It cannot be changed using XSET while the session is active.

  When defining a copy pool, you can specify whether an XRC primary volume is eligible to become a FlashCopy target volume for FRBACKUP or FRRECOV. In ISMF, you use these new fields: FRBACKUP to XRC Primary Volumes allowed and FRRECOV to XRC Primary Volumes allowed.

  Invoke RPFC on an FCESTABLISH request of the ANTRQST macro, with the new TGTXRCPR keyword.

  When RPFC is enabled (with the RemotePairFlashCopy parameter of parmlib member ANTXIN00), XRECOVER checks the secondary and tertiary volumes for support of Cascading FlashCopy.

  XQUERY reports display new values related to RPFC. XQUERY ENV(PARM) shows the values for the new parmlib settings. XQUERY XFEATURES shows the microcode and software enablement status for RPFC for XRC.

  For more information, see *Remote Pair FlashCopy for XRC* .

Part 5. Using new DFSMS functions in z/OS V2R2

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

Chapter 24. 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.

Catalog forward recovery: This enhancement allows for a complete point-in-time recovery of a catalog, where all data is restored with consistency. MVS system logger REDO log records are used for logging updates. Recovery can be automated with the CICSVR utility. For more information, refer to the Procedure for Forward Recovering a Catalog in *z/OS DFSMS Managing Catalogs*.

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.

- **REPROMERGECAT output enhancement**: IDCAMS will provide new optional keywords, MESSAGELEVE(ALL|SHORT) for REPRO MERGECAT. 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 IGGCAT*xx* 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 25. 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. Space Constraint Relief is disabled by default. 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:

- Space Constraint Relief is available on Secondary space as of V2R2 and is also known as Secondary Space Reduction (SSR).
- 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 about the secondary space reduction support, see the following documentation:

- IBM Redbooks® section on space constraint relief for secondary allocation in *IBM z/OS V2R2: Storage Management and Utilities*
- Specifying attributes to handle space constraints during allocation in *z/OS DFSMSdfp Storage Administration*
- Allocation of data sets with the space constraint relief attributes in *z/OS DFSMS Using Data Sets*

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, COMMDS and SCDS. 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 job logs 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 IGDSMS*xx* 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 Defining ACS routines in *z/OS DFSMSdfp Storage Administration*.
- For details on the IGDSMS*xx* member of PARMLIB, refer to IGDSMSxx in *z/OS MVS Initialization and Tuning Reference*.
- For details on the SETSMS command, refer to the SETSMS command 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 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 26. 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:

- **DEVSUP*xx* enhancements**: In previous releases, a subset of DEVSUP*xx* 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 DEVSUP*xx* member is processed, unless the member explicitly specifies new values.

  In addition, beginning in V2R2, multiple DEVSUP*xx* 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 *z/OS DFSMS Macro Instructions for Data Sets* .

Chapter 27. 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 28. 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 Record 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 29. 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 SMFPRM*xx* to disable VTOC audit logging, as follows: NOTYPE(42(27)). For information, refer to SMFPRMxx (system management facilities (SMF) parameters) 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 30. 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 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" on page 100	• "Using the MIGRATE command to perform class transitions at the volume level" on page 100 • "Using the MIGRATE command to perform class transitions at the storage group level" on page 101 • "Using the MIGRATE command to perform class transitions at the data set level" on page 101 • "Using the MIGRATE command to perform data set moving" on page 101
"Administering distributed tape processing" on page 101	• "Using the SETSYS COMMONQUEUE function(CONNECT(basename)) command to connect hosts as members of a functional group in an HSMplex" on page 101 • "Using the QUERY COMMONQUEUE command to display requests queued or active in the group's common queue (plus other QUERY options)" on page 102 • "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 102 • "Using the UPDTCDS command to update fields in the control data sets" on page 102 • "Using ALTERPRI to change the priority of queued requests in the common queue" on page 103
"Administering message simplification" on page 103	• "Using the SETSYS FASTREPLICATION command to direct messages to a message data set" on page 103

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. Use the new RECOVER(CONNECT(*basename*)) subparameter on the SETSYS COMMONQUEUE command to specify remote full volume recover from dump processing. When specified, this system will be allowed to send and receive dumps/recovers 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/recover queue master scheduler when the MASTERSCHEDULERCANDIDATE(YES) subparameter is specified with either SETSYS COMMONQUEUE DUMP(CONNECT(*basename*)) or SETSYS COMMONQUEUE RECOVER(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 or RECOVER subparameter to display the status of dump/recover requests that are being processed and queued in the common dump/recover queue of which the host issuing the command is a member. This query will provide a snapshot of all dump/recover 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 or recover 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 31. 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.

Appendix A. Accessibility

Accessible publications for this product are offered through IBM Documentation for z/OS (www.ibm.com/docs/en/zos).

If you experience difficulty with the accessibility of any z/OS documentation see How to Send Feedback to IBM to leave documentation feedback.

Notices

This information was developed for products and services that are offered in the USA 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 grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

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.

This information could include missing, incorrect, or broken hyperlinks. Hyperlinks are maintained in only the HTML plug-in output for IBM Documentation. Use of hyperlinks in other output formats of this information is at your own risk.

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

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

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:

IBM Corporation
Site Counsel
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 document 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.

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

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

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

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

COPYRIGHT LICENSE:

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

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or

reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's name, email address, phone number, or other personally identifiable information for purposes of enhanced user usability and single sign-on configuration. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at ibm.com®/privacy and IBM's Online Privacy Statement at ibm.com/privacy/details in the section entitled "Cookies, Web Beacons and Other Technologies," and the "IBM Software Products and Software-as-a-Service Privacy Statement" at ibm.com/software/info/product-privacy.

Policy for unsupported hardware

Various z/OS elements, such as DFSMSdfp, JES2, and MVS, 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 (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 on the Web at Copyright and Trademark information (www.ibm.com/legal/copytrade.shtml).

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

Java™ and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux®® is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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.

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 to 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

Product Number: 5655-ZOS

SC23-6857-70